Developing Design Features to Facilitate AI-Assisted User Interactions

A Technical Report submitted to the Department of Systems and Information Engineering

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Anika Sharma
Spring, 2022
Technical Project Team Members
Stacy Meng
Parker Schell
Anmol Kaur
Ghislain Ventre
Rebecca Dollahite

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Gregory Gerling, Department of Systems and Information Engineering

Developing Design Features to Facilitate AI-Assisted User Interactions

S. Meng[1], R. Dollahite[1], A. Kaur[1], P. Schell[1], A. Sharma[1], G. Ventre[1], I. Kranz[2], G. Nudelman[2], G. J. Gerling[1]
1. University of Virginia, {swm8xb, red5zk, ak4pqt, tps9us, as5ssq, gjv9uf, gg7h}@virginia.edu
2. Sumo Logic, {gnudelman, ikranz}@sumologic.com

***Abstract*—Interactive software tools employing generative artificial intelligence (AI) that help users formulate custom system queries are increasingly needed with growth in data quantities, relationships, and complexity. The need to afford such interactions is not new. Indeed, chatbots have long sought to bridge gaps between an individual's intent and the system's response. However, generative AI chatbots – in contrast to traditional chatbots that navigate pre-defined, rules-based decision trees – are unique in their promise to accept and respond to highly customized queries. At present though, most still rely upon the precise articulation of a structured prompt. The work herein develops and evaluates design features to facilitate AI-assistive user interactions in query formulation. The design features attempt to balance functional needs of users to make specific, goal-oriented, customized queries, with minimal constraints on exactly articulating pre-defined prompts. In a case study, we wireframe user interface prototypes in the domain of data log management, for evaluation with expert and novice users. Key elements of the design features revolve around the 1) refinement of search categories, 2) context-aware prompt recommendations, and 3) customization of query input per a user's technical ability.**

Keywords: User Experience Design, Interaction Design, Prompt Engineering, AI-Assistive Technologies

I. INTRODUCTION

With the integration of generative AI-assistive tools into a user's experience is a promise to accept and respond to nuanced, specific, and individualized intent. Indeed, a user's desire for a personalized response to his or her request has been long sought. For example, traditional search engines accept user queries via open-ended word entry, with subsequent access to curated lists of frequent and/or recent queries based on keyword associations. However, their users are typically expected to retrieve their search keywords and criterion from memory and the system's response can vary heavily depending on keyword interpretation [1]. Along more interactive lines are chatbots that employ natural language processing (NLP) and follow pre-defined decision trees to arrive at the formulation of a final response. Chatbots are semi-conversational and funnel users along restrictive paths, based on an individual user's step-by-step sequence of input choices [2]. Perhaps as a result, interactions with chatbots are perceived negatively about 40% of the time [3].

Building upon the positive interactive attributes of chatbots, new generative AI-assistive approaches promise to accept and respond to highly customized input queries. For example, Microsoft Copilot [4] uses large-language models (LLMs) to integrate closely with Microsoft (MS) Office applications. While Copilot streamlines productivity across many tasks, it can be unreliable in terms of accuracy and often fails to grasp the specificity of a task [5]. For instance, MS Copilot streamlines complex data interpretation, allowing users to focus on more strategic tasks. However, the suggested analysis is not always compatible with the context of the original problem. Similarly, GitHub Copilot can help write lines of code in software, though it can lack sufficient context to provide an accurate response at the initial iteration [6]. Rather than offering a single response, which a user may not critically examine, other tools employ a suggestion-based approach. For instance, Canva's Magic Studio transforms a user's query into visual design suggestions, such as banners, presentations, videos, and other digital media. This approach to AI integration enables a more interactive and iterative workflow [7]. Such situations raise important questions about the nature of AI assistance in the prompt experience. In particular, we must be cognizant of how much an interaction designer can simplify the user input prompt in terms of efficiency while still affording highly individualized intent.

