ONYX - User Interfaces for Assisting in Interactive Task Learning for Natural Language Interfaces of Data Visualization Tools

Marcel Ruoff
marcel.ruoff@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Brad A. Myers
bam@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, USA

Alexander Maedche
alexander.maedche@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

ABSTRACT

While natural language interfaces (NLIs) are increasingly utilized to simplify the interaction with data visualization tools, improving and adapting the NLIs to the individual needs of users still requires the support of developers. *ONYX* introduces an interactive task learning (ITL) based approach which enables NLIs to learn from users through natural interactions. Users can personalize the NLI with new commands using direct manipulation, known commands, or by combining both. To further support users during the training process, we derived two design goals for the user interface, namely providing suggestions based on sub-parts of the command and addressing ambiguities through follow-up questions and instantiated them in *ONYX*. In order to trigger reflections and gain feedback on possible design trade-offs of *ONYX* and the instantiated design goals, we performed a formative user study to understand how to successfully integrate the suggestions and follow-up question into the interaction.

CCS CONCEPTS

• **Human-centered computing** → *User interface programming*; **Natural language interfaces**; *Participatory design.*

KEYWORDS

Interactive task learning, Natural language interfaces, Participatory design, Data visualization tools

ACM Reference Format:

Marcel Ruoff, Brad A. Myers, and Alexander Maedche. 2022. ONYX - User Interfaces for Assisting in Interactive Task Learning for Natural Language Interfaces of Data Visualization Tools. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '22 Extended Abstracts), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3491101.3519793

1 INTRODUCTION

Extending data visualization tools through natural language interfaces (NLIs) is considered to be a promising approach to provide inexperienced users a more intuitive way of interacting [10, 16, 29, 31]. Typically, NLIs provide users the opportunity to articulate questions or natural language commands in their own words without the need to learn the underlying detailed user interface of the data visualization tool [32]. Based on the user's input, NLIs extract relevant information, such as the chart type, aggregation and required data fields, and translate them either into an adaptation of the displayed visualization or to create a new visualization [24, 29, 31, 33]. However, despite the continuous improvements in natural language processing, current NLIs still break down, for example, if the natural language command is not supported or is missing required information [38]. To address these breakdowns, informative prompts are used to notify users of a breakdown [6, 13, 31, 38] or users are requested to provide additional information to the NLI through subsequent commands [5, 8, 28, 29] and disambiguation widgets [10, 23]. While this enables users to better understand how they can adapt their behavior to the NLIs integrated in the data visualization tool, current NLIs lack the ability to learn from previous breakdowns and to subsequently adapt their functionality.

We are working on integrating interactive task learning (ITL) capabilities into NLIs for data visualization tools to provide users the ability to extend and adapt the functionality provided by NLIs to their individual needs. ITL is an emerging approach that allows users to interactively teach NLIs through natural interactions [14]. Specifically, through our approach, users can personalize existing natural language commands and extend NLIs with new natural language commands by interactively demonstrating their intended response either through direct manipulation, known natural language commands or a combination of both. Recent studies showed that users appreciate the ability to automate longer tasks and to provide aliases to existing commands through ITL [3, 17]. Furthermore, ITL can enable users to extend existing NLIs through new natural language commands by mapping them to a sequence of GUI interactions [27]. However, while current research shows how the results of the training process can be processed, for example, to generalize the natural language command to similar tasks (e.g. [17]), users still encounter difficulties during the training process and are in need of assistance [3]. To address this issue, we aim to contribute to the CHI community by investigating *how users can be assisted during the training process through suggestions and follow-up questions and identify possible design trade-offs in their instantiation.*

We conducted a participatory design process including a formative user study that explores how to successfully integrate suggestions and follow-up questions in an NLI with ITL capabilities. Based on the feedback from eight participants (so far), we designed *ONYX*, a data visualization tool integrating an NLI with ITL capabilities, and derive two design trade-offs in its instantiation. *ONYX* is named

CHI '22 Extended Abstracts, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9156-6/22/04…$15.00
https://doi.org/10.1145/3491101.3519793

after a gemstone and stands for: **O**ptimizing **N**atural language interfaces for **Y**our e**X**perience. Building on previous research, our results show possible pathways to provide users with assistance during the training process of NLIs with the ability of extending and adapting their natural language commands. Further research is needed to confirm the designs that best fit to the various pathways discovered in our work.

