Chatbot Guided Domain-science Knowledge Discovery in a Science Gateway Application

Sai Swathi Sivarathri, Prasad Calyam, Yuanxun Zhang,
Ashish Pandey, Cong Chen, Dong Xu, Trupti Joshi and Satish Nair
University of Missouri-Columbia, USA
{*sss26x, yzd3b, apfd6, ccv5f*} *@mail.missouri.edu,*
calyamp@missouri.edu, xudong@missouri.edu, joshitr@health.missouri.edu, nairs@missouri.edu

***Abstract*—Neuroscientists are increasingly relying on high performance/throughput computing resources for experimentation on voluminous data, analysis and visualization at multiple neural levels. Though current science gateways provide access to computing resources, datasets and tools specific to the disciplines, neuroscientists require guided knowledge discovery at various levels to accomplish their research/education tasks. The guidance can help them to navigate them through relevant publications, tools, topic associations and cloud platform options as they accomplish important research and education activities. To address this need and to spur research productivity and rapid learning platform development, we present "OnTimeRecommend", a novel recommender system that comprises of several integrated recommender modules through RESTful web services. We detail a neuroscience use case in a CyNeuro science gateway, and show how the OnTimeRecommend design can enable novice/expert user interfaces, as well as template-driven control of heterogeneous cloud resources.**

***Index Terms*—neuroscientists, science gateways, recommender system, ontimerecommend, chatbot guided user iterface, knowledge discovery**

I. Introduction

Research and training in neural science and engineering increasingly deals with diverse and voluminous multi-parameter data [1], posing unique challenges outlined in an NSF iNeuro report [2] as limited access to: multi-omics data archives [3], heterogeneous software [4] and computing resources (Neuroscience Gateway [5], Amazon Web Services (AWS)), and multi-site interdisciplinary expertise (e.g., engineering, biology and psychology). Existing distributed high-performance computing resources (HPC) and other cyber infrastructures (CI) enable access to analyze and visualize such data. However, to fully utilize their capabilities, neuroscientists (often with limited CI skills) are required to take valuable time away from the focus of knowledge discovery in neuroscience, to learn how to use such technology.

There is a consensus that neuroscience research is also hampered by limited exchange of ideas, sharing of data, and collaboration within the community. In addition to these challenges, other factors such as reproducibility, usability, automation, and distributed cognition [6] have been significant barriers for use of science gateways in computational neuroscience. Labs pursue their individual research independently, distribute their research via journal articles, and reinvent the computing pipelines used in their analysis [7] time and again. To reproduce experiments done in other labs, it is imperative to disseminate the exact code and procedure used during the experimentation. This can be challenging especially to share distributed data for different artifact types and locations.

Current practices often results in redundant analysis scripts from these independent labs, which are often difficult to reproduce, due to poor coding skills, lack of version control and manual implementation of certain tasks [8]. Moreover, additional data at all levels of neuroscience continues to accumulate at rapid rates and volumes [9] [10]. However, the community lacks effective CI tools to harness such data sets and to also foster effective interdisciplinary interactions to advance the 'team science' research needed in neuroscience, in a scalable, reproducible, sustainable, and efficient manner. Productivity is also influenced by automating the workflow templates of common tasks that are repetitive. Hence, the template composition should be customized based on users' domain knowledge and workflow needs.

In this paper, we address the above needs and aim to spur research productivity and interdisciplinary collaborations within neuroscience (i.e., our target community). Specifically, we describe our "OnTimeRecommend", a novel recommender system that comprises of several integrated recommender modules through RESTful web services. OnTimeRecommend features a novel recommender architecture to increase the effectiveness of novice/expert neuroscientists using CI resources in workflow management within next-generation science gateways, as well as template-driven control of heterogeneous cloud resources. The recommender system requirements are motivated by science application drivers within a CyNeuro science gateway that we are developing [11]. CyNeuro utilizes CI resources (e.g., Neuroscience Gateway, JetStream, XSEDE, AWS) in order to integrate data, tools for data analytics, computing, and visualization with cyber and software automation.

