RESEARCH-ARTICLE

Rethinking Dataset Discovery with DataScout

RACHEL LIN, University of California, Berkeley, Berkeley, CA, United States

BHAVYA CHOPRA, University of California, Berkeley, Berkeley, CA, United States

WENJING LIN, University of California, Berkeley, Berkeley, CA, United States

SHREYA SHANKAR, University of California, Berkeley, Berkeley, CA, United States

MADELON HULSEBOS, Center for Mathematics and Computer Science - Amsterdam, Amsterdam, Noord-Holland, Netherlands

ADITYA GANESH PARAMESWARAN, University of California, Berkeley, Berkeley, CA, United States

Open Access Support provided by:

University of California, Berkeley

Center for Mathematics and Computer Science - Amsterdam

Published: 28 September 2025

Citation in BibTeX format

UIST '25: The 38th Annual ACM Symposium on User Interface Software and Technology
September 28 - October 1, 2025
Busan, Republic of Korea

Conference Sponsors:
SIGCHI
SIGGRAPH

UIST '25: Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (September 2025)
https://doi.org/10.1145/3746059.3747727
ISBN: 9798400720376

Rethinking Dataset Discovery with DataScout

Rachel Lin[*]
University of California, Berkeley
Berkeley, California, USA
raelin@berkeley.edu

Bhavya Chopra[*]
University of California, Berkeley
Berkeley, California, USA
bhavyachopra@berkeley.edu

Wenjing Lin
University of California, Berkeley
Berkeley, California, USA
wenjing.lin@berkeley.edu

Shreya Shankar
University of California, Berkeley
Berkeley, California, USA
shreyashankar@berkeley.edu

Madelon Hulsebos
Centrum Wiskunde & Informatica
Amsterdam, Netherlands
madelon@cwi.nl

Aditya G. Parameswaran
University of California, Berkeley
Berkeley, California, USA
adityagp@berkeley.edu

Abstract

Dataset Search—the process of finding appropriate datasets for a given task—remains a critical yet under-explored challenge in data science workflows. Assessing dataset suitability for a task (e.g., training a classification model) is a multi-pronged affair that involves understanding: data characteristics (e.g. granularity, attributes, size), semantics (e.g., data semantics, creation goals), and relevance to the task at hand. Present-day dataset search interfaces are restrictive—users struggle to convey implicit preferences and lack visibility into the search space and result inclusion criteria—making query iteration challenging. To bridge these gaps, we introduce DataScout to proactively steer users through the process of dataset discovery via—*(i)* AI-assisted query reformulations informed by the underlying search space, *(ii)* semantic search and filtering based on dataset content, including attributes (columns) and granularity (rows), and *(iii)* dataset relevance indicators, generated dynamically based on the user-specified task. A within-subjects study with 12 participants comparing DataScout to keyword and semantic dataset search reveals that users uniquely employ DataScout's features not only for structured explorations, but also to glean feedback on their search queries and build conceptual models of the search space.

CCS Concepts

• **Human-centered computing → Systems and tools for interaction design**; • **Information systems → Search interfaces**; **Collaborative search**.

Keywords

Exploratory Dataset Search, LLMs, Human-AI Interaction

ACM Reference Format:
Rachel Lin, Bhavya Chopra, Wenjing Lin, Shreya Shankar, Madelon Hulsebos, and Aditya G. Parameswaran. 2025. Rethinking Dataset Discovery with DataScout. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28–October 01, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3746059.3747727

[*]Co-first authors. Corresponding author: Bhavya Chopra.

UIST '25, Busan, Republic of Korea

ACM ISBN 979-8-4007-2037-6/25/09
https://doi.org/10.1145/3746059.3747727

1 Introduction

Finding the right dataset, given a data analysis or machine learning task, is one of the most challenging problems for data scientists and analysts today [8]. This problem of *dataset search* is only growing more urgent—with organizations often accumulating tens of thousands of tables in their data lakes [25]. Dataset search is difficult for a couple of reasons. First, real-world data is inherently messy: tables vary widely in quality and metadata completeness, with many lacking proper descriptions, having ambiguous column names, or containing outdated information [55]. Second, users rarely know exactly what they are looking for [22]. They might have a general task in mind, like training a machine learning model to predict some phenomenon, but do not know which datasets would be compatible with their task.

Recent advances in Large Language Models (LLMs) have demonstrated the potential to address some of the aforementioned challenges. Embedding models enable us to transform unstructured text into numerical representations (i.e., embeddings) that capture semantics, allowing systems to perform a *semantic search* to find relevant datasets, even when the exact terminology differs [64]. For example, Olio [51] can interpret a natural language (NL) question like "how has unemployment changed since 2020" and find relevant datasets—even if the metadata does not have a perfect keyword match with the question. However, semantic search of this form is often opaque to users, making it difficult to understand *why* a particular dataset appears in the search results, or *how* it relates to their query—plus users are unable to adaptively explore the content within datasets, including the columns/attributes, and temporal/spatial granularity. Overall, despite these advances in interpreting NL queries, present-day dataset search interfaces—be it semantic or keyword-based—provide limited support for search expressiveness—illustrating a wide gap between what technology can enable, and what interfaces currently facilitate.

Moreover, users typically lack awareness of available datasets, and must learn about the dataset landscape through the search results themselves, which subsequently inform refinements of their queries. This makes dataset search an inherently exploratory, iterative, and often tedious process requiring multiple query reformulations and result assessments [22]. Users have to rely on the assistance of colleagues for starting points, or even direct identification of the relevant datasets—indicating just how poor present-day dataset search interfaces are in supporting iterative exploration.

