Towards Interactively Contextualizing Natural Language Input in Data Visualization Tools

Marcel Ruoff*
Karlsruhe Institute of Technology

Brad A. Myers†
Carnegie Mellon University

Alexander Maedche‡
Karlsruhe Institute of Technology

Abstract

While natural language interfaces (NLIs) integrated in data visualization tools are an opportunity to facilitate an analytical flow through conversation, they still exhibit unexpected system behavior due to ambiguities in the conversation between users and the data visualization tool. In our initial natural language (NL) elicitation study, we found that for over 70% of NL inputs that exhibited ambiguities, the goal of users could be clarified through contextual conditions, such as the current data fields selected in the data visualization. However, there are numerous challenges in deriving these contextual conditions by developers upfront or automatically by the system during actual use. Instead, we propose *ContexIT*, a mixed-initiative system that is able to continuously learn the contextual conditions for NL inputs based on the visualization state and clarifications from the actual users.

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

1 Introduction

Enhancing data visualization tools through natural language interfaces (**NLIs**) has been identified as a promising approach to facilitate an analytical flow through a conversation with the visualization [15, 17]. Complementary advantages from both the NLI and the graphical user interface (**GUI**) of the data visualization tool can be combined. With NLIs, users do not have to translate their information needs into an action performed in the data visualization tool but can utilize natural phrasings as part of their natural language (**NL**) input. GUIs on the other hand provide users with control over the system and display all of the details of the current visualization state. For this reason, users can benefit by interacting with the data visualization tool in a conversational style (using the *NLI*) while remaining in control of the system during unexpected system behavior (using the *GUI*).

In current NLIs, unexpected system behavior still occurs since conversations between users and a data visualization tool exhibit ambiguities [3, 18, 20], similar to human-human interaction [12]. These ambiguities may lead to misinterpretations by the NLI and can either result in incorrectly recognized and utilized attributes or even in the NLI having a different understanding of the user's overall goal. The importance of clarifying the users' goal is supported by our initial NL input elicitation study in which 12.9% of all elicited NL inputs are ambiguous with regards to the goal of the user. For example, in our data visualization tool, the NL input 💬 Deselect the Energy Types was utilized by users to both i) remove the data field *Energy Type* from the x-axis as well as to ii) remove an associated filter from the filter pane while retaining the data field in the visualization.

While these ambiguities can be addressed using the GUI, taking control of the system during a conversation with a data visualization tool can interrupt the analytical flow. Hence, the frequency with which users have to intervene must be reduced. However, the key challenge of clarifying these ambiguities remains as the tradeoff between interrupting the analytical flow through requiring user involvement versus unexpected system behavior. Consequently, previous systems aim to reduce how often ambiguities arise by integrating the linguistic context into the interpretation [15] or by pre-defining conditional statements for the actions to be performed by the NLI based on the context (hereafter referred to as *contextual conditions*) [20]. While these efforts have been able to improve the overall experience of users, several opportunities remain to enhance the clarification of ambiguities.

Unexploited Context: Context is crucial for clarifying ambiguities as humans assume in conversations that the knowledge they possess, such as the current state of the data visualization tool, is shared [12]. In data visualization tools, which exhibit by nature visual elements, the context includes previous interactions as well as the visualization state [20]. However, existing studies with a working artifact mainly focus on the linguistic context by utilizing previous NL inputs and only utilize the visual state to augment underspecified NL inputs [15]. This misses crucial information needed to clarify additional ambiguities since both the current elements visualized as well as previous interactions performed through direct manipulation remain mostly unexploited. For example, utilizing the contextual condition of whether filters have been specified for *Energy Types* would be able to clarify the ambiguity in the previous example in our initial NL input elicitation study.

Limited Learning from Past Interactions: It is difficult, if not impossible, to determine in advance the goal of every possible NL input that a user might use [20]. The same goal can be described in numerous ways [18]. At the same time, NL inputs can have multiple goals even for the same user since identical NL inputs might be different for the user based on additional knowledge, such as the context. Thus, it is crucial for the NLI to interactively learn from past interactions to differentiate in future occurrences between the different goals to appropriately select the correct actions without continuously interrupting users. While some NLIs for data visualization tools address the first difficulty of unknown NL inputs [13], current NLIs are still unable to continuously learn NL inputs with multiple meanings and how to distinguish between them.