OnTimeRecommend features aim to supplement science gateways such as e.g., CyNeuro with application-facing modules that include: (i) Domain-specific Topic Recommender, (ii) Publication Recommender, (iii) Jupyter Notebook Recommender, and (iv) Cloud solution Template Recommender. Throughout a given workflow's lifecycle, users interact with a context-aware chatbot that is embedded with OnTimeRecom-

Presented at Gateways 2019, San Diego, CA, USA, September 23–25, 2019.
https://osf.io/meetings/gateways2019/

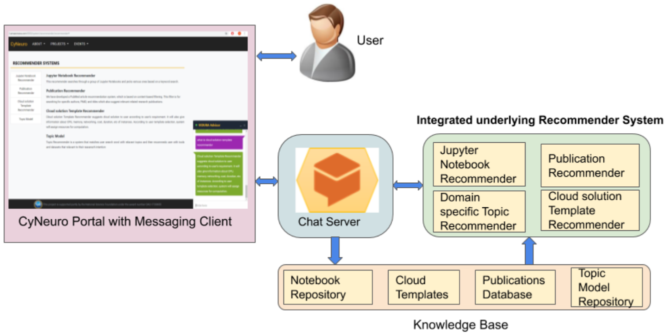

Fig. 1. Recommender system interacting with the CyNeuro portal through a conversational agent and chat server built using Dialogflow.

mend as a guided user interface. The chatbot provides step-by-step navigational support and generates distinct responses for users based on user intent for guided educational and research purposes. The ultimate goal of OnTimeRecommend development is to ensure extensibility via RESTful APIs for custom configuration capabilities to support e.g., “BringYourApp”, “BringYourOwnRecommender” as well as even “BringYourOwnHardware” within other science application community platforms using federated CI resources.

The remainder of the paper is organized as follows: Section II presents the overview of our recommender system. Section III presents an exemplar neuroscience use case for our recommender system. Section IV concludes the paper.

II. Recommender System Overview

Development efforts of science gateways in the recent past are increasingly brokering federated national resources to improve accessibility in HPC resources to domain science users. At the University of Missouri, a proof-of-concept for such a federated resource brokering has already been established to serve neuroscience educators and researchers [12]. Our CyNeuro science gateway efforts are in collaboration with the nationally accessible NeuroScience Gateway (NSG) [13] that provides a variety of hardware/software services for neuroscience scientists. Following the best practices from NSG, CyNeuro features a test deployment of the CIPRES workflow management software in addition to Matlab image analysis, and other simulation tools.

Current practices require users to fully install all necessary technologies on their local laptops and need to understand or have expertise over the relevant configurations with little to no expert guidance. These specific activities pose a challenge for students beginning to learn neuroscience-related courses that require software to be installed and used. Our CyNeuro development as shown in Fig. 1 features a number of technologies and recommender modules. Via knowledge base, the various features help users to easily and quickly find necessary tools or data sets for their research/education activities, establish cloud infrastructure configurations, search publications and execute custom codes in a guided manner. To facilitate multiple users to concurrently run simulations, we have deployed Jupyter Notebooks to run user codes through Python and MATLAB kernels. For knowledge discovery, we support tools that use topic models to find latest topics and trends in a scientific domain. In addition, CyNeuro uses cloud solution templates to configure CI resources.

Recommender systems with strong algorithms are at the core of today’s most successful online companies such as Amazon, Google, Netflix and Spotify. With the growing amount of information on the Internet and with a significant rise in the number of users, it is a difficult task for science gateways to search, map and prescribe relevant information suiting diverse user’s preferences and needs. For OnTimeRecommend, we have developed recommender modules to provide recommendations within science gateways to guide users based on their needs and preferences. By providing recommendations that address users’ requirements, our integrated recommender modules provide a personalized/expert advice across diverse computing platforms and data archives.

There are several examples of chatbots becoming user support agents in recent years [14]. In [15] and [16] authors present a conversational recommender based on user functional requirements for suggesting the most appropriate smartphone. More recently, the bibliometric study on chatbots [17] reveals the scope of contributions in the state-of-art for chatbot applications in many domains with a high number of alternatives involving automated user support.

In our previous work [14], we have proposed a conversational recommender that is used to provide a guided user interface and a chatbot functionality for neuroscience researchers and educators/students. The conversational agent design and its CyNeuro integration for a variety of recommenders is the novel contribution of this work. The users interact with our context-aware chatbot that is currently embedded within CyNeuro to help researchers/educators to define new/diverse scientific workflows using local and remote HPC resources and foster collaborative efforts through knowledge sharing.