2 RELATED WORK

2.1 Natural Language Interfaces for Data Visualization Tools

Despite the increasing interest in NLIs for data visualization tools (e.g. [5, 8, 10, 20, 26, 29–31, 35]), NLIs are still vulnerable to breakdowns due to various potential errors in executing natural language commands. The errors can be classified into three categories [38]: (1) The natural language command is not supported, (2) the context is invalid or information is missing, or (3) the result does not match the user's expectations. The first and third error categories are often addressed through prompts, such as "Unable to process that command. Please try a different one" [31, p. 7] or "Sorry, I couldn't understand." [13, p. 6]. Even though this tells users that the NLI did not recognize their natural language command, it lacks the information how they could adapt their natural language command to accurately achieve their intended goal. In contrast, the second error type category is addressed by requesting additional information from users [5, 8, 28, 29] or providing disambiguation widgets [10, 23] to clarify ambiguities in their natural language command and to handle missing information. This provides users the information about what is missing in their initial natural language command for future interactions so they can directly address the error in their current natural language command. However, these current approaches have in common that the NLI is unable to learn from past clarifications of errors contained in the natural language command.

2.2 Interactive Task Learning

In contrast to humans, contemporary interactive systems, such as NLIs, are still "limited to a fixed set of innate or pre-programmed tasks" [14, p. 6]. However, it is unlikely that all tasks users want to perform with a certain system can be pre-programmed by their developers prior to their usage. Previous research has also shown that usage behavior can vary greatly from user to user, highlighting the issue of pre-programmed fixed systems [39]. Therefore, ITL takes inspiration from how humans teach other humans new concepts and tasks to enable systems to learn from their users to extend the tasks users are able to perform [14].

Laird et al. [14] derived desiderata for complete and comprehensive ITL capabilities, namely providing efficient and effective task learning, task performance and interaction during the training. Previous research investigating the aspect of task learning explored how to generalize a learned task to enable performing similar tasks (e.g. [1, 3, 17]). The major limitation of these systems is that they do not address the ambiguities that arise due to the utilization of a command in different contexts. This is especially crucial in application contexts, such as data visualization tools, where even the same action sequence can result in completely different visualizations based on the initial configuration of the tool. For example, Appelgren and Lascarides [2] aim to address this issue by utilizing the structure of the natural language command after the training process. However, in this approach, users are unable to handle ambiguities that are incorrectly addressed by the system. APPINITE [18], on the other hand, 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 training process, the interaction may be impaired since users are often reluctant to demonstrate lengthy sequences [1].

3 ONYX - AN INTERACTIVE TASK LEARNING AGENT

3.1 Design Goals

To address the gaps identified in existing work of NLIs integrated in data visualization tools and ITL, we distilled the following two key design goals (**DGs**) for assisting users in ITL:

DG1. Derive suggestions based on sub-parts of the articulated natural language command and background knowledge of the system.

Users utilizing an ITL agent to demonstrate the meaning of the natural language command often struggle to initially understand what kind of concepts the system understands [3, 15, 19] and strive for an efficient interaction [14]. To address this issue, we propose that the system should break down the original natural language command into its sub-parts based on the semantic structure of the command to analyze if the sub-parts are similar to known commands or whether they describe actions to be performed with the GUI elements. If the system identifies existing knowledge about sub-parts of the command, such as a similarity to known commands, it should provide this knowledge back to users in form of suggestions for the next possible actions.

DG2. Address ambiguities through follow-up questions during the training process.

ITL-based systems often struggle to learn new natural language commands based on only one example due to ambiguous user actions [18]. In *ONYX*, this occurs since identical actions of users can have multiple meanings based on the current configuration of the data visualization tool. To address this issue, we propose that the ITL-based system should address these ambiguities through follow-up questions during the demonstration process. Specifically, the system should check after each interaction if this interaction has multiple interpretations based on the configuration of the data visualization tool or previous interactions. In this way it can derive in collaboration with users how to perform the actions provided in different configurations of the data visualization tool.

3.2 User Interface

As depicted in Figure 1, the current version of the data visualization tool integrating *ONYX* consists of A) a visualization canvas, B) GUI elements, such as buttons for the visualization type and encodings, to adapt the visualization using direct manipulation, C) a filter pane to provide constraints on the data retrieved to display the visualization, and D) an NLI providing text input and feedback to users. We