In this paper, we first describe the results of our initial NL input elicitation study demonstrating how context can be utilized to clarify ambiguities. We highlight that 12.9% of the elicited NL inputs are ambiguous with regards to the goal of users and detail how we were able to clarify 71.4% of these ambiguities through contextual conditions. Based on these results, we identified five challenges of deriving conditional constraints without user involvement. Subsequently, we propose a design for a new tool, *ContexIT*, which aims to address these challenges by interactively learning to contextualize NL inputs to clarify ambiguities. More specifically, *ContexIT* learns and generalizes contextual conditions to appropriately select between multiple goals of an utilized NL input based on

*e-mail: marcel.ruoff@kit.edu
†e-mail: bam@cs.cmu.edu
‡e-mail: alexander.maedche@kit.edu

past clarifications of ambiguities in similar NL inputs.

2 Related Work

2.1 Natural Language Interfaces for Data Visualization Tools

NLIs have been increasingly utilized to assist users in analyzing and exploring data in data visualization tools (e.g., [1–3, 5, 8, 15–17, 19]). Previous studies showed that extending data visualization tools through NLIs particularly helps users perform tasks that would otherwise require multiple adjustments in the GUI [15] or complex filter settings [5]. However, while the variety of use cases has grown over the last decades, a large gulf between user expectations and the capabilities of NLIs still exists [20].

The major challenge of NLIs is that users expect the NLI to both understand their intent as well as the current context [20]. However, most systems focus on understanding only the intent and involve users to clarify ambiguities through widgets when they arise because they did not take into account the context [3, 4, 16]. In contrast, Eviza [15] aimed to address this challenge by utilizing the linguistic context of the previous NL input and user settings to clarify ambiguities by augmenting the current NL input. While this enabled Eviza to clarify some ambiguities of underspecified NL inputs, Eviza was not yet able to clarify NL inputs with multiple differing meanings.

2.2 Natural Language Interfaces with Learning Capabilities

Since expectations for an NLI differ between users and developers, researchers are looking more and more into enabling the NLI to interactively *learn* from users during their interactions [6, 7, 9, 11, 13, 21]. Through these efforts, NLIs are now able to interactively learn concepts, such as *hot*, *cold* or *born in* [7, 9], to map actions in the system to previously unknown NL inputs [6, 13, 21], and to refine the understanding of existing NL inputs [3].

However, currently the NLIs either learn a too narrow understanding of the NL input based on the specific attributes included in the NL input during learning [3] or learn a too abstract understanding by generalizing the attributes included to allow for all valid attributes to be utilized in the learned NL input [6, 13, 21]. This leads to a tradeoff between the need to involve users more frequently but be precise (*narrow understanding*), and to involve users occasionally but risk unexpected system behavior (*generalized understanding*). Therefore, we aim to utilize a generalized understanding of the NL input but make it more precise through an understanding of the current context (i.e., visualization state) to reduce unexpected system behavior.

3 Identifying and Clarifying Ambiguities in NL Inputs through Context

We conducted an initial NL elicitation study to understand how end users verbally instruct the NLI to perform various actions in the data visualization tool and to what extent the ambiguities could be clarified through the current context. We draw inspiration from a previous study on eliciting NL inputs for NLIs in data visualization tools that focused on creating visualizations from a blank sheet [18]. However, we focused on editing existing visualizations and not creating visualizations from scratch. Furthermore, to reach a broader participant group, we utilized Amazon Mechanical Turk with a restriction of only workers from the U.S. with fluency in English. In line with our goal, of our 22 participants (Gender: 13 female, 9 male; Age: M = 37.5, SD = 11.2) only 18% rated their skills with data visualization tools as either good or excellent. In the study, the participants were asked to provide NL inputs for 19 unique actions of the data visualization tool (see Table 1 for examples). To investigate whether context matters, we elicited these 19 unique actions in up to 5 different contexts each, resulting in 40 contextual actions. For example, NL inputs for removing data fields from the values were elicited when only the data field was included in values and when an additional data field was also included and remained in the values. We further asked participants to provide two NL inputs per action to elicit a larger variety of NL inputs. This resulted in a total of 1760 labeled interactions (22 workers x 40 actions x 2 commands per action elicited). The study took on average 45 minutes and was compensated with US$ 9.

