ONYX: Towards Extending Natural Language Interfaces for Data Visualization Tools through Interactive Task Learning

Marcel Ruoff*
Karlsruhe Institute of Technology

Brad A. Myers
Carnegie Mellon University

Alexander Maedche
Karlsruhe Institute of Technology

ABSTRACT

While natural language interfaces (NLIs) are increasingly utilized to simplify the interaction with data visualization tools, adapting NLIs to the individual needs and requirements of end users still requires the support of developers. Our *ONYX* system introduces an interactive task learning (ITL)-based approach which enables NLIs to effectively learn from end users through natural interactions. End users can enhance the NLI with new commands or adapt existing commands using direct manipulation, natural language instructions, or a combination of both. *ONYX* guides end users through the demonstration process and provides them with recommendations for possible actions based on background knowledge of the system to enable an efficient interaction. In order to trigger reflections and gain feedback on the design of *ONYX*, we are currently preparing a formative study to understand how to best integrate guidance and recommendation capabilities provided by the *ONYX* system into the interaction.

Index Terms: Human-centered computing—Natural language interfaces; Human-centered computing—Participatory design

1 INTRODUCTION

To provide end users a more intuitive way of using data visualization tools, natural language interfaces (NLIs) have been increasingly leveraged as an addition to established graphical user interfaces. Particularly, NLIs for data visualizations enable end users to specify commands and ask questions in their own words to work on their tasks [7, 15, 19]. To generate or adapt visualizations, contemporary NLIs extract the required chart type, encoding, aggregation and other relevant attributes from the user command using natural language processing (NLP) techniques and utilize this information to determine the best response to the command.

However, adapting NLIs to individual user needs and requirements requires developers experienced with NLP techniques and corresponding toolkits [14]. When end users experience a breakdown of the NLI due to a limited set of supported commands or because of an incorrect determination of the best response, they are currently unable to extend or personalize NLIs by themselves.

Interactive task learning (ITL) is a promising approach providing end users the ability to interactively teach the NLI using natural interactions [11]. End users can extend NLIs by *demonstrating* their intended interaction using direct manipulation, through natural language instructions, or by combining both. Following the demonstration, ITL systems can generalize the demonstrated action sequence based on the initial user command to allow the use of different parameter values [12]. In our approach, *ONYX* is embedded in a data visualization tool that integrates data from the COVID-19 pandemic [5] for all US states and dates since January 2020, such as the number of people infected, partially vaccinated and fully vaccinated. After teaching *ONYX* the command "Show me **fully vaccinated** vs. the **population** for all **states** as a **scatter plot**", *ONYX* also learns how to handle "Show me **fully vaccinated** vs **deaths** for all **dates** as a **bar chart**". However, current ITL-based NLIs do not guide the user during the demonstration process to improve the resulting commands, provide recommendations for actions based on background knowledge or involve the user in efficiently resolving ambiguity of the action sequence used to train the command.

In our paper, we describe the current design of *ONYX*, an advanced ITL-based NLI for data visualization tools. Furthermore, we outline our participatory design process which includes a formative study that explores how best to extend *ONYX* to guide end users through the demonstration process and provide them with recommendations for possible actions based on the system's background knowledge. *ONYX* is named after a gemstone and stands for: **O**ptimizing **N**atural language interfaces for **Y**our e**X**perience. We plan to contribute through our work with:

- *ONYX*, an ITL-based NLI for data visualization tools that enables end users to extend an existing NLI based on their needs and common command sequences.
- An ITL mechanism that provides recommendations based on existing commands and background knowledge of the data and visualization as well as guiding the end users to assist them in creating meaningful visualizations following existing guidelines.