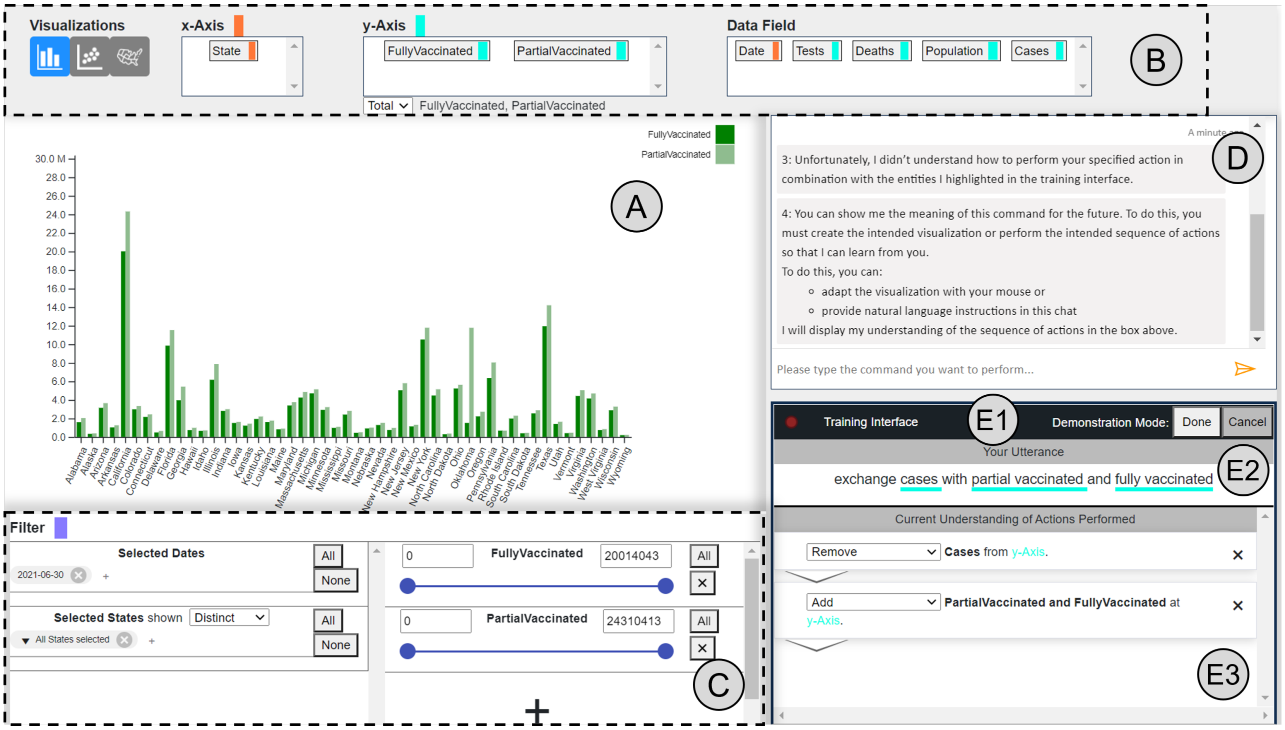

Figure 1: User Interface of the Data Visualization Tool with integrated ITL-based NLI during the Training Process. A) Visualization canvas, B) Buttons to adapt visualization and encodings, C) Filter pane to provide constraints, D) NLI providing text input and feedback, E1) Signalling light and buttons to finish the demonstration mode, E2) Utterance to be demonstrated, E3) Actions performed by the user.

provided equivalence between the interaction with GUI elements and the NLI [4], to provide users with a nonmodal dialog with the tool and the ability to switch between the interaction modalities at any point. Hence, users are able to adapt the visualization, its constraints and its encodings, such as adding a data field at the x-Axis, using either the NLI or the GUI elements.

Furthermore, if the training process is invoked either by the user proactively due to an incorrect response to a natural language command or by *ONYX* reactively after a breakdown, an additional training interface is shown in the overall user interface. The training interface consists of E1) a header with signalling icon and buttons to finish the demonstration mode, E2) display of the user's utterance with highlighted entities, and E3) the current understanding of the actions performed by users during the training process. The display of the current understanding of actions performed by users is inspired by block-based programming tools like Blockly [9] and Scratch [21].

4 PARTICIPATORY DESIGN PROCESS: A FORMATIVE STUDY

4.1 Methodology

In order to design the interactions for users as well as to demonstrate and refine the instantiation of the design goals, we conducted a formative study. In particular, we use our current instantiation of *ONYX* as a technical probe [11] to trigger reflections and to gain feedback and requirements from participants. Through this formative study, we aim to answer the questions about how the design goals should best be addressed and what design trade-offs arise in their realization. Furthermore, this enables us to generally improve the usability, and to design consistent interactions with the tool through iterative feedback from participants. Therefore, we utilize a high-fidelity prototype to explore how users react to the assistance provided based on their own natural language commands unrestricted by possible constraints and to inspect how well our recognition of ambiguities and the resulting suggestions work.

Procedure. We utilize an iterative approach to incrementally improve *ONYX*'s design in cooperation with the study participants. Similar to Kim et al. [12], we start the first iteration with two participants and our initial prototype. Each participant takes part in an individual participatory design task that takes around 1 hour. During this participatory design task, the participants are first introduced to the goal of the study, the overall procedure, and then the informed consent is requested from the participant for each session of the study. Second, the participants are trained on how to interact with the data visualization tool through direct manipulation using the mouse and keyboard, since the goal of the study is to improve the natural language interface and the demonstration of new natural language commands. Additionally, the COVID-19 dataset that is integrated in the data visualization tool is introduced to the participants.