Table 1: Examples for Actions and Corresponding NL Inputs.

Action	Example Utterance
Add Data Field to x-Axis	Move State to the x-Axis
Change Aggregate of Data Field	Select max amount invested
Add Data Field to Values	Select max amount invested
Change Filter	Solar and Wind
Highlight Values in Visualization	Highlight sum of projects
...	...

3.1 Data Visualization Tool

We implemented a data visualization tool for our NL input elicitation study which employs a web-based, client-server model. After executing the query, the visualization is rendered using Vega-Lite [14] (see Figure 1). The data visualization tool enables users to specify the data fields visualized, aggregates of numeric data fields, numeric and categorical filters, as well as highlighting values and coloring elements in the visualization itself.

We utilized a dataset about renewable energy projects in the US that included *categorical* (e.g., States, Energy Types), *quantitative* (e.g., Amount Invested, Number of Projects), and *temporal* values (e.g., Years).

3.2 Procedure

Each session consisted of four phases: (1) *introduction & consent*, (2) a questionnaire about their *demographics & previous experience with data visualization tools and NLIs*, (3) *dataset description & exemplary analytical task* and (4) *elicitation of NL inputs*. For the elicitation of the NL inputs, we showed videos demonstrating the actions in our data visualization tool and asked participants what NL input would feel intuitive to them to invoke the previously shown action. The actions were shown to participants in a randomized order.

It was crucial to make the goal of the actions clear to participants so that they would not simply repeat what was performed in the video. For example, in a pre-test participants stated NL inputs, such as Click the drop-down menu and scroll to Solar [...]. In this example, participants did not understand that these steps are needed to specify a filter for the energy type Solar but just repeated the steps shown in the video. To address this issue, we adjusted our procedure and introduced participants in phase 3 of our study to the data visualization tool through an exemplary analytical task. Specifically, users were asked to utilize the data visualization tool *without* an integrated NLI to determine whether the statement ”In 2020 the amount invested in wind was lower than for renewable biomass” is true or false. This provided them with a better understanding of the functionality of the data visualization tool and led to fewer NL inputs that only describe the direct manipulations shown in the videos.

3.3 Results

First, we removed responses that were completely irrelevant or apparently due to laziness (6.5% of the total) to ensure proper quality