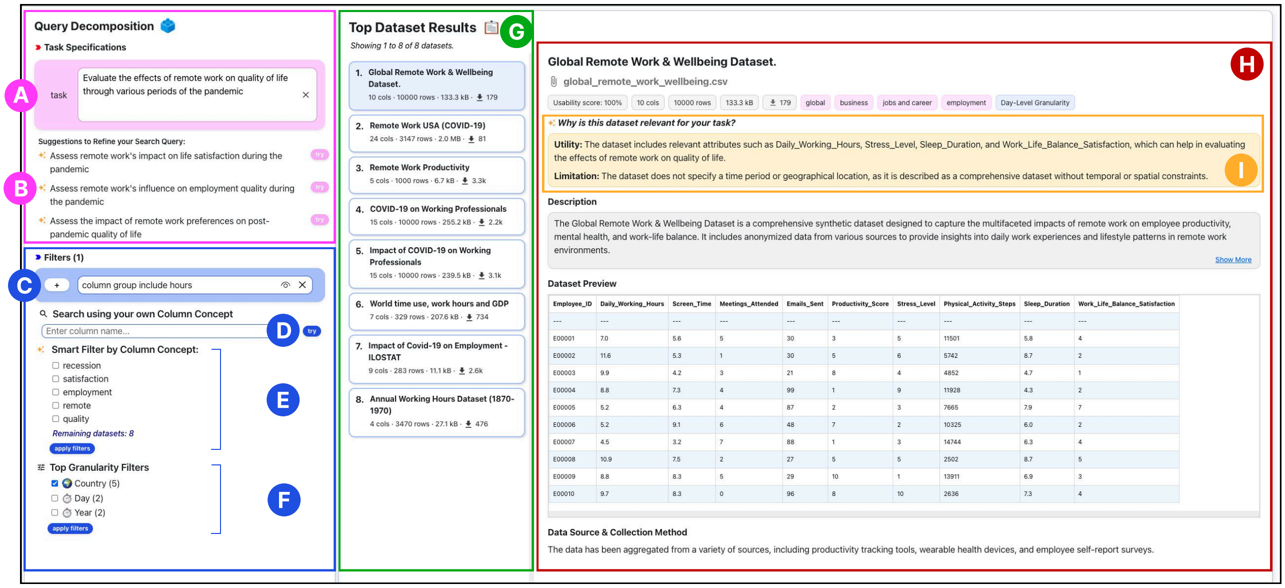

Figure 1: DataScout—a proactive dataset discovery interface. (A) Users begin by specifying their dataset discovery query as keywords, phrases, or complete sentences. (B) DataScout provides proactive query reformulation suggestions to bridge the gap between the user's query and datasets available in the search space. (C) Users may add exact matching-based or semantic filters, (D) search by attribute, apply (E) suggested attribute filters, or (F) suggested temporal and spatial granularity filters. (G) Users can explore ranked dataset search results in a consolidated view. (H) Selecting a dataset reveals its metadata, tags, description, preview, collection details, and (I) task-specific relevance indicators generated on-the-fly, highlighting utilities and limitations of the dataset.

In this work, we explore the design of dataset search systems that can proactively support users' iterative discovery process. To do so, we first conducted a formative study to identify aspects of users' dataset search workflows that could be amenable to automated assistance. Our study findings reveal that:

- Users ***lack efficient means to express intent***, finding dataset search interfaces to be restrictive in filtering based on content, attributes (columns) and granularity (rows);
- Users receive ***limited insight into characteristics of the dataset search space***, such as the space of possible columns across the returned results;
- Users ***struggle to reformulate their queries*** when encountering overly selective or irrelevant datasets; and
- Users ***spend significant time in assessing dataset relevance*** in context of their analytical needs, especially when the dataset description focuses on what the dataset contains, not the purposes it can be used for.

These limitations underscore the need for dataset search interfaces that *proactively empower users with feedback and assistance to iteratively reformulate queries, interpret search results, and navigate the dataset search space.* We present DataScout, a dataset search tool that proactively steers users through the process of dataset discovery (Figure 1). DataScout assists users in finding target datasets by being cognizant of both the user-specified task as well as the underlying space of results. DataScout offers three key LLM-powered semantic assistance features: *(i)* **proactive query reformulation (Figure 1B)** to bridge the gap between users' search queries and the underlying search space, ensuring that each reformulation is both diverse and grounded in actual search results by covering a subset of the dataset corpus, *(ii)* **semantic search (Figure 1D)** and **filtering** based on dataset content, including **attributes (Figure 1E)** and **granularity (Figure 1F)** to help users appropriately narrow down the search space, and *(iii)* **semantic relevance indicators (Figure 1I)** generated on-the-fly based on the user-specified task to help them assess dataset relevance rapidly.

To enable these interactions, we split DataScout's workflow across offline and online components—balancing a trade-off between semantic expressiveness and latency. We precompute embeddings, indexes, and inferred metadata where possible (e.g., for semantic dataset and attribute searches), while relying on LLMs-in-the-loop for dynamic features requiring search context (i.e., generating query reformulations, semantic filtering suggestions, and task-specific relevance indicators). This hybrid architecture allows DataScout to deliver rich, personalized assistance without prohibitive latency, reflecting a broader systems-level challenge of designing intelligent interfaces that combine responsiveness with semantic assistance.

To evaluate DataScout, we conducted a within-subjects study with 12 participants; comparing its semantic reformulation, filtering, and relevance assessment modalities with traditional keyword and semantic search interfaces (Section 7). We find that users leveraged DataScout's features not only for more structured and intentional navigation of the dataset search space, but also as implicit feedback mechanisms—helping them reflect on their queries, make sense of individual datasets, and better understand the overall search landscape. Overall, we make the following contributions:

- Design considerations for semantic dataset discovery interfaces, derived from prior work and our formative study (n = 8);
- Design and implementation of DataScout, a dataset discovery tool to proactively steer users towards desirable datasets; and
- Empirical findings from a within-subjects user study (n = 12) demonstrating how users uniquely leverage DataScout's suggestions and assistance for sensemaking.

2 Related Work

DataScout builds on research in information seeking theories, dataset discovery interfaces, and web search tools.