Third, during the main part, the participants are asked to perform certain tasks with the data visualization tool and demonstrate how to handle natural language commands that are currently not implemented in the NLI. The tasks are provided as target replication tasks to achieve the benefits of evaluating low-level operations and to minimize the risk of phrasing bias [34]. Specifically, the participants are handed four target visualizations that they should each achieve through a single natural language command. We deliberately chose adaptations to the visualizations that the NLI is not yet capable of. Therefore, participants are assisted by *ONYX* during the demonstration of these new natural language commands. Since the main drawback of target replication tasks is "that the target state or sequences may not mimic a real-world analysis scenario" [34, p. 101] we subsequently ask participants to conduct an open-ended data exploration. They are hence asked what visualization they are interested in and requested to utilize direct manipulation or the NLI to try to create these visualizations. This enables us to assess the overall system features and usability with a more natural workflow and assess the performance of *ONYX* in assisting users in the demonstration of natural language commands with a wider range [34]. During the complete session the participants are encouraged to think aloud. Their voice and the data visualization tool screens are recorded for later analysis. Finally, we conduct a post-interview to inform the requirements and functionalities of the next iteration. The interviews have been conducted in a non-directive manner to allow for an in-depth focus on the the users' interactions and experiences with *ONYX*.

A few days later, after revising the current artifact, we show the new version to the two initial participants, so we can receive feedback from more experienced users, and also add two new participants so we can receive novices' insights. For each following iteration, we collect feedback from four participants: two experienced participants from the previous iteration and two new participants. This procedure continues until only minor adaptations to the functionality or minor requirements are elicited in the iteration.

Participants. So far we have recruited eight participants (6 male; 2 female) from a panel of students. The participants had an average age of 28.4 (SD = 13.5) and were either enrolled in their undergrad (N = 4) or in their graduate studies (N = 3) and one participant completed the graduate degree. To reach consensus for the instantiation of the design goals in *ONYX*, we estimate that one additional iteration is required. After completing the study, which consists of two sessions for each participant, they were paid 30 US dollars for their participation.

4.2 Design Trade-Offs

In this section we describe the feedback from the participants of the formative study and the derived design trade-offs based on an inductive qualitative analysis [36] of the recordings and notes. For each design trade-off we first describe the potential designs and then further discuss the participants' feedback to these designs and potential trade-offs.

Timing of Assistance in the Training Process.

Providing assistance to users during the demonstration of natural language commands through follow-up questions and suggestions is a key element of *ONYX*. However, currently it is unclear *when* to provide this assistance. The assistance could either be provided (1) *synchronously* with the interaction triggering the assistance or (2) *asynchronously*.

In a *synchronous* approach, assistance would be provided directly after a state is reached or action performed by users that requires assistance. Suggestions (DG1) for possible actions based on the articulated natural language command would not be provided all at once, but in connection with the sub-part of the natural language command that users are currently demonstrating. Specifically, in the beginning of the interaction, users are provided with suggestions for the main sub-part of the natural language command that is derived based on the sentence structure. The suggestions are added as blocks in the current understanding of actions performed (Figure 1 E3). New suggestions are only provided after all unknown concepts, such as entities, associated with the current sub-part are included in the actions performed by users. Follow-up questions (DG2) would be directly provided after an ambiguity arises. Hence, users would be aware of the mapping between the action invoking the follow-up question and the follow-up question itself. Since *ONYX* provides a nonmodal dialog, users are able to continue with their demonstration without utilizing suggestions or addressing the follow-up questions.

In an *asynchronous* approach, assistance would be provided at a time that would minimize the interruption of users' interaction flow. All suggestions that the tool derived based on the articulated natural language command and its background knowledge would be provided at the beginning of the training process as a collection of blocks in the current understanding of actions performed. Follow-up questions would be accumulated during the complete demonstration of the natural language command. Only at the end of users' turns would the tool ask the follow-up questions to clarify remaining ambiguities. The tool would therefore need to maintain a complete record of unaddressed ambiguities and remove ambiguities if they are already addressed by the user's actions. Furthermore, the tool would need to infer when users have finished their turn. In NLIs, users usually signal the end of their turn by terminating their natural language command, e.g., by hitting the enter keyboard key. However, in direct manipulation, turns often consist of multiple interactions with the GUI elements. Hence, in a tool enabling interaction with both an NLI and GUI, users either need to explicitly indicate the end of their turn or follow-up questions can be provided at the end of the overall training process.