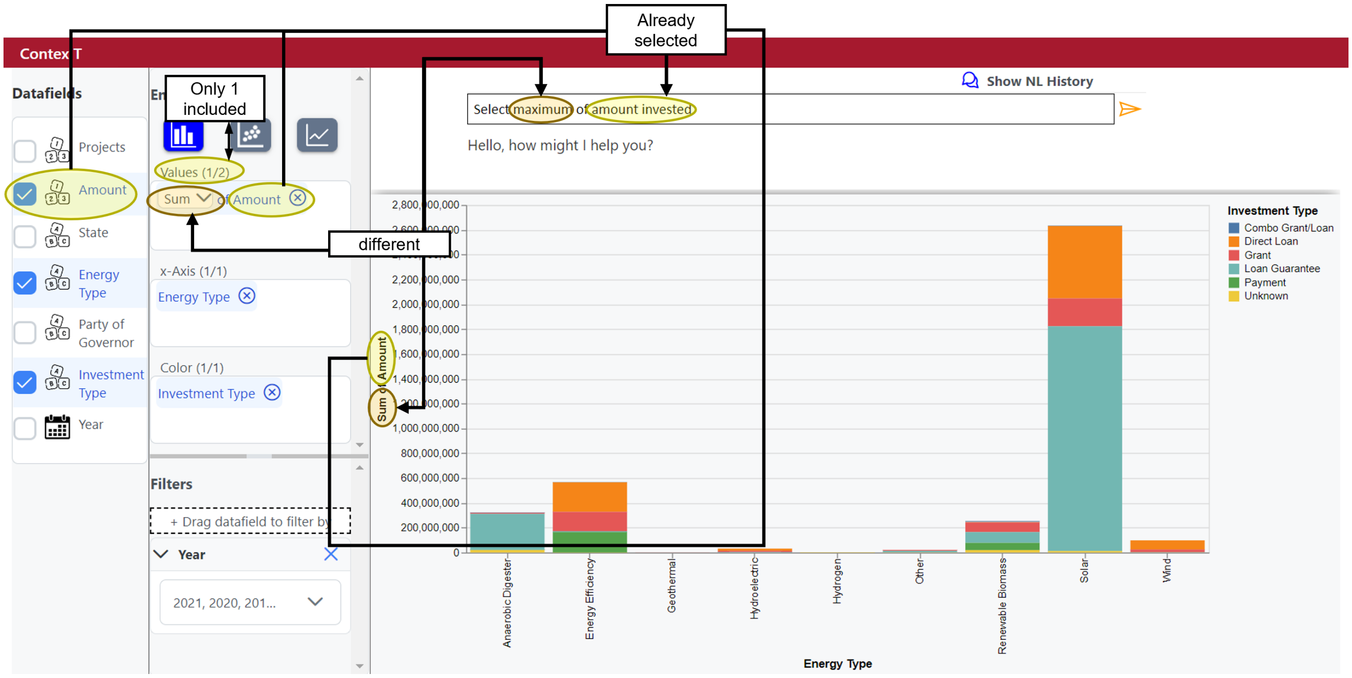

Figure 1: *ContexIT*'s User Interface as well as additional highlighting of crucial and supporting visual elements for understanding the NL Input Select *maximum* of *amount invested*. The NLI would be able to identify that the user wants to adapt the aggregation of the amount invested since the amount invested is already selected as a value in the visualization and is the only value included, and most importantly, the aggregation identified in the NL input differs from the one in the current visualization state.

of the NL inputs and to mitigate the limitations of utilizing Amazon Mechanical Turk workers instead of real users. Of the 1646 remaining NL inputs, 54.7% (900) were unique NL inputs that were proposed only once by participants. Additionally, 12.9% (213) of the total NL inputs exhibited ambiguities with regard to the goal of users. An NL input was classified as ambiguous if there existed an identical NL input after removing stop-words (i.e., *me*, *a* and *the*) which was suggested for a different unique action as well. For example, if an NL input was proposed both for adding and removing a data field, then it was ambiguous, but not if it was proposed for the *same* action in a different context. When weighted by the number of occurrences, 26% (56) of NL inputs that exhibited ambiguities with regards to the goal of users were associated with exactly two actions, while 74% (157) were used for more than two actions.

To analyze how these ambiguities could be clarified, we looked at both utilizing a probabilistic approach (associating the NL input with the action it was proposed for the most) and a context-dependent approach that utilizes contextual conditions. For the context-dependent approach, we looked into whether conditional statements based on the current visualization state (i.e., if a data field is selected or not) could be utilized to clarify ambiguous NL inputs. In the post-study analysis, selecting the appropriate contextual conditions to clarify ambiguities was difficult for us, especially if more than two actions were associated with an NL input. To decide between multiple plausible contextual conditions that could help clarify ambiguities, we first excluded those plausible contextual conditions that were not associated with any of the visual elements utilized or with attributes extracted from the NL input. For example, excluding whether a bar chart or scatter plot was selected to clarify the ambiguity of the NL input Deselect the Energy Types. From the remaining plausible contextual conditions, we heuristically selected those which enabled us to minimize the number of contextual conditions needed to clarify the ambiguities and were in line with our abstract understanding of the overall goal of the NL input, the dataset as well as the functionalities of the data visualization tools.

With a probabilistic approach, 42.7% (91) of the occurred ambiguities could be clarified. Through a context-dependent approach, we were able to improve this to 71.4% (150) correctly clarified ambiguities, which is a 167% increase. In our context-dependent approach, ambiguities with two different associated actions (26% of ambiguous NL inputs) could be clarified through single conditional statements. However, NL inputs with more than two actions associated often required multiple conditional statements that are linked together through AND clauses. For example, the NL inputs identical to Select maximum of amount invested were associated with the three distinct actions of (1) changing the aggregate (*maximum*) of a data field (*amount invested*), (2) adding a data field (*amount invested*) to values, and (3) highlighting the elements in the visualization associated with a data field (*amount invested*). The appropriate action could be selected by comparing whether the data field is selected and/or highlighted, how many values are selected, and if the extracted aggregate is different or identical to the currently selected aggregate of the associated data field. Utilizing these contextual conditions, the NLI would be capable of choosing the action to change the aggregate of *amount* to max in the visualization state of Figure 1 versus choosing to highlight the data field *amount* in the visualization state of Figure 2.