As with user experience design in general, the precise use of design features – i.e., strategies used in task representation, presentation, and interaction – is key to user comprehension and acceptance. For example, AI-assistive chatbots, such as OpenAI's ChatGPT, solicit user input via a blank text box while listing example queries to help the user get started, and then rely on the user to iterate and refine their queries to improve the system's response. Meanwhile, MS Copilot can provide hyper-specific query outputs due to its awareness of functionalities within and user data across the MS Office suite. On the other hand, traditional search engines afford levels of search granularity in filtering output by keywords. Here for example, Google Search allows a user to query "tiger" and then subsequently refine the range of responses per media type, e.g., "images," "videos," and "news." A user selecting "images" can then further categorize types of tiger images with system-generated suggestions, e.g., "clipart," "cute," and "realistic." It is also possible to initially input "tiger images" to which the system's response is to automatically refine and prioritize the display of images, as opposed to hyperlinks to relevant sites or videos. As these examples indicate, streamlined interaction design, with the right balance of suggestive assistance, that is customized to the user's intent and mental model, may ultimately incsrease user satisfaction and task efficiency.

II. METHODS

This work develops and evaluates design features to facilitate AI-assistive user interactions in the domain of log management. Applications in this domain must address needs

of experienced users wishing to formulate queries by writing lines of code, as well as novice users lacking coding skills and an understanding of relationships between system variables and attributes. The design features seek to balance user desires to make goal-oriented, customized queries, with minimal constraints on exactly articulating pre-defined prompts. To conceptualize user interface designs and workflows, a series of wireframe prototypes were created. Usability evaluations were conducted with subject-matter experts in log management, and usability experts in heuristic evaluation.

We introduce three important design features, which include 1) refinement of search categories, 2) context-aware prompt recommendations, and 3) customization of query input per a user's technical ability. First, the refinement of search categories limits the user's query to a subset of log data, which is critical to improving response time and result relevancy. Second, context-aware prompt recommendations provide each user with a list of query suggestions that expand upon that user's selected search category and/or previously written prompts. This feature helps expedite the exploration of data by leveraging AI language processing capabilities to understand the user's search context and engineer ready-to-run prompts. Third, the customization of query input per a user's technical ability allows for an inclusive interface to accommodate levels of software coding experience from those well-versed in platform-native syntax to beginners who rely upon NLP capabilities to formulate a query.

III. Methods: Requirements Gathering

Information and functional requirements were derived from interviews with domain experts in log management and security monitoring. Logs are digital records of events and activities within a system, application, or device. In the field of monitoring software, log data are grouped by data sources into search categories. Users write queries to search, filter, and analyze log data to gain insights into a company's infrastructure performance, potential security threats, and other technical issues. To perform an effective search, users must use relevant keywords written in platform-native syntax and specify a timeframe and search category.

Herein, we examine a popular scenario where users search logs to identify potential for malicious SQL code injection into a company's databases. To frame the information and functional requirements, we incorporated AI assistance in two use cases: 1) when the AI tool produces an accurate category suggestion with which the user chooses to engage, and 2) when the AI tool fails to provide an adequate suggestion, and the user must recover by manually selecting a category.

A. Information Requirements

Collection of Search Categories: A client environment can host thousands of search categories. Thus, the collection of all categories should be readily accessible for reference during the user's search workflow.

Search Category Level: Search categories often consist of multiple layers of depth that gradually narrow the scope of log datasets. Users need visibility into this hierarchy.

Log Query Result: The response to a completed query should consist of relevant log entries and corresponding metadata. The results should be optionally displayed in graphical forms, such as a bar chart with other format options.

Log Context: Users must be given sufficient context of the log events recorded by the system such as timestamps, error codes, and source addresses. This information will allow the user to derive and form conclusions from their search.

B. Functional Requirements

Refinement of Search Category (SC): In the aforementioned use case 1, users should be provided a list of relevant search categories based on the input query, with the choice to self-select. In use case 2, if the user does not find the AI assisted recommendations suitable, the system should support reverting to manual category selection and be able to view the full collection of categories. The visual display of search categories must be hierarchically organized and align with the user's mental model. The integration of AI assistance may alter the user's understanding of the search process; thus, it is vital to embed AI assistance within the existing search workflow rather than as a separate, companion-style assistant that drastically changes the current user flow.

Context-Aware Prompt Recommendations: A redesigned AI-assisted search process should leverage the language processing capabilities of AI and allow users to perform queries using natural language rather than platform-native syntax. To help a user write prompts, the system should suggest completely formulated prompts that expand upon a user's prior interactions, such as search categories and queries. The system should attain a deep understanding of the user's environment, work domain, and previous search history to produce context-aware, well-informed recommendations.