In our initial instantiation of the design goals, we utilized a *synchronous* approach for the assistance. After participants receiving a suggestion in the form of blocks added to the current understanding of actions (at Figure 1 E3) and an explanation provided in the NLI (Figure 1 D), participants did notice both parts of the suggestion due to their spatial proximity. Participants expressed that connecting the suggestions to a sub-part of the natural language command and highlighting the unknown concepts helps them understand which parts of the natural language command they still need to demonstrate. Regarding the follow-up questions, participants expressed satisfaction with the clear mapping between the follow-up questions and why they arose due to the temporal connection. After they received the follow-up question, they were able to understand the question and the reason for the request for clarification. However, in some cases, because of the nonmodal dialogs, participants did not notice that the NLI provided them with a follow-up question and continued their interaction without addressing the ambiguity. To address this issue, participants proposed that either their attention could be guided to the follow-up question by a pop up on top of the visualization canvas or that the interaction with the data visualization tool could be restricted to a modal dialog until the follow-up question was answered. However, most participants either implicitly or explicitly expressed their preference for the current nonmodal dialog due to the freedom of choice it provides to them.

To better understand the design trade-offs with regards to this issue, we discussed with participants the possibility of providing assistance *asynchronously* only at the beginning and end of the demonstrations, and its implications. Participants stated that providing them the suggestions all at once at the beginning of the training process could enable them to perform the demonstration in a shorter time. They reasoned that because they can perceive the complete current understanding of the natural language command all at once they could better infer what actions need to be adapted and which need to be added. However, some participants were already overwhelmed by suggestions that consisted of more than one action. Therefore, receiving the suggested actions all at once could overwhelm users of *ONYX* and lead to mistakes because of missing understanding and misplaced trust in the correctness of the suggestions. Furthermore, to provide follow-up questions *asynchronously*, the tool would need to ensure that users understand why the follow-up question is asked and based on which action the associated ambiguity arose without a temporal connection between these elements. To substitute the temporal connection, a possible alternative for ITL-based NLIs utilizing an *asynchronous* approach can be to provide the connection as *reactive feedback*. For example, when users hover their mouse over a follow-up question provided in the NLI, the system can highlight the associated block in the current understanding of actions performed.

Modality of Follow-up Questions.

A key challenge in systems providing more than one modality as output (i.e., GUI and NLI) is to decide how to provide information back to users [7, 22, 25]. Developers are encouraged to maximize the advantages of each modality to reduce a user's cognitive load during the interaction with the system [25]. Since *ONYX* provides users the means to interact with it through GUI elements and the NLI, *ONYX* can either provide (1) *visual* or (2) *textual follow-up questions* to users to address ambiguities.

Since the current understanding of actions performed is provided in block-based form, *visual follow-up questions* can be provided to users as a choice between two blocks depicting *ONYX*'s different possible interpretations of the action performed (see Figure 2 Visual). Users are then able to compare the differences in the blocks and decide which block fits better to their intended goal. These visual cues can help users in addressing ambiguities by mapping internal cognitive process to their external representation in the tool [37].

Textual follow-up questions are provided to the users in the NLI, as depicted in Figure 2 Textual. Due the expressiveness of natural language, the tool can express the difference between the interpretations of the action performed in a concise way. Specifically, *ONYX* analyzes in which configuration different results would arise due to

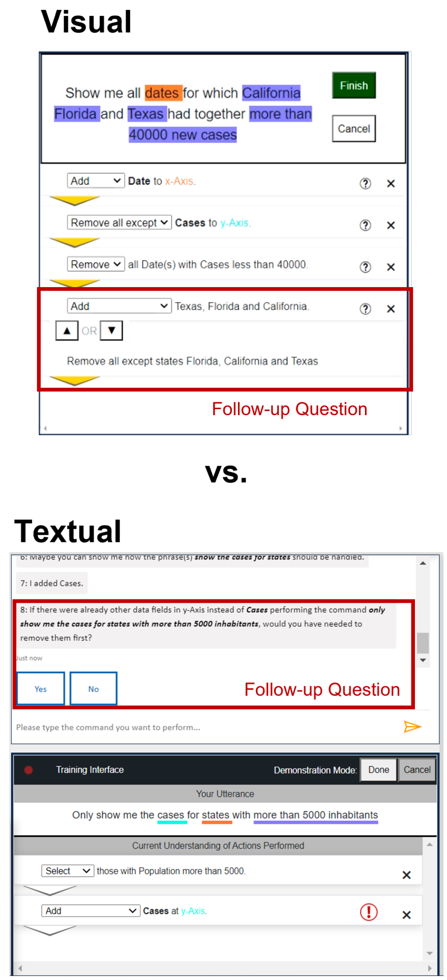

Figure 2: Comparison of the Provision of Visual and Textual Follow-up Questions.

the ambiguity. The tool then prompts users to specify which result they would expect in this hypothetical configuration with a Yes/No question. Based on the answer, the tool adapts the associated block in the current understanding of actions performed.