2 RELATED WORK

2.1 Natural language interfaces for Data Visualization Tools

The increasing interest in NLIs for data visualization tools leads to different approaches for providing natural language-based interaction to end users. The majority of NLIs for data visualization tools provide users with a wide range of possible commands [6, 7, 16, 17, 21]. These NLIs aim to handle ambiguity [7] or underspecification [6, 17] through follow-up prompts within the graphical user interface (GUI) or follow-up questions articulated by a conversational agent. While this may enable end users to state complex commands, these NLIs are often more error prone [19] and end users may not know what commands and questions are supported by the NLIs [18]. In contrast, "restricted" NLIs like the initial prototype presented by Cox et al. [4] or InChorus [19] only support a limited range of questions as well as low-level commands. This type of NLI is not as prone to errors during the speech recognition and the determination of the best response [19]. However, these NLIs limit the end users in their expressiveness.

At its core, *ONYX* is a "restricted" NLI that provides the user low-level commands for frequent data visualization operations. Furthermore, we address the expressiveness of the "restricted" NLIs by providing end users with the ability to extend the current commands through ITL.

2.2 Interactive Task Learning

Interactive task learning as an approach for interactive end user development is increasingly leveraged for extending NLIs in GUI-based systems [1–3, 12]. While existing NLIs are programmed and

*e-mail: marcel.ruoff@kit.edu

Figure 1: User interface of the data visualization tool with integrated ITL-based NLI and active demonstration mode. A) GUI elements, such as filters and encodings, to adapt the visualization using direct manipulation, B) Visualization canvas, C) Demonstration panel that includes both the initial command on top with recognized parameters highlighted and the demonstrated action sequence below, D) NLI providing speech and text input as well as feedback.

designed by developers, ITL provides NLIs the ability to learn like humans do: from demonstrations and natural language instructions. However, most NLIs integrating ITL are solely using background knowledge after the demonstration process is finished to address ambiguities and the problem of generalizing the demonstrated interactions sequence. To target these challenges *during* the demonstration, ITL-based systems are increasingly utilizing a mixed-initiative approach. For example, APPINITE [13] requires users to describe the intended goal for each action during the demonstration process in their own words to clarify the goal of the task. Based on the natural language description of the goal, APPINITE tries to infer possible ambiguities and asks the users to clarify them. While this helps to address ambiguities during the demonstration process, end users are often reluctant to demonstrate lengthy sequences [1]. Compared to prior systems, *ONYX* utilizes background knowledge, such as the initial command and visualized data, to make the demonstration process more efficient by recommending possible actions and by helping the user creating data visualizations that follow existing guidelines.

3 *ONYX* System

We built *ONYX*, as shown in Figure 1, to investigate how to combine guidance and recommendations with an ITL-based approach to enable end user to efficiently enhance and adapt existing NLIs with natural interactions. Our approach differs from prior systems by attempting to leverage background knowledge, such as the initial command and visualized data, to effectively inform the demonstration process through recommendations and guidance.

3.1 Key Design Features

The interaction of the end user through the NLI is managed by a conversational agent. If the NLI fails to understand the utterance, the conversational agent first highlights the information it understood from the command, such as the chart type or other attributes as depicted in Figure 2. This is aimed to help the user understand whether the NLI failed because of a speech recognition error or due to the inability to determine a best response. This ensures that end users do not unnecessarily demonstrate functionalities that are already implemented in the NLI. If an end user determines that she wants to extend the functionality of the NLI after a breakdown, she can switch to the demonstration mode.