III. OnTimeRecommend Recommender System

As shown in Fig. 2, “OnTimeRecommend” is a novel recommender system that comprises of several integrated recommender modules. The recommender modules are accessible through RESTful web services listed in Table I. The OnTimeRecommend design enables novice/expert user interfaces within next-generation science gateways, as well as template-driven control of federated CI resources. The OnTimeRecommend features application-facing modules that include: (i) Domain-specific Topic Recommender, (ii) Publication Recommender, (iii) Jupyter Notebook Recommender, and (iv) Cloud solution Template Recommender.

Each recommender module has a corresponding knowledge base, which is basically a trained model that uses a specific set of data sources. As an exemplar case, our domain-specific topic recommender [18] uses openly accessible publication archives and creates a knowledge base using back-end scripts for serving to the front-end (chatbot-enabled) web application. Through an expert user interface, the recommender collects

TABLE I

REST API FRAMEWORK FOR THE RECOMMENDER MODULES IN THE ONTIMERECOMMEND PROTOTYPE

Recommender Module	URL	HTTP Method	Parameters	Response	Description
Topic Recommender	/api/topics	POST	”text” : string	”topics”: [”id”:int, ”summary”:string,”tools”:string, ”datasets”:string, ...]	Receives topic as query input from user and returns JSON array of recommended topics that contain topic id, summary, tools and datasets that are relevant to the topic
Publication Recommender	/articles/title	GET	”title” : string	”title”: [”_id”:string, ”ID”:int, ”Abstract”:string,”PMID”:string, ”Title”:string, ...]	Receives title as query input from user and returns JSON array with content and necessary information based on the input title
Publication Recommender	/articles/authors	GET	”author” : string	”title”: [”_id”:string, ”ID”:int, ”Abstract”:string,”PMID”:string, ”Title”:string, ...]	Receives name of the author as query input from user and returns JSON array with content and necessary information based on the input author
Publication Recommender	/articles/pmid	GET	PMID : int	”title”: [”_id”:string, ”ID”:int, ”Abstract”:string,”PMID”:string, ”Title”:string, ...]	Receives PMID as query input from user and returns JSON array with content and necessary information based on the input PMID
Jupyter Notebook Recommender	/api/jupyter	GET	keyword : string	”notebooks”: [”filename”:string, ”cell_no”:string, ...]	Receives text as query input from user, searches and returns JSON array with Jupyter Notebooks related to a given keyword
Cloud solution Template Recommender	/api/template	GET	application_type : string, site: string,memory:string,cpu:int, networking:string	”resources”: [”sgn”:string, ”memory”:string, cpu:int, networking:string, duration:int,cost:int]	Receives cpu, memory and other functional parameters as query input from user and returns JSON array of CI resources assignment and expected cost

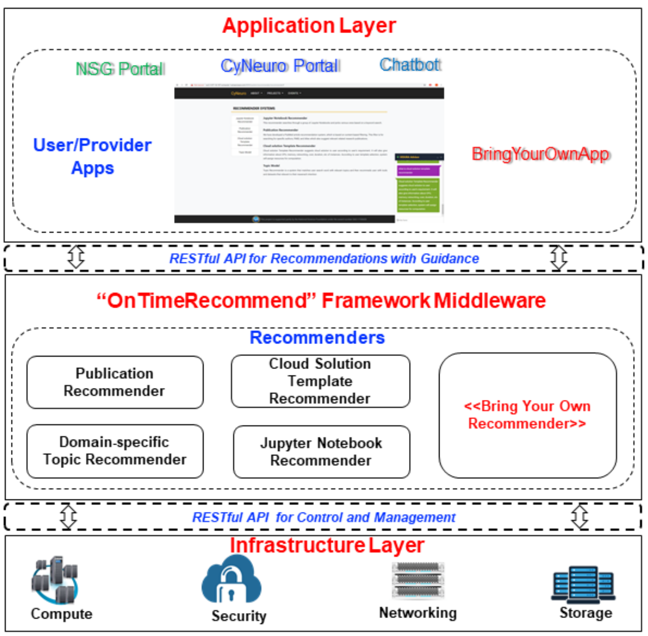

Fig. 2. OnTimeRecommend architecture with application-facing and infrastructure-facing RESTful APIs.

domain-specific knowledge (e.g., popular tools, data sets) and reputed publication archives. This guidance is then used to generate the recommendations for any domain-specific user (e.g., scientist, researcher, educator) queries. Updates to the knowledge base (i.e., the trained model) can be made periodically by querying various data sources, as well as by incorporating refinements from expert or target customer/user feedback to keep the knowledge base most up-to-date. Our approach to use REST APIs can further help in the continuous evolution of the OnTimeRecommend modules by exposing the most latest individual recommender capabilities. This in turn can help make the individual recommenders become both flexible and scalable for different data sources and user inputs.