Input Maturity: The interface should accommodate varying levels of user expertise in log management, ranging from those well-versed in platform-native syntax to beginners who rely upon NLP capabilities to formulate a query. Users should be able to quickly transition between the varying input methods and should saliently display the current mode.

IV. Methods: Prototype Design

Regarding the first feature of search category refinement, four design options were developed to address relevant information and functional requirements.

A. Option 1: Federated Search

Option 1 for refinement of search categories employs a federated search, which simultaneously runs a query on log data from multiple sources and displays results on one screen (Fig. 1A). When the user searches with an NLP prompt, multiple AI-suggested search categories populate as a horizontal row of clickable tags above the results data table,

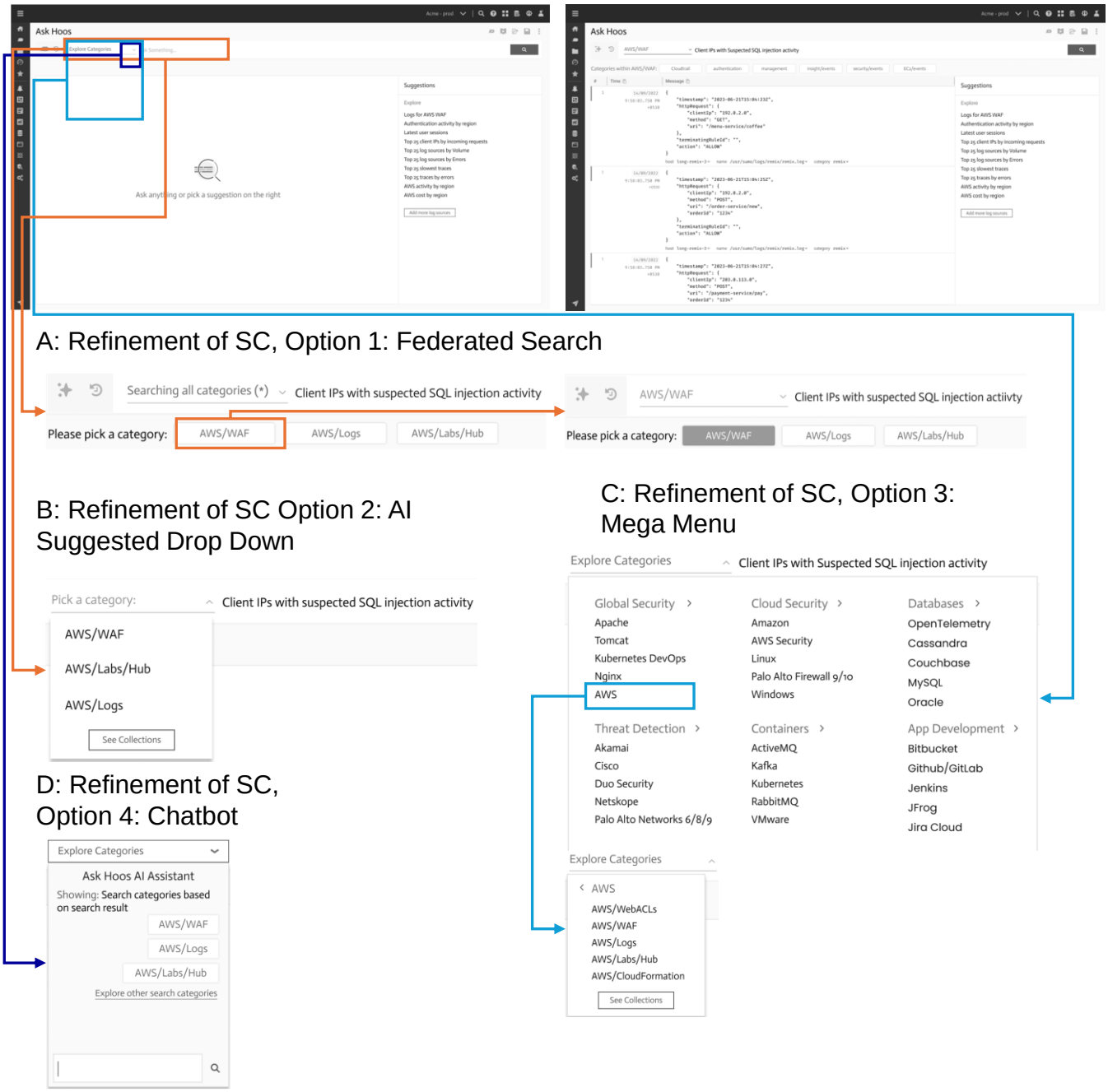

Fig. 1. **Interaction Elements in Refinement of Search Categories.** (A) Federated search allows users to quickly select AI suggested categories displayed in card format laterally under the search bar. (B) A drop down menu appears for AI suggested categories after processing an NLP query. (C) A mega menu appears when a user selects the drop-down on the explore sources button, displaying different categories based on similar groupings, which transitions into sub-categories. (D) The chatbot option features unique AI capabilities, allowing users to communicate with the AI via text to determine the correct category to use.

ready for the user to select and retrieve that subset of data. Because search categories are typically multi-leveled, users can select multiple categories under a hierarchy to filter the data. After a category is chosen, the suggested list will regenerate to become the next most relevant set of categories based on what the user previously selected.