Information Seeking Models and Interfaces. Information seeking has a long history of theories and successful interfaces [18]. Traditional information seeking theories describe iterative cycles of query specification, examination of results, and reformulation, until the need is satisfied [37, 52]. Other classical models conceptualized this as information foraging [44], where users follow "information scents" across content "patches." This framework was then extended to encompass a subsequent stage of "sensemaking," the process of synthesizing and contextualizing information [45]. Sensemaking helps users understand what they are finding along the way and contextualize it with their own objectives [2, 50]. In our context, these models underscore the intertwined nature of exploration and sensemaking—where users refine goals and progressively discover dataset characteristics "along the way." Ideal dataset discovery systems must guide users to: *(i)* formulate their query to narrow down to the correct subset of the search space, and *(ii)* contextualize the surfaced search results with their analytical intents and assess their relevance. These information seeking models have notably shaped web search systems. Modern web search interfaces support keyword search, auto-suggestions, related query suggestions, and empower users to filter results based on attributes and facets like time and file type [32, 33, 54, 59].

Recent work on web search and information retrieval continues to build on these foundations. Palani et al. [43] show that users' objectives evolve through inspecting search results, particularly as they gather more information about a new problem area with ill-defined information seeking goals. This is relevant in dataset search, since users may still be learning domain-specific vocabulary and assessing possibilities in early stages—as opposed to knowing precise datasets of interest upfront. Tools like Sensecape and CoNotate also provide suggestions for web search queries grounded in the user's context to close information gaps [43, 57], while other recent work explores how to best support the sensemaking process in a lightweight in-context manner [30, 31, 39]. Luminate uses an LLM to generate structured "dimensions" of design spaces for creative exploration [56]. While these papers show the value of LLM-driven reformulations, unconstrained reformulations can derail the dataset search experience and erode user confidence by yielding queries that have no matching datasets. Instead, DataScout's reformulations are informed by search results: ensuring that each reformulation covers a subset of the results, and is diverse—thereby being grounded in actual dataset availability.

LLM-generated relevance indicators have also shown promise. Liu et al. [35] find that users benefit when systems surface decision-relevant cues aligned with criteria previously found helpful for decision-making. DataScout extends this idea by generating dynamic, query-specific dataset relevance indicators. Koesten et al. [27] identify key dimensions users assess during search for dataset suitability. These include data distributions, granularity, quality, possible questions the data can answer, and creation details. DataScout surfaces relevance cues aligned with these dimensions to support dataset sensemaking.

In recent years, conversational search has emerged as a new search paradigm, leveraging clarifying questions as mixed-initiative probes to iteratively refine user intent [38, 46, 47, 61, 63]. This paradigm has been adapted by dataset search tools like Olio [51] and MetaM [14]. However, these methods still rely on users to identify and formulate their dataset requirements as queries or questions, providing limited proactive guidance to them.

Dataset Search: Challenges and Recommendations. Dataset search poses unique challenges, distinct from traditional web search. Users span a range of expertise and goals, where in many cases the goals (e.g., training a machine learning model) are far removed from the datasets. The datasets themselves are often hard to peruse manually. Despite advances in interpreting natural language intents, users still struggle with incomplete and inconsistent metadata [11, 55], expressing information seeking needs as structured search constraints [29], and assessing dataset relevance [28, 55]. These challenges lead users to face gulfs of execution (difficulty articulating intents to dataset search interfaces) and evaluation (difficulty interpreting if the system perceived their search intent, and if it is reflected by the surfaced datasets) [40].

A recent survey by Hulsebos et al. [22] further highlights how data practitioners rely on trial-and-error search refinements to overcome these barriers, calling for interfaces that better support iterative search refinement and focus on users' analytical goals. Recent work also emphasizes the need for better query assistance, dynamic metadata filters, and clearer descriptions to aid sensemaking [64]. We build on these papers by conducting a formative study (Section 3.1) that directly observes users' search workflows in modern dataset search interfaces to identify pain points and inform the design of DataScout.

Dataset Search: Mechanisms and Objectives. Popular dataset search tools employ various approaches to retrieve relevant datasets, in both the input dataset space and the underlying search mechanisms. Repositories such as Kaggle and HuggingFace support keyword-based search over dataset descriptions. Others use semantic approaches—for example, Google Dataset Search indexes datasets from repositories and individual web pages, and uses semantic matching [6, 55]. Databricks Search and Snowflake Universal Search combine keyword and semantic search [58, 62]. However, these systems typically offer static metadata filters, lack support for iterative exploration by helping users reformulate questions, and provide no cues for *why* a given dataset matches a query.

Dataset search spans two separate stages [8]: *(i)* task-based dataset search—finding an initial dataset for a given task; and *(ii)* join/union dataset search—enriching an already-identified dataset via dataset joins or unions. The former is driven by keyword or semantic queries, while the latter uses an input table targeted for enrichment. For task-based search, recent efforts focus on scalability, privacy, and efficiency [3, 7, 13, 20]. For join/union dataset search, recent efforts identify semantically equivalent attributes for "joins", or aligned schemas for "unions" to enrich the previously identified dataset [3, 10, 12, 14, 21, 26, 34], but do not focus on interface design. Overall, qualitative findings from multiple studies highlight that task-based dataset search remains largely unsupported [22, 29]. With DataScout, we aim to address this gap by exploring proactive interfaces for task-based search.

Perhaps most closely related to our work is Olio [51], a semantic question-answering system that surfaces datasets by combining natural language queries with dynamically generated and pre-authored visualizations. Olio enhances exploratory search by letting users scan visualizations to assess dataset relevance. We build on the semantic dataset search approach adopted by Olio and redirect our focus on iterative—and proactive—query refinement: guiding users to progressively explore the search space as they learn about the underlying data. Unlike Olio, which assumes a predefined question for which a visualization exists in the data, we support the iterative process of discovering the search space and task requirements.

3 Design Considerations for DataScout

To identify dataset discovery workflows that could benefit from assistance with the challenges noted in prior work (Section 2), we conducted a formative study with 8 participants (F1–F8), and identified four design considerations (DC1–DC4) for DataScout.

3.1 Formative Study

Participants were recruited via: *(i)* contacting a mailing list of data science professionals maintained by our research group, *(ii)* messaging on Slack and Discord channels with data science, ML, and AI graduate students, and *(iii)* posting to X. All participants voluntarily participated in the study and agreed to have their screen-sharing sessions recorded for transcription and analysis. Table 1 reports participant background and formative study tasks.