IV. NEUROSCIENCE TASK USE CASE FOR THE RECOMMENDER SYSTEM

In this section, we illustrate an exemplar computational neuroscience use case that shows utility and benefits owing to our OnTimeRecommend system. For a neuroscience researcher who wants to build biophysical realistic single cell model, they typically have to perform extensive parameter searches to develop a model cell using biologically measured properties. We have two methods that can help researchers accomplish such a parameter search.

The first part is to simulate a single neuron model and quantify its electrophysiology properties given specific model parameters. The user can use the *Publication Recommender* to discover that a Hodgkin-Huxley type of neuron model can be used as an example. A corresponding literature search reveals the parameters such as membrane capacitance, maximum conductance and reversal potentials of channels, etc., that determine the electrophysiological properties such as resting membrane potential, membrane time constant and rheobase current, etc. For example, the rheobase current is defined as the smallest injected step current of infinite duration which results in one action potential. The general protocol is to inject currents of various amplitudes, observe if any action potential is produced, and then further refine the injected current magnitude until the boundary between spiking and non-spiking behavior is identified. Next, the user inputs model parameters such as gbar_na, gbar_k, gbar_leak to the *Juypter Notebook Recommender*, and relevant notebooks will returned with quantified properties for running simulations.

The second part is to predict the model parameters using a machine learning tool and related data sets given desired electrophysiology properties. To find the relevant tool and data set, the user relies on the *Domain-specific Topic Model Recommender*. For example, the user gets recommended publications that provide FIR (frequency-current relationship) curves from experimental data and information about which set of model parameter such as gbar_na, gbar_k and gbar_leak can provide the particular FIR curve. Using inputs such as membrane time constant, membrane resistance, resting membrane potential, and sample points on the FIR curve, the tool leverages a trained neural network to predict the model parameters. The tool also helps to compare the FIR curve given by predicted model parameters and the target FIR curve in one plot. All other passive properties and spiking properties will be generated using the predicted parameters for validation.

For both the above methods, the researcher has access to the relevant notebooks, tools, data sets required to run the simulations to meet the research/education objectives. To scale the simulations, the researcher may or may not have required CI resources to run the simulations on special hardware such as GPUs. In this scenario, the user can use the *Cloud Solution Template Recommender* to obtain a relevant solution as per the user's preferred functional and nonfunctional requirements such as cost, performance and agility. The user can then deploy suggested cloud templates, install relevant tools and run notebooks over the resources to get relevant outputs. Based on the results, users can have their own evaluations and can further deep dive into their research with the feedback obtained by again using the *Publication Recommender*. In this manner, the integrated recommender system acting as a single source of knowledge discovery as illustrated in Fig. 2 can be navigated by a chatbot interface using a dialog design methodology we used in [14] to profile a user's intent and proficiency profile. Thus, our integrated recommender system with a chatbot interface can cut down the manual/tedious work of knowledge discovery, and overall enhance the user's research/education productivity with convenience and resourcefulness.

V. Conclusion

In this paper, we have proposed an integrated recommender system, the "OnTimeRecommend" that comprises of four recommender modules along with a guided user interface and a chatbot functionality for neuroscience researchers and educators/students. The OnTimeRecommend design and integration with a CyNeuro science gateway was described supporting an exemplar neuroscience research use case. The recommender modules can guide researchers and educators to discover relevant resources, enhance interdisciplinary knowledge sharing and train on Jupyter notebook enabled learning exercises. A novel recommender architecture in OnTimeRecommend increases the effectiveness of novice/expert neuroscientists using CI resources in workflow management. Using text mining methods and topic modeling, the proposed recommender modules are aimed at fostering interdisciplinary collaborations around distributed databases, parallel and distributed computing resources for analysis and visualization of neuron simulations.