Additional anecdotal insights from the results are that while the similarity of verbs could be utilized to generalize the contextual conditions, one must be careful in doing so. For example, **Show** all energy types and **Select** all energy types have identical ambiguities. However, **Show** maximum of amount invested is not associated with the action of changing the aggregate which is in contrast to the ambiguities of **Select** maximum of amount invested. Therefore,

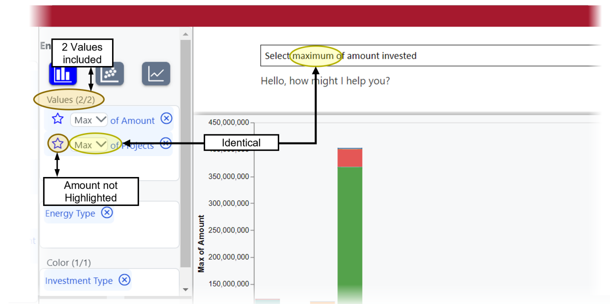

Figure 2: Highlighting of crucial and supporting visual elements for understanding the NL Input Select *maximum* of *amount invested* in an alternative context. The NLI would be able to identify that the user wants to highlight the amount invested in the visualization since amount invested is already selected as a value in the visualization, more than one value is displayed in the visualization, the aggregation identified in the NL input is identical to the one in the current visualization state, and amount invested is not yet highlighted.

while some similarities between ambiguities associated with similar verbs still exist, the results show that the similarity between verbs might only be utilized as an indicator and not as a robust condition.

3.4 Challenges of Pre-Defining Contextual Conditions

We wondered if a developer of an NLI system could just pre-program the identified contextual conditions to solve the ambiguities we identified. From our initial NL input elicitation, we derived the following challenges that such developers would face if they tried to do this:

Challenge 1. Deriving contextual conditions is a time-consuming activity
As with all labeling tasks, deriving contextual conditions from user interaction would require a lot of time from developers. First, appropriate actions and contexts with possible ambiguities would have to be identified by developers and integrated into an NL input elicitation study similar to ours. Subsequently, developers would need to derive contextual conditions from the results of these studies. Furthermore, this activity would need to be repeated with each new dataset since the NL inputs and their context differ across datasets [20].

Challenge 2. Unknown if *correct* conditions are identified
We were about to identify that the contextual conditions derived were able to clarify 71.4% of the ambiguities in our elicited NL inputs because we knew what the users were trying to do in our study. However, in a real situation, due to our lack of knowledge of the users' actual mental model, incorrect contextual conditions might be derived by developers.

Challenge 3. Unknown if *all* conditions are identified
While we already identified multiple ambiguities with our 22 participants, we did not reach saturation yet as the last participant still introduced new ambiguities. Hence, it is unclear when and if all possible ambiguities can be identified even for these 40 contextual actions on this one dataset. Furthermore, additional ambiguities might arise if new contexts for the unique actions are introduced or the dataset is changed. This indicates that even if developers would put a lot of effort in deriving the contextual conditions, there would likely still be a plethora of ambiguities that would require user involvement as they have not been addressed during the initial development of the NLI.

Another approach might be to automatically try to figure out the correct contextual conditions from what users do *at run time*, rather than trying to pre-program them all in advance. However, we additionally derived the following challenges of automatically deriving contextual conditions **without** user involvement during actual usage:

Challenge 4. Multiple potential conditions are valid
Automatically deriving contextual conditions would require the system to accurately select the appropriate contextual condition from all plausible contextual conditions. In our post-study analysis, we aimed to extract a procedure that could be programmatically performed by a system with the visualization states of the occurrences of ambiguity and the associated NL input as parameters. While the pre-selection of plausible contextual conditions based on the NL input and the visual elements utilized in the action is feasible, the final selection of the appropriate contextual condition required an abstract understanding of the overall goal, the dataset as well as the functionality of the data visualization tool for which existing NLIs are currently not intelligent enough [20].

Challenge 5. Distinction between preference and context
In our NL input elicitation study, we identified that some participants used identical NL inputs interchangeably in the same context for different actions. This indicated that some of the NL inputs that are ambiguous with regards to the users' goals cannot be clarified based on the context since they are due to the preferences of users. In our initial NL input elicitation study, 28.6% of the ambiguous NL inputs could not be clarified based on the context. A system, however, might over-interpret small differences in the context and would learn incorrect contextual conditions. Consequently, while a system might approximate if the usage could be due to preference because the differences in the visualization state are only minor, a user would be needed to ultimately clarify if they selected the NL input due to preference or context.