To investigate the design trade-offs between *visual* and *textual follow-up questions*, we instantiated both designs during the formative study. Participants were randomly either first provided with the *visual* and then with the *textual* follow-up questions or in the opposite order to understand their subsequent behavior and thought processes.

Users' behavior and feedback showed that they expected new blocks to appear in the current understanding of actions performed and usually checked whether the new blocks appearing in the interface represented their intended goal. Therefore, they quickly noticed the *visual* follow-up questions prompting them to address the ambiguity that arose due to their last action. Participants further stated that the decision between two blocks representing the different interpretations of the action by the tool further enabled them to make fast decisions without disturbing their interaction flow too much. However, when participants were asked during the formative study to explain *why* the ambiguity arose they were not able to identify the reason based on the *visual follow-up questions*. Furthermore, participants frequently chose the wrong option based on their overall goal of the functionality associated with the natural language command due to this missing understanding.

In contrast, after receiving *textual* follow-up questions participants had a clear understanding why the ambiguity arose based on the abstraction provided in natural language in the NLI. Participants stated that based on this understanding they were easily able to decide the appropriate answer for the Yes/No question. This was further supported by their behavior, since participants were deciding more accurately between the different interpretations of their action compared to the *visual follow-up questions*. However, participants expressed that they did not expect new prompts in the NLI if they used direct manipulation for the training. Therefore, they sometimes overlooked the *textual follow-up questions* and continued the training process without addressing the ambiguity. Participants suggested that this issue can be addressed by providing a pop-up in the visualization canvas, which is the center of attention of users, that quickly notes that the user received a follow-up question in the NLI. Another possibility to address this issue would be to graphically connect the GUI element that users utilized for their demonstration and the provided follow-up question in the NLI using a line to guide the attention of users from their current point of interaction to the NLI.

5 FUTURE WORK AND CONCLUSION

In this paper, we introduced two design goals for assisting users in ITL for NLIs of data visualization tools that aim on addressing current gaps in existing work, namely addressing ambiguities and providing an efficient interaction during the training process. We have created an instantiation of the two design goals in *ONYX*. We used *ONYX* in a formative study to identify potential design trade-offs in the instantiation of these two design goals and describe participants viewpoints on the design trade-offs.

Even though our formative study already shows promising results for assisting users during the training process, the effects on the accuracy of the training and the efficiency of the interactions need to be rigorously evaluated through further qualitative and quantitative studies. Building on the results of our formative study, qualitative studies could elaborate under which circumstances designers should choose one of the various design trade-offs we identified. Furthermore, quantitative studies could confirm or reject initial hypothesis we derived in our formative study, such as that addressing ambiguities during the training process through follow-up questions would lead to more accurate demonstrations. Finally, we believe that our design goals and their design trade-offs provide future research interesting perspectives in investigating the assistance of users during the training process of NLIs with ITL-capabilities beyond the context of data visualization tools.

REFERENCES

[1] James Allen, Nathanael Chambers, George Ferguson, Lucian Galescu, Hyuckchul Jung, Mary Swift, and William Taysom. 2007. PLOW: A collaborative task learning agent. In *Proceedings of the National Conference on Artificial Intelligence*, Vol. 2. AAAI Press, 1514–1519. https://doi.org/10.5555/1619797.1619888

[2] Mattias Appelgren and Alex Lascarides. 2020. Interactive task learning via embodied corrective feedback. *Autonomous Agents and Multi-Agent Systems* 34, 2 (10 2020), 1–45. https://doi.org/10.1007/S10458-020-09481-8/TABLES/6

[3] Merav Chkroun and Amos Azaria. 2019. LIA: A Virtual Assistant that Can Be Taught New Commands by Speech. *International Journal of Human-Computer Interaction* 35, 17 (2019), 1596–1607. https://doi.org/10.1080/10447318.2018.1557972

[4] Joëlle Coutaz, Laurence Nigay, Daniel Salber, Ann Blandford, Jon May, and Richard M. Young. 1995. Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The Care Properties. In *Human—Computer Interaction*. 115–120. https://doi.org/10.1007/978-1-5041-2896-4{_}19

[5] Kenneth Cox, Rebecca E. Grinter, Stacie L. Hibino, Lalita Jategaonkar Jagadeesan, and David Mantilla. 2001. A multi-modal natural language interface to an information visualization environment. *International Journal of Speech Technology* 4, 3-4 (2001), 297–314. https://doi.org/10.1023/A:1011368926479