In future, our ongoing project activities seek to enable CI providers to integrate data analytic tools such that they improve efficiency of CyNeuro and other science gateway users in their research and education tasks.

Acknowledgment

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

References

[1] NSF (2016) Understanding the brain, [Online]. Available at: https://nsf.Gov/about/budget/fy2016/pdf/42_fy2016.pdf [Online]. Available at: https://www.Nsf.Gov/news/special_reports/brain/

[2] NSF (2015) Ineuro conference report, [Online]. Available at: https://mdcune.Psych.Ucla.Edu/modules/ineuro/reports/ineuro_conferencereport_20150804.pdf

[3] Neuro-Databases ((2019)) List of neurosciene databases, [Online]. Available at:https://en.Wikipedia.Org/wiki/list_of_neuroscience_databases

[4] OCNS (2016) Organization for computational neurosciences, software [Online]. Available at: https://www.Cnsorg.Org/software

[5] A. Majumdar, S. Sivagnanam, N. Carnevale, K. Yoshimoto, P. Gleeson, A. Quintana, "Neuroscience gateway - cyberinfrastructure providing supercomputing resources for large scale computational neuroscience research", *Frontiers in Neuroinformatics*, 2016.

[6] A. Rule, "Design and Use of Computational Notebooks", *PhD Thesis, University of California at San Diego*, 2018.

[7] J. Freeman, "Open source tools for large-scale neuroscience", *Current Opinion in Neurobiology*, No. 32, pp. 156 - 163, 2015.

[8] J. Kubilius, "A framework for streamlining research workflow in neuroscience and psychology", *Frontiers in Neuroinformatics*, Vol. 7, No. 52, 2014.

[9] D. Usoskin, A. Furlan, S. Islam, H. Abdo, P. Lonnerberg, D. Lou, J. Hjerling-Leffler, J. Haeggstrom, O. Kharchenko, P. Kharchenko, S. Linnarsson, P. Ernfors "Unbiased classification of sensory neuron types by large-scale single-cell rna sequencing", *Nat Neurosci*, 2015.

[10] J. Poulin, B. Tasic, J. Hjerling-Leffler, A. Trimarchi, " Disentangling neural cell diversity using single-cell transcriptomics", *Nat Neurosci*, Vol. 19, pp. 1131-1141, 2016.

[11] S. S. Nair, P. Calyam, D. Xu, T. Joshi, "Cyber and software automation in neuroscience". [Online - May 2019]. Available at: http://cyneuro.org/

[12] P. Calyam, & S. S. Nair, "Experiences from a Multi-disciplinary Course Sequence Development on Cyber and Software Automation in Neuroscience", *13th Gateway Computing Environments Conference (Gateways)*, 2018.

[13] S. Sivagnanam, A. Majumdar, K. Yoshimoto, V. Astakhov, A. Bandrowski, M. Martone, N. T. Carnevale, "Introducing the Neuroscience Gateway", *IWSG*, 2013.

[14] A. Chandrashekara, R. Talluri, S. Sivarathri, R. Mitra, P. Calyam, K. Kee, S. Nair "Fuzzy-Based Conversational Recommender for Data-intensive Science Gateway Applications", *IEEE International Conference on Big Data (Big Data)*, 2018.

[15] D. Widyantoro, Z. K. A. Baizal, "A framework of conversational recommender system based on user functional requirements", *Proc. of IEEE International Conference on Information and Communication Technology (ICoICT)*, 2014.

[16] Z. Baizal, D. Abdurahman, H. Widyantoro, N. Maulidevi, "Design of knowledge for conversational recommender system based on product functional requirements", *Proc. of IEEE International Conference on Data and Software Engineering (ICoDSE)*, 2016.

[17] H. Io, C. Lee "Chatbots and Conversational Agents: A Bibliometric Analysis", *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2017.

[18] Y. Zhang, P. Calyam, T. Joshi, S. Nair, D. Xu, "Domain-specific Topic Model for Knowledge Discovery through Conversational Agents in Data Intensive Scientific Communities", *IEEE International Workshop on Conversational Agents and Chatbots with Machine Learning (ChatbotML), in conjunction with IEEE Big Data*, 2018.