Participants took part in a 40-minute contextual inquiry session via Zoom. We began with a round of introductions, and observed participants perform a dataset search task of their choice with any preferred tool(s) (Table 1), as they thought-out-loud about their actions. We concluded by asking clarifying questions and gathering open-ended feedback on their dataset search experiences. This study received approval from our Institutional Review Board (IRB).

We analyzed transcripts supplemented with notes documenting participant actions. Two authors performed reflexive thematic analysis through open coding of the transcripts, notes, and screen recordings, followed by identifying axial codes [4, 5]. The authors subsequently performed a second iteration to refine themes and motivate design considerations for DataScout.

3.2 Findings

Here, we present our findings, identifying challenges in how users express and reformulate their dataset search intents, while attempting to assess dataset suitability and the underlying dataset landscape. We further highlight design considerations (DCs) stemming from these insights in-situ.

Table 1: Formative study of participants' backgrounds, tasks, and choice of platforms.

ID	Background	Task	Platform(s)[1]
F1	HCI, AI Research	Collections of web-service URLs	Perplexity, Google Dataset Search
F2	ML Engineer	Game actions data for emulations	HuggingFace
F3	Data Analyst	Pharmaceutical drug marketing	Kaggle, Google Dataset Search
F4	Art & technology	Art History and Provenance data	Kaggle, Artsy Genome
F5	ML Engineering	Populating a data lake	Kaggle
F6	Bioinformatics	RNA Sequences for Epilepsy	GEO, Google Dataset Search
F7	AI Code-Gen	Code performance benchmarks	Papers with Code
F8	Marine Science	Land use for Clean Energy	Census Data

[1] *Platforms spanned semantic-based (Perplexity, Google Dataset Search), keyword-based (Kaggle, GEO, Census Data, HuggingFace, Artsy Genome), and hybrid (Papers with Code) dataset search mechanisms.*

Users do not express search criteria due to the fear of missing out on potentially-relevant datasets. Participants had several implicit relevance criteria which were not specified to dataset search platforms. For instance, when looking for datasets to train a classifier on misinformation, F4 wanted their dataset to have as many features (columns) as possible, and while looking for a collection of URLs of web-services belonging to varied economic sectors, F1 wanted the dataset to have at-least 1000 rows. On the other hand, when F1 switched from using Google Dataset Search to Perplexity, they explicitly mentioned their preference for "1000+ rows" in their prompt. While such criteria could be specified as filters, participants preferred to keep their search open-ended to avoid filtering out potentially useful datasets.

(DC1) Expression of Free-Form Intent
Enable users to express varied facets of their analytical and dataset search intents in as much detail as desired, without significantly constraining the volume of dataset search results.

Users desire dataset content-based filtering after initial rounds of sensemaking. Several participants wanted to filter datasets based on their content (F1, F4–F6, F8), that *"simply cannot be specified to the interface"* (F2). Filtering based on content such as attributes (columns) and data granularity (rows) is not supported by present-day dataset search interfaces, as also identified by Hulsebos et al. [22]. F5 mentioned that even if the system did support searching or filtering by column names, they would run into a *"schema misalignment"* problem, defining it as *"datasets using different vocabulary to refer to the same concepts,"* and elaborated using an example from movie datasets—*" datasets can have different column names for the movie title, such as 'title', 'movie name', or 'movie title,' making it impossible to apply filters."* F3 and F8 wanted to filter datasets based on data granularity, e.g., drug-specific sales records, as opposed to pharmaceutical brand-level sales for F3; and latitude/longitude-level spatial resolution, as opposed to region names for F8.

Further, participants incrementally developed an understanding for desirable attributes they wanted to be present in their data as they inspected dataset search results, echoing the findings of Palani et al. [42]. For instance, after looking through top search results for LLM-code generation benchmark datasets, F7 realized that most datasets do not contain the prompt provided to the LLM to generate code, and expressed the need have the "prompt" column in all dataset results. F4 articulated this as an instance of *"recognition over recall,"* i.e., having to recognize the need for specific attributes or data granularity after initial sensemaking of search results—as opposed to consciously acknowledging them from the get-go.

(DC2) Semantic Dataset Content-based Filtering
Provide users the agency to identify and place fine-grained attribute (column) and granularity (row) semantic filters at the dataset content level, rather than just the dataset description.

Lack of query-specific dataset relevance indicators slows-down dataset discovery. Traditional dataset search tools failed to offer indications of relevance to the query beyond the dataset title and preview, number of downloads, and column distribution histograms to users. Some participants vocalized challenges with having to read long data descriptions to identify any caveats, and oftentimes realized critical limitations of the data after having downloaded it and spent significant amounts of time to perform exploratory data analysis (EDA) (F1, F2, F5–F8). In contrast, we observed F1 using Perplexity[1] to enlist dataset sources along with contextualized explanations for how a given dataset might fit their needs—helping them assess dataset suitability. Additionally, multiple participants frequently questioned why the surfaced datasets in the search results were relevant to their search query, especially for semantic search engines like Google Dataset Search (F1, F3, F4, F6, F8). F8 brought up feedback mechanisms provided by Google's traditional web search, such as the bold-font highlighting of matched terms—helping them infer how the search result is relevant to their query—and pointed out their absence in dataset search tools.

(DC3) Dataset Suitability Assessment
Facilitate sensemaking of dataset relevance and result inclusion criteria in context of the user-specified search query and filters.

Irrelevant or overly selective dataset search results halt query iteration. As users of semantic dataset search systems lacked transparency on dataset inclusion criteria, they were frequently confused by irrelevant search results, blocking them from iterating over or reformulating their query (F1, F3, F6, F7). On the other hand, users of keyword-search platforms expressed frustration with overly selective search results (F2, F3, F4, F8).

For instance, F4's search query to look for "historical artworks with images" yielded only 4 search results, none of which were related to art history. In such cases, participants engaged in the well documented trial-and-error query reformulation workflows to widen their scope [37]—while still failing to identify relevant datasets. Prior work has also identified how gauging the dataset search space is overwhelming for users [22].

(DC4) Guide Query Reformulation
Bridge the gap between search queries and underlying dataset landscape to overcome overly selective or irrelevant results.

We shaped DataScout with the derived design considerations (DC1–DC4). In the following section, we present a walkthrough of DataScout's key features and capabilities.