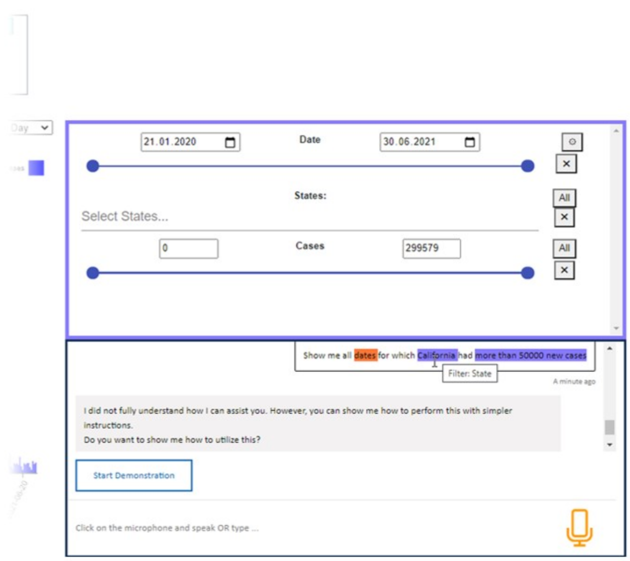

Figure 2: Example of ONYX providing end users feedback on what parameters have been recognized through highlighting the parameters and its background knowledge for the recognized parameters, such as California being recognized as a possible state filter.

End users can proactively start the demonstration process if they want to teach the system how to handle an original command correctly. For example, if an end user enters the command "Show me deaths for all states" and wants to see a bar chart but the system defaults to showing a map chart. Here, the end user can start the demonstration process proactively. After realizing that the best response does not match the command, end users can activate the demonstration mode through a natural language command. They will then get the current action sequence of the command displayed in 1 (C) and can adapt the action sequence by adapting either the action sequence or the visualization according to their specific needs using direct manipulation or natural language instructions.

In our current system, the end user is shown a display of the sequence of actions performed through direct manipulation of the GUI or natural language instructions. As depicted in Figure 1 (C), *ONYX* already combines similar actions, such as selecting multiple states, to provide end users a clear overview of the current sequence. Based on the impact of the actions on the data visualization tool, *ONYX* automatically recognizes whether the current and previous action complement or oppose each other or if they are independent and chooses appropriate language to reflect this to the end user.

Furthermore, after finalizing the demonstration, the ITL mechanism determines which parameters of the action sequence are variable and which are constant. For example, if the user demonstrates that "Show me deaths vs vaccinations for all states" means that deaths and vaccinations are selected as the y-axis of a bar chart with states of the USA as the x-axis, then the ITL mechanism determines based on the command that deaths, vaccinations and states are variables and bar chart is constant for this command.

3.1.1 Guidance

The two key challenges of enabling end users to extend NLIs on their own are their limited programming and data visualization expertise as well as difficulty of addressing the ambiguity of the demonstrated sequence. *ONYX* provides guidance for end users during the demonstration to address both challenges. First, ambiguity in the action sequence can lead to unintended training results from the demonstration. For example, if a demonstration would start with a visualization that shows deaths for all states, the command "Add California and Texas to the selected states" and "Show me deaths only for California and Texas" would have the same action sequence. However, if a different state was previously selected, then the commands would have different end results. To handle the ambiguity of this action sequence, *ONYX* provides follow-up questions during the demonstration. As depicted in Figure 1 (C), *ONYX* detects during the demonstration that there is a possible ambiguity and tries to resolve the ambiguity with the end user.

Second, the limited expertise of end users could lead to visualizations that violate guidelines for clarity and would impede the effective solving of the given task. For example, end users may be inclined to use a bar chart to compare the number of deaths due to COVID-19 with the number of fully vaccinated people. However, because of the large difference in the numbers this would violate Kelleher and Wagener's ninth guideline of effective data visualization, namely "keeping axis ranges as similar as possible to compare variables" [9]. In future versions, *ONYX* is planned to provide guidance during the demonstration using heuristics based on established visualization guidelines in order to prevent violations of the guidelines. This is especially helpful if the implications of the demonstration are not directly visible to the end user due to the generalization of the command and its parameters. For example, if end users demonstrate the comparison of two metrics with similar extent, such as fully vaccinated and partially vaccinated people, they may not notice the possible violation of a guideline if they later use two different metrics in the utilization of the learned command. Therefore, *ONYX* guides the user using feedback to address such possible violations during the demonstration process.