B. Option 2: Search Bar Hosted Drop Down List

Option 2 presents AI-recommended categories as a drop down list immediately to the left of the search bar, closely associating the category selection with the user's search prompt (Fig. 1B). As the user types their prompt, the AI would interpret their working input and continuously generate appropriate category suggestions. Hosting category selection

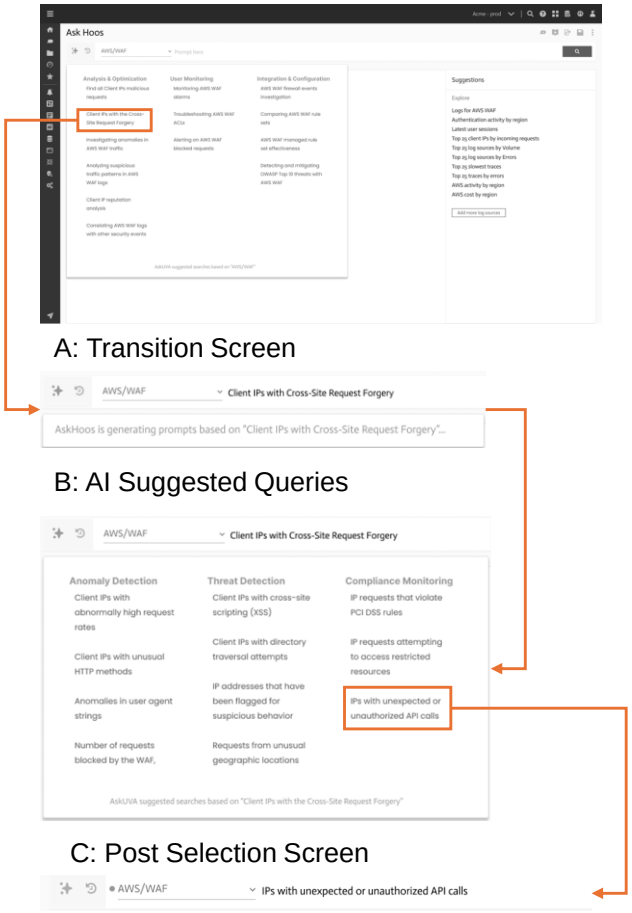

Fig. 2. **Interaction Elements in Context-Aware Prompt Recommendation.** (A) System feedback to indicate new suggestions are loading based on the selected query. (B) Display of final suggestions in menu format. (C) The secondary query is updated in the search bar, the notification dot next to search category name in drop down indicates new search categories are recommended.

in the search bar provides users with the flexibility to start their search by either selecting a category already in mind manually or by typing their prompt and leveraging the AI interpretation to identify the best category.

C. Option 3: Mega Menu

Option 3 for refinement of search categories provides a recovery option for cases where AI assistance is not utilized in the form of a comprehensive category menu: a mega menu (Fig. 1C). The mega menu is a large two-dimensional drop down panel organizing search categories into affinity groups with a system-defined heading. The most relevant options of categories within the domain are visible at once. Because these categories span multiple levels, once a user selects a category from the mega menu, the menu facets down to a truncated version of itself featuring subcategories based on the previously selected category.