4 The ContexIT System

Because of these challenges, it is nearly impossible without user involvement to derive contextual conditions to clarify all possible ambiguities by developers upfront or by the system automatically during actual use of the NLI in the data visualization tool. Especially since the datasets and associated tasks vary greatly from user to user.

Therefore, we aim to *involve users* in clarifying ambiguities when they arise and enable the system to continuously *learn from* clarifications by the users to refine its understanding of the NL inputs and their meaning based on the context, so it would gradually reduce the need to ask for such clarifications.

4.1 Design Goals

To address the previously derived challenges, we distilled the following three design goals (DGs) based on our NL input elicitation study:

DG1. Enable users to specify contextual conditions for their NL inputs based on the current visualization state of the data visualization tool.
Users assume that the NLI has the same knowledge about the current context (i.e., visualization state) as they do [12]. Their knowledge about the current context is derived based on what they see in the data visualization tool. Hence, when users are asked to clarify ambiguities, the system should elicit the visual elements that are important to users for the current ambiguity and from this input derive contextual conditions.

DG2. Enable users to continuously refine or abstract contextual conditions.
Since users might over- or under-specify contextual conditions, the system should continuously adapt its existing conditions based on new clarifications from users. Over-specification might happen when users specify too narrow conditional statements. This includes when users specify conditional statements based on the current instance

of an attribute (e.g., Amount is visualized as values) while in reality the condition should be based on the abstract class associated with that attribute (e.g., a numeric data field is visualized as values). Under-specification might happen when the users are unsure what visual elements were important for clarifying the ambiguity and hence specify a vague area of importance.

To mitigate these issues, the system should continuously analyze the similarities and differences between visual elements utilized in the clarification of ambiguities in similar NL inputs. Based on abstract similarities of the visualization state of identical actions, the system can abstract the contextual conditions to avoid over-specification. Vague areas can be refined by the system using subsequent clarifications by specifically comparing the difference in the elements of the visualization state associated with the previously specified vague visual area to avoid under-specification.

DG3. Utilize background knowledge to support users. Users sometimes have difficulty in correctly identifying the visual element that would help the NLI understand their reasoning [10]. Therefore, users should be supported in specifying contextual conditions based on the visual elements through the background knowledge of the NLI and information derived from the ambiguity clarification.

As identified in the post-analysis of our NL input elicitation study, the system can be utilized to narrow down the plausible contextual conditions. After users clarify the ambiguity, the system can thus highlight the visual elements that are associated with the remaining plausible contextual conditions to simplify the selection. For example, when the NLI first tries to learn contextual conditions for our example in Figure 1, the system might highlight the drop-down menu associated with the aggregation and the selected data field associated with Amount to guide the user.

4.2 Architecture

ContexIT builds on the learning capabilities integrated in our previous ONYX system [13]. In addition to the previous capabilities, we improved the complexity of commands the NLI is able to understand. The NLI of *ContexIT* derives a collection of possible actions associated with the NL input and rates them. The features for the rating include (1) the score of the semantic parsing, (2) how well the visualization state fits the contextual conditions, and (3) the user ID. We introduce the user ID as a feature of the rating as users can have unique linguistic styles [21].

Furthermore, we introduce a mechanism to address unexpected system behavior through which users are able to (1) clarify the correct meaning of their command through disambiguation widgets similar to Datatone [3] and then (2) specify the contextual conditions important for clarifying ambiguities based on the users' interpretation of the visualization state. The key difficulty is to not only learn which specific visual element was crucial in this specific context (DG1) but to understand the abstract meaning behind the selection of this visual element (DG2). Hence, we are currently investigating how to display for users the different levels of abstraction of a visual element after they select it as a crucial visual element. For example, if users select the visual element of *Energy Type* in the x-axis as crucial, the system needs to clarify whether this ambiguity can be only clarified when the data field is specifically *Energy Type*, any data field, or only categorical data fields.