3.1.2 Recommendations

In previous ITL for NLIs, the demonstration is tedious because every action that needs to be part of the final action sequence of the command needs to be proposed by the end user [2, 12]. To assist the end user during the demonstration and to make the demonstration process more efficient, we are currently integrating a recommendation mechanism into *ONYX*. For example, if the user wants to demonstrate the command "Aggregate the deaths by month and select August and September" and the NLI already implemented the functionality "Aggregate **metric** by **aggregation parameter**", *ONYX* proposes the initial actions based on the command and the known functionality. Furthermore, if *ONYX* detects during the demonstration process that the action sequence and the command are similar to a previously demonstrated command from another user, it can propose the subsequent actions and provide an explanation why these actions have been proposed.

4 Formative Study including Participatory Design

In order to design the interactions for the end user and to seamlessly integrate the guidance and recommendations into this interaction, we plan to conduct a formative study. In particular, we plan to use our current instantiation of *ONYX* as a technical probe [8] to trigger reflections and to gain feedback and requirements from participants. Through this formative study we plan to answer the questions about how the recommendation and guidance features should best be provided to the end users. There are different alternative solutions, e.g., through GUI elements, through natural language by the conversational agent or both. Furthermore, we need to better understand how strictly to enforce data visualization guidelines for end users.

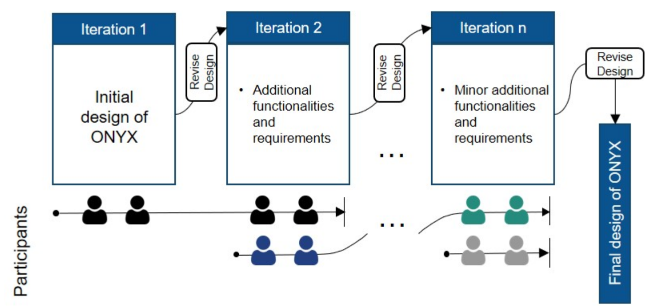

Figure 3: Iterations of the participatory design process

We utilize an iterative approach to incrementally improve *ONYX*'s following a participatory design approach in cooperation with potential users. Similar to [10], we will start the first iteration with two participants and our initial *ONYX* prototype described in this paper. Each participant will take part in an individual participatory design task that takes around 1 hour. During this participatory design task, the participants will be introduced to *ONYX* and its direct manipulation functionality. Subsequently, the participants will be asked to perform certain tasks with the data visualization tool and demonstrate how to handle commands that are currently not implemented in the NLI. The tasks will be provided to the users in form of jeopardy-style facts to achieve the benefits of mimicking realistic analytical findings and to engage the participants [20]. After 30 minutes, the participants will be asked to conduct an open-ended data exploration. Thus, participants will try out approaches that we may not have expected. During the complete session the participants will be encouraged to think aloud and will be recorded for later analysis. Finally, we will conduct a post-interview to inform the requirements and functionalities of the next iteration. We will show the revised version to the two initial participant, so we can receive feedback from more experienced end users, and will add two new participants so we can receive novel insights. For each following iteration the two experienced participants from the previous iteration are excluded and two new participants are included, as depicted in Figure 3. This procedure will continue until only minor adaptation to the functionality or minor requirements are elicited in the iteration.

5 Conclusion

ITL enables end users to enhance and adapt NLIs through the demonstration of natural interactions without the help of developers. *ONYX* is able to learn from end users through direct manipulation, natural language instructions, or through the combination of both. It provides to end users guidance and recommendations during the demonstration process to make the interaction more effective. In our upcoming formative study fwe aim to answer how to best integrate guidance and recommendation capabilities provided by the *ONYX* system into the interaction through a participatory design process.

References

[1] J. Allen, N. Chambers, G. Ferguson, L. Galescu, H. Jung, M. Swift, and W. Taysom. PLOW: A collaborative task learning agent. In *AAAI*, vol. 2, pp. 1514–1519, 2007.

[2] A. Azaria, J. Krishnamurthy, and T. M. Mitchell. Instructable intelligent personal agent. In *AAAI*, pp. 2681–2689, 2016.