D. Option 4: Interactive Chatbot

Option 4 for refinement of search categories incorporates an alternative recovery option in the form of a chatbot which provides a second instance of NLP exclusively for search category selection (Fig. 1D). This additional instance of NLP features a messaging window for users to input specific inquiries related to search category selection. From this, the chatbot will continue to populate categories and allow additional exploration, leading to a larger infrastructure that shows all available search categories.

E. Extended AI Functionalities

The following two design features address aforementioned functional requirements — context-aware prompt recommendations and input maturity — accommodating multiple levels of users and search category selection.

Contextually Aware Prompt Recommendations: In *Methods sections A-D*, AI assistance was integrated within the category selection process to provide suggestions based on the user's active search. These interactions can be expanded by exploring usage of AI to generate entire search prompts. This AI tool would be activated in two situations, after: 1) selecting a search category, and 2) executing a query. In the first situation, the AI will interpret the category context from patterns in prior usage and generate potential search prompts of interest. For the second situation, the AI will generate related prompts expanding on the previous prompt's general goal (Fig. 2A-B). If the user had already loaded a search category from their previous search, the AI would produce new category recommendations based on the alternate prompt

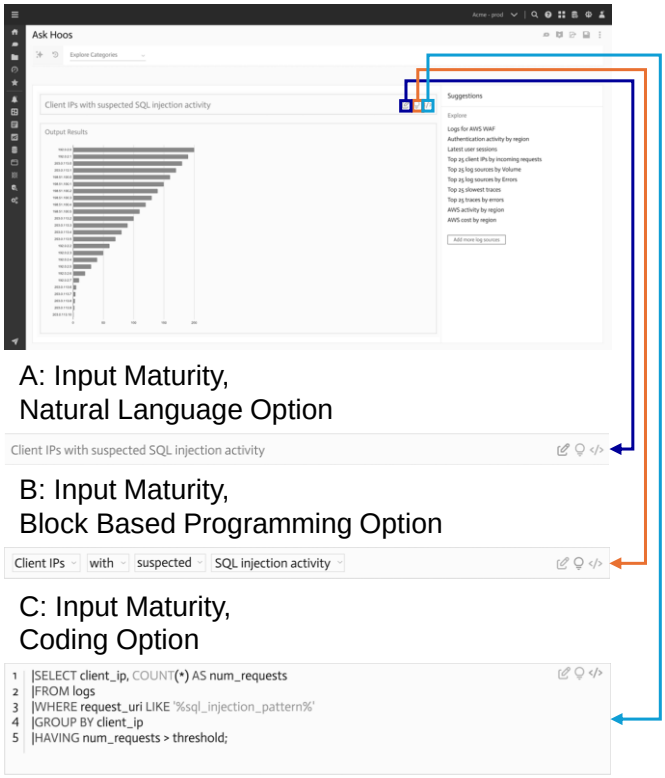

Fig. 3. **Interaction Elements in Input Maturity Customization.** (A) Users to input natural language to generate queries. (B) Users can alter individual relevant keywords from prompt via a drop down. (C) Users can use the platform-specific coding syntax.

selected and indicate the change via a notification dot (Fig. 2C). This tool is accessible via the leftmost icon on the search bar palette, which displays as a drop down selection menu and categorizes prompts by column based on broad search theme.

Customized Query Input Maturity: In the previously introduced design features, the users could directly use natural language within the search bar. However, the system must accommodate a wide range of users from experienced to novice. Here, users can choose among three levels of input maturity: 1) natural language, 2) block-based programming (BBP), and 3) platform-native code syntax (Fig. 3).

The initial screen displays the three input maturities, the search box, and the output results section. Natural language (Fig. 3A) allows novice users to type their intent if they have low familiarity with coding. In contrast, BBP (Fig. 3B) allows users to modify their search by breaking down NLP intent into smaller blocks with drop downs. This intermediary option allows users to refine their analysis by changing individual keywords. The third option displays platform-native code syntax (Fig. 3C) that is tailored for experienced users who can directly code their query or alter specific lines. This tool is accessed from the main screen. Changes made for one option are reflected across all options and save a user's progress.