5 Conclusion

Clarifying ambiguities in NL input utilizing the current context, such as the visualization state, is an opportunity to enable the NLI to continuously learn from user interactions and to improve the overall experience. Our NL input elicitation study supports this opportunity while also highlighting the challenges in deriving contextual conditions to clarify these ambiguities by developers upfront or by the system automatically during actual use. *ContexIT* will aim to address these challenges and to enable users to specify contextual conditions based on the visualization state to realize the opportunities of a contextualized NLI. Subsequently, we will improve our design of the mechanism of *ContexIT* to address unexpected system behaviors by learning contextual conditions and implement this mechanism into *ContexIT* to rigorously evaluate the efficacy of our design in a quantitative evaluation.

References

[1] 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. doi: 10.1023/A:1011368926479

[2] 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*, vol. 2018-April, pp. 1–12. ACM, New York, NY, USA, 4 2018. doi: 10.1145/3173574.3174047

[3] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *UIST 2015 - Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, pp. 489–500. ACM Press, New York, New York, USA, 2015. doi: 10.1145/2807442.2807478

[4] E. Hoque, V. Setlur, M. Tory, and I. Dykeman. Applying Pragmatics Principles for Interaction with Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):309–318, 2018. doi: 10.1109/TVCG.2017.2744684

[5] Y. H. Kim, B. Lee, A. Srinivasan, and E. K. Choe. Data@hand: Fostering visual exploration of personal data on smartphones leveraging speech and touch interaction. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1–17. ACM, New York, NY, USA, 5 2021. doi: 10.1145/3411764.3445421

[6] 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, 10 2018. doi: 10.1109/VLHCC.2018.8506506

[7] T. J.-J. Li, M. Radensky, J. Jia, K. Singarajah, T. M. Mitchell, and B. A. Myers. 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*, pp. 577–589. ACM, New York, NY, USA, 10 2019. doi: 10.1145/3332165.3347899

[8] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin. Natural Language to Visualization by Neural Machine Translation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):217–226, 1 2022. doi: 10.1109/TVCG.2021.3114848

[9] S. Mazumder, B. Liu, S. Wang, and N. Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 21–31. Association for Computational Linguistics, Stroudsburg, PA, USA, 2019. doi: 10.18653/v1/W19-5903

[10] R. G. McDaniel and B. A. Myers. Getting more out of programming-by-demonstration. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 442–449, 1999. doi: 10.1145/302979.303127

[11] S. Mirchandani, S. Karamcheti, and D. Sadigh. ELLA: Exploration through Learned Language Abstraction. *Advances in Neural Information Processing Systems*, 35(NeurIPS):29529–29540, 2021.

[12] T. Reinhart. Pragmatics and Linguistics: an analysis of Sentence Topics. *Philosophica*, 27(0):53–94, 1 1981. doi: 10.21825/philosophica.82606

[13] M. Ruoff, B. A. Myers, and A. Maedche. ONYX - User Interfaces for Assisting in Interactive Task Learning for Natural Language Interfaces of Data Visualization Tools. In *Proceedings of the 2022 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, vol. 1, pp. 1–10. Association for Computing Machinery, 2022. doi: 10.1145/3491101.3519793

[14] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on*

Visualization and Computer Graphics, 23(1):341–350, 1 2017. doi: 10.1109/TVCG.2016.2599030

[15] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pp. 365–377. ACM, New York, NY, USA, 10 2016. doi: 10.1145/2984511.2984588

[16] 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. ACM, New York, NY, USA, 3 2019. doi: 10.1145/3301275.3302270

[17] 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. ACM, New York, NY, USA, 4 2020. doi: 10.1145/3313831.3376782

[18] A. Srinivasan, N. Nyapathy, and B. Lee. Collecting and characterizing natural language uterances for specifying data visualizations. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1–10. ACM, New York, NY, USA, 5 2021. doi: 10.1145/3411764.3445400

[19] 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. doi: 10.1007/978-3-642-13544-6

[20] M. Tory and V. Setlur. Do What I Mean, Not What I Say! Design Considerations for Supporting Intent and Context in Analytical Conversation. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 93–103. IEEE, 10 2019. doi: 10.1109/VAST47406.2019.8986918

[21] S. I. Wang, S. Ginn, P. Liang, and C. D. Manning. Naturalizing a Programming Language via Interactive Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 929–938. Association for Computational Linguistics, Stroudsburg, PA, USA, 2017. doi: 10.18653/v1/P17-1086