[1]an AI-powered search engine and chatbot: https://www.perplexity.ai/

4 Walkthrough of DataScout

Here, we provide a walkthrough of DataScout with Dana, a journalist, who has been inspecting the world happiness reports spanning 2015–2025.[2] She wishes to observe the impact of fine-grained lifestyle changes on the reported aggregate happiness scores. To do so, Dana decides to focus on datasets overlapping with the COVID-19 pandemic—to observe the impact of stark differences in lifestyles (e.g., confinement, reduced physical activity, and remote work and education) on happiness scores.

Dana now turns to DataScout to search for datasets. Since this is a new area of exploration for her, she begins by using the ***Getting Started card*** (Figure 2A), where she specifies her intent as a regression analysis task, while expressing her query in natural language as "datasets indicating quality of life before, during, and after the COVID-19 pandemic" (supporting **DC1**). In response, DataScout surfaces search results and proactively inspects them to identify pertinent themes. For Dana's query, DataScout learns that the search results spanned shifts in inflation, social media trends, and employment patterns. DataScout then uses these insights to propose three ***query reformulation suggestions*** (Figure 2B) centered around Dana's task, in an attempt to bridge the gap between her query and the underlying dataset search space (supporting **DC4**). The suggestions help Dana by providing her inspiration for analytical directions she can pick. She hovers over each suggestion to inspect explanations for the suggested queries, and the number of datasets matching the theme. She selects the suggestion: "analyze the impact of the pandemic on remote work and work-life balance," since it is an evident indicator of happiness owing to sudden transformations in work patterns during the pandemic. DataScout refreshes the search results.

As Dana inspects the datasets, she realizes the need for three additional requirements. First, since Dana mentioned the pandemic in her query, DataScout's ***task-specific relevance indicators*** (Figure 2E) surface the data collection time-period for each dataset she explores. This reminds her to look for datasets where the time-range of data collection overlaps with the 2015–2025 year bracket. DataScout's semantic relevance indicators allow her to quickly glean this information, helping her efficiently identify data sources that align with her intent (supporting **DC3**).

Second, DataScout inspects all search results and identifies attributes most relevant to Dana's query—surfacing them as ***semantic column concept filters*** (Figure 2C). Observing suggestions for 'hours,' 'vacations,' and 'stress' help Dana realize that she wants to have these attributes in her target dataset. To only focus on datasets with quantitative measures like logged work hours, Dana applies the semantic column concept filter to narrow down the results (supporting **DC2**). Third, as she continues to inspect datasets, she realizes that to make meaningful comparisons with the world happiness reports, she needs the geographical granularity of her data to be country-level. To do so, she uses DataScout's ***semantic geo-granularity filter*** (Figure 2D), setting "country" as the data granularity level (supporting **DC2**). Dana applies these filters and continues to iteratively evaluate dataset suitability.

[2]The World Happiness Report is an annual publication that ranks countries based on how happy their citizens perceive themselves to be. URL: https://worldhappiness.report/

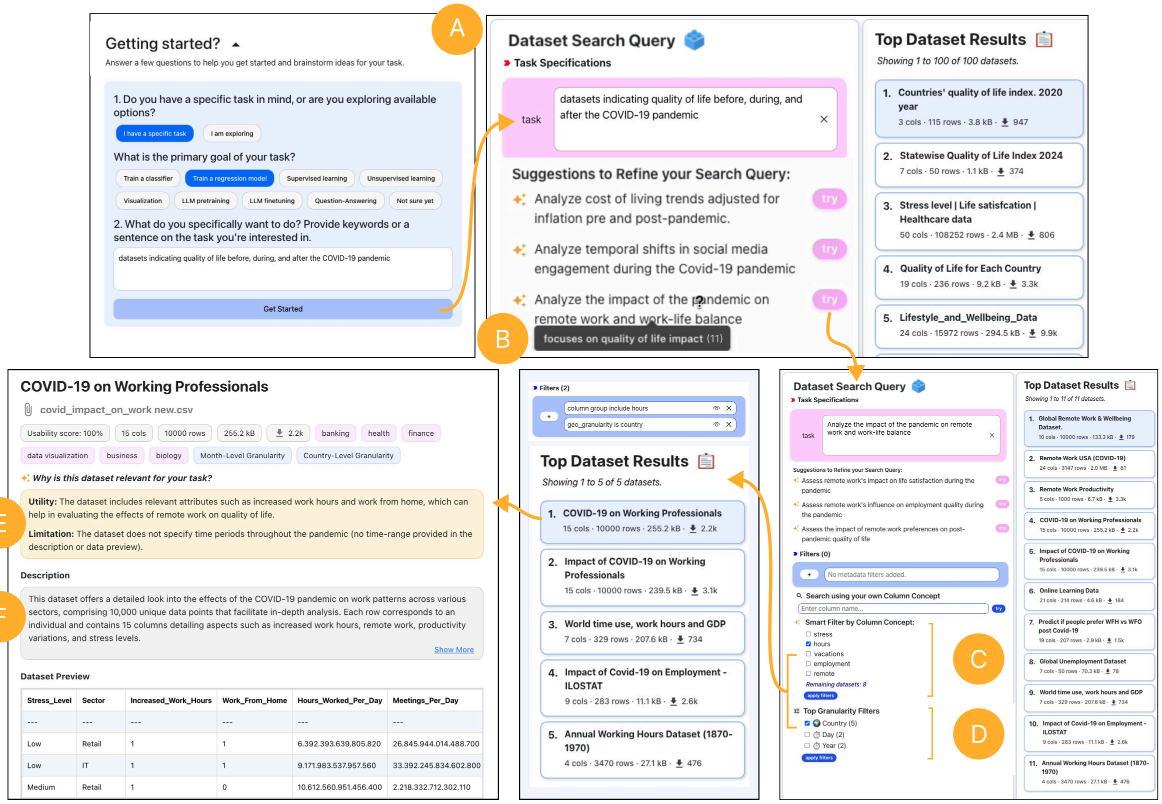

Figure 2: Walkthrough of DataScout. Dana expresses her intent using the (A) getting started card. DataScout retrieves results. Dana reviews (B) query reformulation suggestions and hovers to view explanations. She clicks on the third suggestion—refreshing the results. Dana uses semantic (C) attribute and (D) granularity filter suggestions to narrow her search to datasets containing logged employee hours and country-level data. She inspects dataset relevance using (E) dynamic task-specific relevance indicators, and (F) dataset description summaries.