V. Methods: Usability Evaluation

A. Usability Evaluators

A usability evaluation was performed to assess the effectiveness of and satisfaction with the design features. Distinct findings were derived separately for experienced and novice users, seeking to capture technical diversity. The evaluators included two data analysts with previous log management experience and a novice graduate student with no domain experience but with experience in usability.

B. Evaluation Metrics

Five of Nielsen's 10 Usability Heuristics [8] were most informative in assessing the effectiveness and satisfaction of the design features. These are described below.

Flexibility and Efficiency of Use: The system should support a wide range of users with varying skill levels and enable users to quickly accomplish familiar system tasks. Shortcuts that quicken experienced users' interactions should not pose a barrier to inexperienced users.

Consistency and Standards: The system should follow the previously set standards of the platform, decreasing the user's cognitive load when it comes to learning new functions and features. It should match a user's expectation with the functionality in a consistent fashion.

Match Between the System and the Real World: The system should present information aligned with real-world concepts and conventions familiar to the user. Symbols and terminology should be consistent with expectations and prior experience, complimenting the user's mental model.

User Control and Freedom: The system should allow users to return to and refine previous actions. In cases where users encounter roadblocks, the system should be equipped with salient features enabling error-recovery and fluid backtracking.

Help Users Recognize, Diagnose, and Recover from Errors: The system should enable users to adequately recover from mistakes by clearly conveying error messages in plain language and offering a feasible solution.

C. Procedures

Novice Users: The study evaluator walked the user through the four search category refinement options (Fig. 1) and prompted the user to provide qualitative feedback. Options 1 and 2 included the user selecting the most appropriate AI-suggest search category, with the ideal path: 1) the user types in "Client IPs with suspected SQL injection activity, 2) the system says, "Please pick a search category", 3) the user selects the target search category (AWS/WAF) from the presented lists, 4) the system runs the query and displays the log results. Options 3 and 4 address user navigation when AI fails to output the desired category, highlighting the ability to recover from AI-induced errors and manually navigate through categories. Options 3 and 4 proceed as follows: 1) the user types in "Client IPs with suspected SQL injection activity", 2) the system prompts selection of a search category from incorrect options, 3) the user rejects the suggestions and instead browses through additional sources, 4) the user selects "AWS/WAF", 5) the system runs the new query and displays the results.

Expert Users: The expert evaluators conducted an explorative usability evaluation, being presented three of the four search category selection options: federated search, search bar hosted drop down menu, and the mega menu. They were also asked to provide feedback on improvements to the filtering and aggregating log data process. For each option, research questions were asked to collect qualitative feedback to test assumptions and uncover issues within the interface.

VI. Results

A. Federated Search

The novice evaluator was unable to successfully identify the function of the federated search buttons (Fig. 1A), instead mistaking them for a set of filters on a completed query. This increased the user's cognitive load to learn the tool's exact purpose and hindered the novice's performance as well as the system's flexibility and efficiency of use. The expert evaluators were doubtful about the feasibility of the federated search, as it would fail to display categories in instances when an overwhelming number of possible search category refinement options were present, disrupting the consistency and standards present within the log management sphere.

B. AI Suggested Drop Down

The novice user gave positive feedback on the drop down

(Fig. 1B) with little critique. The expert evaluators gave positive feedback on the AI suggestions in the search category drop down, especially for circumstances when users are unsure of the most appropriate search category. They cited that an AI suggested drop down would help narrow the thousands of search categories within the system and increase the system's flexibility and efficiency as well as align with existing consistencies and standards. They were, however, behaviorally drawn to typing out the search category compared to choosing a suggested option through UI elements. Typing allowed the evaluators the ability to enter a wild card (*), which encompasses all the categories in a specific domain and emphasizes a user's control and freedom over the system. This preference contributed to the expert evaluators favoring the drop down option.

C. Mega Menu

The novice evaluator expressed that the category hierarchy of the mega menu (Fig. 1C) was disorienting, especially when presented with a truncated version of a separate hierarchy. This created inconsistencies in the system organization pertaining to the increased cognitive load on the user. The mega menu was not within the scope of the expert evaluators' mental model, creating a disconnect between the system and the real world. Yet, they did acknowledge that it would align more with a novice's system expectations.