[6] Kedar Dhamdhere, Kevin S. McCurley, Ralfi Nahmias, Mukund Sundararajan, and Qiqi Yan. 2017. Analyza. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, New York, NY, USA, 493–504. https://doi.org/10.1145/3025171.3025227

[7] Bruno Dumas, Denis Lalanne, and Sharon Oviatt. 2009. Multimodal Interfaces: A Survey of Principles, Models and Frameworks. In *Human Machine Interaction: Research Results of the MMI Program*, Denis Lalanne and Jürg Kohlas (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 3–26. https://doi.org/10.1007/978-3-642-00437-7{_}1

[8] Ethan Fast, Binbin Chen, Julia Mendelsohn, Jonathan Bassen, and Michael S. Bernstein. 2018. Iris. In *Conference on Human Factors in Computing Systems - Proceedings*. ACM Press, New York, New York, USA, 1–12. https://doi.org/10.1145/3173574.3174047

[9] Neil Fraser. 2015. Ten things we've learned from Blockly. In *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*. IEEE, 49–50. https://doi.org/10.1109/BLOCKS.2015.7369000

[10] Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G. Karahalios. 2015. DataTone. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*. ACM Press, New York, New York, USA, 489–500. https://doi.org/10.1145/2807442.2807478

[11] Hilary Hutchinson, Wendy Mackay, Bosse Westerlund, Benjamin B Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, Björn Eiderbäck, Sinna Lindquist, and Yngve Sundblad. 2003. Technology probes: Inspiring design for and with families. In *Conference on Human Factors in Computing Systems - Proceedings*. 17–24.

[12] Yoonji Kim, Youngkyung Choi, Daye Kang, Minkyeong Lee, Tek Jin Nam, and Andrea Bianchi. 2019. HeyTeddy: Conversational test-driven development for physical computing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (12 2019), 1–21. https://doi.org/10.1145/3369838

[13] Young Ho Kim, Bongshin Lee, Arjun Srinivasan, and Eun Kyoung Choe. 2021. Data@hand: Fostering visual exploration of personal data on smartphones leveraging speech and touch interaction. In *Conference on Human Factors in Computing Systems - Proceedings*. ACM, New York, NY, USA, 1–17. https://doi.org/10.1145/3411764.3445421

[14] John E Laird, Kevin Gluck, John Anderson, Kenneth D Forbus, Odest Chadwicke Jenkins, Christian Lebiere, Dario Salvucci, Matthias Scheutz, Andrea Thomaz, Greg Trafton, Robert E Wray, Shiwali Mohan, and James R Kirk. 2017. Interactive Task Learning. *IEEE Intelligent Systems* 32, 4 (2017), 6–21. https://doi.org/10.1109/MIS.2017.3121552

[15] Tessa Lau. 2009. Why Programming-By-Demonstration Systems Fail: Lessons Learned for Usable AI. *AI Magazine* 30, 4 (10 2009), 65–65. https://doi.org/10.1609/AIMAG.V30I4.2262

[16] Bongshin Lee, Arjun Srinivasan, Petra Isenberg, and John Stasko. 2021. Post-wimp interaction for information visualization. *Foundations and Trends in Human-Computer Interaction* 14, 1 (2021), 1–95. https://doi.org/10.1561/1100000081

[17] Toby Jia Jun Li, Amos Azaria, and Brad A. Myers. 2017. SUGILITE: Creating multimodal smartphone automation by demonstration. In *Conference on Human Factors in Computing Systems - Proceedings*, Vol. 2017-May. ACM, New York, NY, USA, 6038–6049. https://doi.org/10.1145/3025453.3025483

[18] Toby Jia-Jun Li, Igor Labutov, Xiaohan Nancy Li, Xiaoyi Zhang, Wenze Shi, Wanling Ding, Tom M. Mitchell, and Brad A. Myers. 2018. 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. IEEE, 105–114. https://doi.org/10.1109/VLHCC.2018.8506506

[19] Toby Jia-Jun Li, Marissa Radensky, Justin Jia, Kirielle Singarajah, Tom M. Mitchell, and Brad A. Myers. 2019. PUMICE: A Multi-Modal Agent that Learns Concepts and Conditionals from Natural Language and Demonstrations. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 577–589. https://doi.org/10.1145/3332165.3347899

[20] Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. 2022. Natural Language to Visualization by Neural Machine Translation. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (1 2022), 217–226. https://doi.org/10.1109/TVCG.2021.3114848

[21] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. The scratch programming language and environment. *ACM Transactions on Computing Education* 10, 4 (2010). https://doi.org/10.1145/1868358.1868363

[22] Richard E. Mayer and Roxana Moreno. 2003. Nine Ways to Reduce Cognitive Load in Multimedia Learning. *Educational Psychologist* 38, 1 (3 2003), 43–52. https://doi.org/10.1207/S15326985EP3801{_}6