5 DataScout: System Implementation

DataScout is implemented as a web-based application using React and TypeScript for the frontend, with a backend powered by Python, Flask, and a PostgreSQL database of datasets fetched from Kaggle, detailed in the next section. In addition to DataScout's features that proactively support and aid semantic dataset search, it also includes a few standard features found in Kaggle and Google Dataset Search, including: ranking of datasets based on semantic relevance; dataset pages with metadata, description, and a preview; and filters over size, shape, title, description, and tags/keywords.

DataScout's design distributes the workload across offline and online stages of interaction. Offline, we precompute embedding collections (i.e., compressed semantic representations) and build indexes for dataset and attribute (or column) search (Figure 3). Then, online, to enable contextualized assistance grounded in the user's search query and surfaced dataset search results, DataScout relies on LLM-in-the-loop workflows (Figure 4)—generating: *(i)* query reformulation suggestions; *(ii)* semantic data content-based attribute and granularity filter suggestions; and *(iii)* dataset relevance indicators. This hybrid architecture enables DataScout to avoid prohibitive latencies, while still providing in-situ and personalized

Table 2: Offline data collection with downstream uses.

Collected Metadata	**Used For**
• Title + filename + tags	Dataset Cards (Fig. 1H)
• Dataset Size	Dataset Cards (Fig. 1H)
• Number of downloads	Dataset Cards (Fig. 1H)
• Dataset Description	Dataset Embeddings, Dataset Cards (Fig. 1H)
• Dataset Sample (10 rows)	Dataset Embeddings, Attribute Embeddings, Dataset Cards (Fig. 1H)
Generated Metadata	**Used For**
• Description summaries	Purpose Embeddings, Dataset Cards (Fig. 1H)
• Attribute descriptions	Attribute Embeddings, Dataset Cards (Fig. 1H)
• Data source/collection	Dataset Cards (Fig. 1H)
• Granularity tags	Granularity Filters (Fig. 1F), Dataset Cards (Fig. 1H)
• Dataset purposes	Purpose Embeddings
Precomputed Values	**Used For**
• Dataset Embeddings	Dataset Index for semantic dataset search (Fig. 1A)
• Attribute Embeddings	Attribute Index for search & filtering (Fig. 1D, E)
• Purpose Embeddings	Query reformulation suggestions (Fig. 1B)

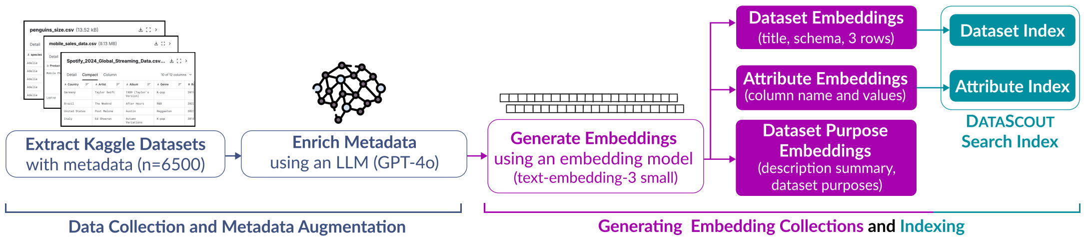

Figure 3: Offline dataset collection, augmentation, embedding generation and indexing for DataScout.

assistance. In the following subsections, we detail our offline data collection and indexing stages, and online feature-specific implementation details.

5.1 Offline Data Collection and Indexing

Figure 3 and Table 2 provide an overview of our data collection, preprocessing and indexing pipeline. We collected datasets from Kaggle using their API, obtaining over 6,500 unique tables (belonging to over 3150 datasets—where each dataset contained one or more tables within). For each table, we extracted metadata, including: title, filename, description, tags, dataset size, number of rows and columns, usability score, number of downloads, and a sample of 10 rows with headers, formatted as a markdown table. To standardize and enrich the available metadata, we used OpenAI's `gpt-4o-mini` model to generate: *(i)* concise one-line dataset summaries using descriptions from Kaggle (**DC3**), *(ii)* column descriptions and inferred data types (**DC3**), *(iii)* data source and collection methods (**DC3**), *(iv)* temporal and spatial granularity by looking at example rows (**DC2**), and *(v)* the set of purposes or use-cases the dataset might support (e.g., regression, classification, visualization, or temporal analysis) (**DC3, DC4**). The prompts to generate these additional dataset metadata are in Appendix B.

Then, to support the previously identified design considerations, we generated three different sets of embeddings[3] using OpenAI's pre-trained `text-embedding-3-small` model.

- **Dataset Embeddings:** Using the dataset title, header, and three example rows as embedding inputs, to support semantic dataset search (**DC1**).
- **Attribute Embeddings:** Using the column name and the first 10 non-null values as embedding inputs, to support attribute-level filtering (**DC2**).
- **Dataset Purpose Embeddings:** Using the previously generated dataset description summary and list of purposes as embedding inputs, to support proactive query reformulations (**DC4**).

We stored the augmented and pre-processed dataset collection with all generated embeddings in a PostgreSQL database. We created two HNSW indexes [36]: *(i)* a **Dataset Index** using the dataset embeddings (**DC1**); and *(ii)* an **Attribute Index** using the attribute embeddings (**DC2**), using the open-source library `hnswlib`.[4] Here, given a dataset schema (or an attribute name), the dataset (or attribute) HNSW index returns k most semantically similar datasets (or attributes).