[3] D. L. Chen and R. J. Mooney. Learning to interpret natural language navigation instructions from observations. In *AAAI*, vol. 1, pp. 859–865, 2011.

[4] K. Cox, R. E. Grinter, S. L. Hibino, L. J. Jagadeesan, and D. Mantilla. A multi-modal natural language interface to an information visualization environment. *International Journal of Speech Technology*, 4(3-4):297–314, 2001.

[5] E. Dong, H. Du, and L. Gardner. An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases*, 20(5):533–534, May 2020.

[6] E. Fast, B. Chen, J. Mendelsohn, J. Bassen, and M. S. Bernstein. Iris: A conversational agent for complex tasks. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1–12, 2018.

[7] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. DataTone. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*, pp. 489–500, 2015.

[8] H. Hutchinson, W. Mackay, B. Westerlund, B. B. Bederson, A. Druin, C. Plaisant, M. Beaudouin-Lafon, S. Conversy, H. Evans, H. Hansen, N. Roussel, B. Eiderbäck, S. Lindquist, and Y. Sundblad. Technology probes: Inspiring design for and with families. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 17–24, 2003.

[9] C. Kelleher and T. Wagener. Ten guidelines for effective data visualization in scientific publications. *Environmental Modelling and Software*, 26(6):822–827, June 2011.

[10] Y. Kim, Y. Choi, D. Kang, M. Lee, T. J. Nam, and A. Bianchi. HeyTeddy: Conversational test-driven development for physical computing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):1–21, December 2019.

[11] J. E. Laird, K. Gluck, J. Anderson, K. D. Forbus, O. C. Jenkins, C. Lebiere, D. Salvucci, M. Scheutz, A. Thomaz, G. Trafton, R. E. Wray, S. Mohan, and J. R. Kirk. Interactive Task Learning. *IEEE Intelligent Systems*, 32(4):6–21, 2017.

[12] T. J. J. Li, A. Azaria, and B. A. Myers. SUGILITE: Creating multimodal smartphone automation by demonstration. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 6038–6049, 2017.

[13] T. J.-J. Li, I. Labutov, X. N. Li, X. Zhang, W. Shi, W. Ding, T. M. Mitchell, and B. A. Myers. APPINITE: A Multi-Modal Interface for Specifying Data Descriptions in Programming by Demonstration Using Natural Language Instructions. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, vol. 2018-Octob, pp. 105–114. IEEE, 2018.

[14] A. Narechania, A. Srinivasan, and J. Stasko. NL4DV: A Toolkit for Generating Analytic Specifications for Data Visualization from Natural Language Queries. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):369–379, 2020.

[15] M. Ruoff and U. Gnewuch. Designing Multimodal BI&A Systems for Co-Located Team Interactions. *ECIS 2021 Research Papers*, June 2021.

[16] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*, UIST '16, pp. 365–377, 2016.

[17] V. Setlur, M. Tory, and A. Djalali. Inferencing underspecified natural language utterances in visual analysis. In *International Conference on Intelligent User Interfaces, Proceedings IUI*, vol. Part F1476, pp. 40–51, 2019.

[18] A. Srinivasan, M. Dontcheva, E. Adar, and S. Walker. Discovering natural language commands in multimodal interfaces. In *International Conference on Intelligent User Interfaces, Proceedings IUI*, pp. 661–672, 2019.

[19] A. Srinivasan, B. Lee, N. Henry Riche, S. M. Drucker, and K. Hinckley. InChorus: Designing Consistent Multimodal Interactions for Data Visualization on Tablet Devices. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1–13, 2020.

[20] A. Srinivasan and J. Stasko. How to ask what to say?: Strategies for evaluating natural language interfaces for data visualization. *IEEE Computer Graphics and Applications*, 40(4):96–103, 2020.

[21] Y. Sun, J. Leigh, A. Johnson, and S. Lee. Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. In *Lecture Notes in Computer Science*, vol. 6133 LNCS, pp. 184–195, 2010.