[23] Arpit Narechania, Adam Fourney, Bongshin Lee, and Gonzalo Ramos. 2021. DIY: Assessing the Correctness of Natural Language to SQL Systems. In *26th International Conference on Intelligent User Interfaces*. ACM, New York, NY, USA, 597–607. https://doi.org/10.1145/3397481.3450667

[24] Arpit Narechania, Arjun Srinivasan, and John Stasko. 2021. NL4DV: A Toolkit for Generating Analytic Specifications for Data Visualization from Natural Language Queries. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2 2021), 369–379. https://doi.org/10.1109/TVCG.2020.3030378

[25] Leah M. Reeves, Jennifer Lai, James A. Larson, Sharon Oviatt, T. S. Balaji, Stéphanie Buisine, Penny Collings, Phil Cohen, Ben Kraal, Jean Claude Martin, Michael McTear, T. V. Raman, Kay M. Stanney, Hui Su, and Qian Ying Wang. 2004. Guidelines for multimodal user interface design. *Commun. ACM* 47, 1 (1 2004), 57–59. https://doi.org/10.1145/962081.962106

[26] Marcel Ruoff and Ulrich Gnewuch. 2021. Designing Multimodal BI&A Systems for Co-Located Team Interactions. *ECIS 2021 Research Papers* (6 2021). https://aisel.aisnet.org/ecis2021_rp/113

[27] Ritam Jyoti Sarmah, Yunpeng Ding, Di Wang, Cheuk Yin Phipson Lee, Toby Jia-Jun Li, and Xiang Anthony Chen. 2020. Geno. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 1169–1181. https://doi.org/10.1145/3379337.3415848

[28] Jaydeep Sen, Greg Stager, Chuan Lei, Fatma Ozcan, Ashish Mittal, Diptikalyan Saha, Abdul Quamar, Manasa Jammi, and Karthik Sankaranarayanan. 2019. Natural language querying of complex business intelligence queries. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 1997–2000. https://doi.org/10.1145/3299869.3320248

[29] Vidya Setlur, Sarah E Battersby, Melanie Tory, Rich Gossweiler, and Angel X Chang. 2016. Eviza. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 365–377. https://doi.org/10.1145/2984511.2984588

[30] Vidya Setlur, Melanie Tory, and Alex Djalali. 2019. Inferencing underspecified natural language utterances in visual analysis. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, Vol. Part F1476. ACM, New York, NY, USA, 40–51. https://doi.org/10.1145/3301275.3302270

[31] Arjun Srinivasan, Bongshin Lee, Nathalie Henry Riche, Steven M. Drucker, and Ken Hinckley. 2020. InChorus: Designing Consistent Multimodal Interactions for Data Visualization on Tablet Devices. In *Conference on Human Factors in Computing Systems - Proceedings*. ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376782

[32] Arjun Srinivasan, Nikhila Nyapathy, Bongshin Lee, Steven M. Drucker, and John Stasko. 2021. Collecting and Characterizing Natural Language Utterances for Specifying Data Visualizations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–10. https://doi.org/10.1145/3411764.3445400

[33] Arjun Srinivasan and John Stasko. 2018. Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 511–521. https://doi.org/10.1109/TVCG.2017.2745219

[34] Arjun Srinivasan and John Stasko. 2020. How to ask what to say?: Strategies for evaluating natural language interfaces for data visualization. *IEEE Computer Graphics and Applications* 40, 4 (2020), 96–103. https://doi.org/10.1109/MCG.2020.2986902

[35] Yiwen Sun, Jason Leigh, Andrew Johnson, and Sangyoon Lee. 2010. Articulate: A Semi-automated Model for Translating Natural Language Queries into Meaningful Visualizations. In *Lecture Notes in Computer Science*. Vol. 6133 LNCS. 184–195. https://doi.org/10.1007/978-3-642-13544-6{_}18

[36] David R. Thomas. 2006. A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation* 27, 2 (2006), 237–246. https://doi.org/10.1177/1098214005283748

[37] David Weintrop. 2017. Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms. *ACM Trans. Comput. Educ* 18, 3 (2017). https://doi.org/10.1145/3089799

[38] Bowen Yu and Claudio T. Silva. 2020. FlowSense: A Natural Language Interface for Visual Data Exploration within a Dataflow System. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (1 2020), 1–11. https://doi.org/10.1109/TVCG.2019.2934668

[39] Sha Zhao, Julian Ramos, Jianrong Tao, Ziwen Jiang, Shijian Li, Zhaohui Wu, Gang Pan, and Anind K Dey. 2016. Discovering different kinds of smartphone users through their application usage behaviors. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, New York, NY, USA, 498–509. https://doi.org/10.1145/2971648.2971696