[3]Embeddings are compressed vector representations of the data; with similarity of two embedding vectors being a proxy for semantic similarity.
[4]https://github.com/nmslib/hnswlib (with `m=16` and `ef_construction=64`)

5.2 Semantic Dataset Search Engine

DataScout leverages the search indexes (Section 5.1 & Figure 3) to support semantic dataset search (**DC1**). Figure 4 details the search framework and actions triggered by DataScout to proactively assist users. The search process begins with users specifying a search query—which may be as brief as a set of keywords, or as detailed as 2–3 sentences. DataScout uses this query to prompt `GPT-4o-mini` to generate three diverse hypothetical schemas for a target dataset that would help with the user's query (prompt detailed in Appendix C). The generated outputs include the dataset name, projected column names and types, and an example row. These hypothetical schemas capture different ways in which the user's intent might align with datasets in our collection. Each of the three generated schemas is then embedded using the `text-embedding-3-small` model, ensuring consistency with previously computed dataset embeddings (Section 5.1). To determine relevance, we compute the cosine similarity between each hypothetical dataset embedding and precomputed dataset embedding pair. Since each of the hypothetical schemas may highlight different aspects of the user's search query, we average the similarity scores obtained for each dataset in our collection for an aggregate similarity score. The datasets are then ranked based on this aggregate score to present the most semantically relevant results. Increasing the number of hypothetical schemas would increase the chances of retrieving highly relevant matches by covering a broader semantic space, but also increase computational costs and query latency. We generate three schemas to balance retrieval effectiveness and response time.

5.3 Supporting Dynamic and Contextualized Assistance

DataScout aims to leverage the semantic abilities of LLMs to facilitate contextualized dataset discovery. Figure 4 highlights DataScout's online assistance features, and the following sections provide corresponding implementation details.

5.3.1 Query Reformulation Suggestions. To support **DC4**, DataScout surfaces query reformulation suggestions to bridge the gap between user specified dataset search queries and the search space of available datasets (Figure 2B). To do so, DataScout proactively analyzes all initial dataset search results—performing k-means clustering (k = 15) over the dataset purpose embeddings (described in Section 5.1) belonging to the surfaced results—semantically grouping datasets that cover similar topics or have similar intended purposes. DataScout then picks three clusters that are most relevant to the original search query, and uses an LLM to surface three

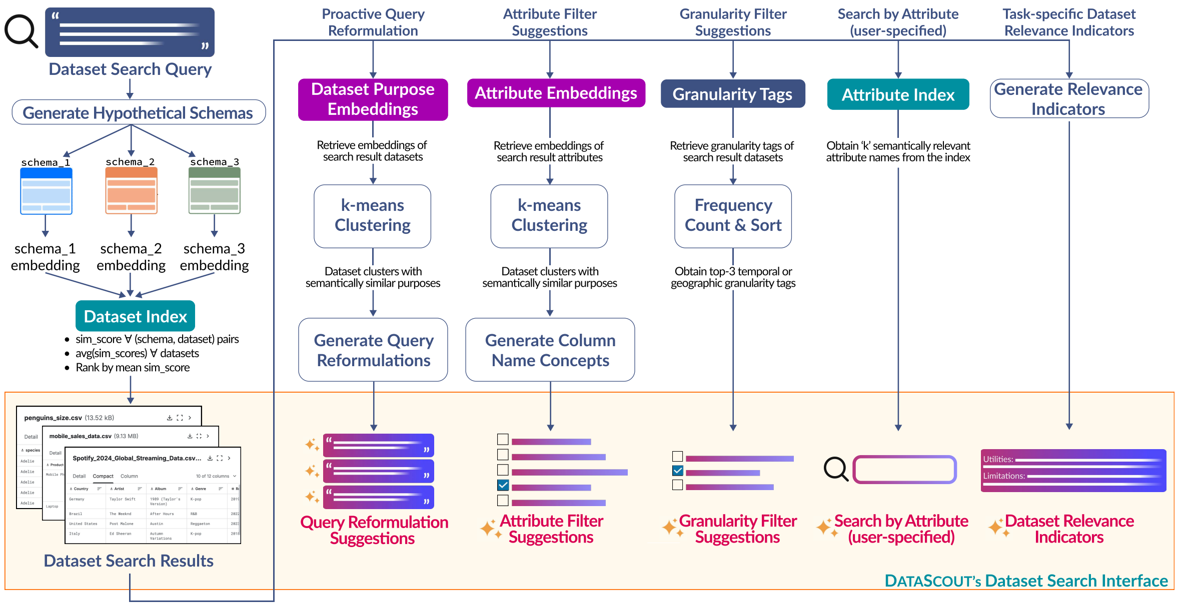

Figure 4: Online dataset search assistance. The user query is used to generate hypothetical schemas to retrieve matching datasets from the Dataset Index (Section 5.1). DataScout proactively generates query reformulations, semantic attribute and granularity filter suggestions, and dataset relevance indicators—grounded in dataset search results and the search query. Users may accept a reformulation, apply filters, search by attributes, or inspect relevance indicators.

corresponding query reformulation suggestions (prompt detailed in Appendix C), e.g., Figure 2B shows the query reformulation suggestion "analyze the impact of the pandemic on remote work and work-life balance." Users may select a query reformulation suggestion to narrow the search scope, or to increase alignment with underlying datasets. Selecting a suggestion refreshes the dataset search results.

5.3.2 Semantic Attribute Search and Filter Suggestions. DataScout introduces two unique affordances—enabling users to search and filter dataset results based on attribute semantics, instead of exact or fuzzy string matching with attribute names (**DC2**). First, DataScout gives users the agency to search by attributes (Figure 1D)—by retrieving relevant datasets based on the HNSW attribute index. That is, given an attribute name, k related attributes from the index are retrieved, and their corresponding datasets are returned, e.g., searching for "movie name" will return all datasets containing attributes semantically equivalent to movie titles. Second, DataScout proactively suggests five "column concepts" as filters—informed by both the dataset search results, as well as the user's search query—to narrow down the search space. To do so, DataScout performs k-means clustering ($k = 15$) over the attribute embeddings (described in Section 5.1) belonging to the datasets in surfaced results and grouping together semantically equivalent attributes. DataScout then computes a mean vector for each embedding cluster, and computes its cosine similarity with the user's search query. Finally, DataScout leverages LLM assistance to assign a concept name to the five most relevant attribute clusters, and surface these as filter suggestions (prompt detailed in Appendix C), e.g., `[stress, hours, vacations, employment, remote]` (shown in Figure 2C).