D. Chatbot

The novice evaluator struggled to find the chatbot icon (Fig. 1D) and launch the interaction, expressing another gap in the match between the system functionality and the real world. This indicated a need to either make the icon more salient or have the model appear automatically. The novice user also acknowledged that the chatbot did not enable frequent error-prevention, as mistakes commonly occurred due to a mismatch between the user's mental model and design. There was not adequate infrastructure to aid users in recognizing, diagnosing, and recovering from errors. The expert users were not presented with the chatbot.

E. Additional Feedback

The expert users expressed that they preferred executing commonly used functions with UI elements rather than natural language across all proposed options. They also found that excess typing for commonly used queries made interaction more arduous. For example, experienced users preferred functions such as changing a graph from a bar to a pie chart to be performed through a panel of one-click buttons, thus aligning the system with the real world. Overall, the experienced users preferred a system that allowed for greater efficiency of use with flexible processes.

VII. Discussion

To explore the bounds between user customization and AI-integrated assistance, we developed design features in the data analytics field to provide transferable insights into the user querying process. Since the design process was evaluated based on two key use cases differing on the accuracy of search category selection, user behavior was better isolated when exposed to different levels of AI assistance. Federated search utilized a suggestions-based approach to allow users to traverse through levels of category selection more easily. Meanwhile the search bar hosted drop down offered a higher level of assistance since it generated suggestions on a more query-specific scale, considering individual and aggregated user history. The mega menu offered a more traditional method for error-correction. On the other hand, the chatbot creates a high level of AI dependency based on NLP. This contrast in depth of suggestive interference for each option illustrated the high impact a single design choice has on guiding the user to a desired result.

In future work, the impact of skill gaps due to user experience on design choices could be further analyzed with additional exploratory feedback in these industries. Since the difference in results was so drastic depending on the level of AI assistance, it is likely that this factor will be relevant across the data analytics industry more broadly relating to user experience. Finally, the design feature pertaining to context-aware prompt recommendations could garner further investigation as an additional extension to the paper. It remains evident, still, that organizations must consider user experience on account of the degree of AI assistance when incorporating any smart solution.

Acknowledgments

We would like to thank the anonymous subject-matter experts and usability evaluators for their feedback.

References

[1] T. M. Parks, "Search Engines," in *Encyclopedia of Information Systems*, Elsevier, 2003, pp. 23–29. doi: 10.1016/B0-12-227240-4/00154-4.

[2] Oracle, "What is a Chatbot?," Oracle Cloud Infrastructure. Accessed: Apr. 02, 2024. [Online]. Available: https://www.oracle.com/chatbots/what-is-a-chatbot/

[3] Gil Press, "One Negative Chatbot Experience Drives Away 30% Of Customers," Forbes. Accessed: Apr. 02, 2024. [Online]. Available: https://www.forbes.com/sites/gilpress/2023/02/01/one-negative-chatbot-experience-drives-away-30-of-customers/?sh=235dfef62510

[4] Pamela Arimoto, Jitin Mathew, Aditi Srivastava, and Priya Rakshith, "Copilot for Microsoft 365," Microsoft. Accessed: Apr. 02, 2024. [Online]. Available: https://learn.microsoft.com/en-us/office365/servicedescriptions/office-365-platform-service-description/microsoft-365-copilot

[5] C. W. P. Lewis and A. Dziewulska, "Improve Your Company's Use of AI with a Structured Approach to Prompts," Harvard Business Review. Accessed: Apr. 02, 2024. [Online]. Available: https://hbr.org/2023/11/improve-your-companys-use-of-ai-with-a-structured-approach-to-prompts

[6] Netguru, "GitHub Copilot Pros and Cons," Netguru. Accessed: Apr. 02, 2024. [Online]. Available: https://www.netguru.com/blog/github-copilot

[7] Canva, "Magic Studio," Canva. Accessed: Apr. 02, 2024. [Online]. Available: https://www.canva.com/magic/#features

[8] J. Nielsen, "10 Usability Heuristics for User Interface Design," Nielsen Norman Group. [Online]. Available: https://222.nngroup.com/articles/ten-usability-heuristics/