With these approaches, users may effectively isolate datasets matching attribute-level specifications even if their search terms do not exactly match with column names in a given dataset (**DC2**).

5.3.3 Semantic Granularity Filter Suggestions. As detailed in Section 5.1, we augmented our collection of datasets with LLM annotations on temporal (e.g. second, minute, hour, ..., year) and spatial (e.g. latitude/longitude, street address, zipcode, ..., country) granularity (**DC2**). DataScout also proactively inspects search results to recommend the three most frequently seen **temporal** and **spatial** granularity tags as filters (Figure 2D). Users may select a filter to view datasets at the required resolution and level of detail.

5.3.4 Dynamic Dataset Relevance Indicators. To assist users in assessing dataset suitability, DataScout uses LLM assistance to provide in-situ relevance feedback by generating dynamic explanations for dataset utilities and limitations on-the-fly (Figure 2E). To do so, DataScout considers the user's search query and applied filters, and leverages LLM assistance to generate utility and limitation indicators for the top-5 search results (prompt detailed in Appendix C); while relying on lazy-evaluation for the remaining search results, i.e., generating the relevance feedback only if the user clicks on the dataset search result for further inspection. Once generated, all relevance indicators are persisted for future visits to a dataset, unless the user modifies their search query or applied filters.

6 User Study: Methodology

To understand how users might leverage DataScout's proactive assistance, we conducted a within-subjects repeated-measures study

Table 3: Participant background and study tasks.

ID	Order	Background	Tasks
P1	B-C-A	Data Provenance	Neighborhood Migrations in the US
P2	B-A-C	(F4) Art & AI	Art History and Provenance Data
P3	A-C-B	Databases Researcher	Fraud Detection via ITR
P4	C-B-A	Data Scientist	Question-Answering for LLM-Eval
P5	A-C-B	Data Analyst	Smart-location Sensor Streams
P6	A-B-C	Data Science Graduate	Entity Resolution for Categoricals
P7	C-B-A	(F8) Marine Scientist	Land use for Clean Energy
P8	C-A-B	(F2) AI/ML Engineering	Game Actions Data for Emulations
P9	C-A-B	Business Analyst	Business News Pre-training Data
P10	B-C-A	(F5) ML Engineering	Populating data lake w/ restaurants
P11	B-A-C	Software Developer	Top rated movies and TV Shows
P12	A-B-C	Finance Data Analyst	Financial Inclusion Indicators

with 12 participants. Our study was guided by the following research questions:

(RQ1) How do DataScout's features guide users to discover their target datasets? (Section 7.1)
(RQ2) How do DataScout's capabilities support users' data discovery and sensemaking workflows? (Section 7.2)

Recruitment. We recruited 12 participants by emailing prior formative study participants, and through a mailing list of data science professionals maintained by our research group. Four participants overlapped with our formative study (F2 as P8, F4 as P2, F5 as P10, and F8 as P7). All participants had expertise in data science and analytics. They voluntarily consented to taking part in the study, and agreed to have the sessions recorded for transcription and analysis. To maintain ecological validity, participants were asked to bring an open-ended dataset search task of personal relevance, reported in Table 3. Participants used the same task across all study conditions to allow for consistent comparisons [18].

Procedure. We conducted a within-subjects repeated-measurements study to facilitate direct comparisons across three conditions:

(A) **Kaggle Dataset Search**: Baseline supporting keyword search (Appendix A, Figure 5)—chosen for being representative of traditional keyword dataset search tools, as well as for providing a relatively direct comparison standpoint—as DataScout's dataset collection is derived from Kaggle;
(B) **Semantic Baseline**: A stripped-down version of DataScout supporting only semantic search and static metadata filters (Appendix A, Figure 6), chosen for an experience representative of semantic search tools like Google Dataset Search and Olio's semantic dataset retrieval [51]; and
(C) **DataScout**: Complete version with semantic search, query reformulations, filtering, and relevance indicators (Figure 1).

Conditions were presented in a randomized order to mitigate learning effects. Participants were also reminded of their original task before each condition to help re-anchor their search, and minimize task drift or carryover from prior conditions and experiences. While participants used the same dataset search task (of their choosing) across conditions, there were no fixed "correct" target datasets to be found. Dataset search, like exploratory data analysis, is inherently open-ended [15, 49]. Participants pursued different exploratory trajectories depending on the condition and its affordances, and a dataset search was considered successful if the participant identified one or more datasets to pass their initial round of inspections, and judged them as promising for further investigation. The study began with a brief round of introductions and demographic questions. Each session lasted 60 minutes, during which participants spent 15–18 minutes per condition. Participants were encouraged to think aloud. After each condition, we asked follow-up questions to assess the perceived ease of use of the interface and the relevance of the search results. Participants remotely accessed and controlled a MacBook equipped with an Apple M2 chip, 8GB RAM, and a 10-core CPU. We found the average latency to retrieve datasets to be 1.6 seconds. Suggestions for proactive assistance streamed into the interface within up to 12 seconds.

Since our system indexed 6,500 datasets from over 50,000 public datasets on Kaggle, we wanted to ensure that participants are not severely restricted by our subset of most popular datasets. To ensure that the semantic baseline and DataScout had access to relevant datasets, we augmented our initial dataset collection by indexing 300 additional datasets, containing top 25 Kaggle dataset search results for each participant's task. All participants were informed of this dataset scope. To avoid biasing participants, no system walkthrough or tutorial was provided—enabling us to glean their raw impressions and organic usage patterns. This study was approved by our Institutional Review Board (IRB).

Analysis. We transcribed all sessions using Zoom's automatic transcription and supplemented them with detailed notes documenting participant actions throughout the sessions. Two authors performed reflexive thematic analysis through open coding of the transcripts, notes, and screen recordings, followed axial coding to surface broader themes. The authors subsequently performed a second iteration of axial coding to further refine the themes, and achieve high inter-rater agreement. We identified 22 open-codes and 9 axial-codes. Additionally, we analyzed the logs for learning effects across study conditions, and highlight emergent patterns in our study findings.

7 User Study: Findings

Here, we discuss our findings from observing participants engage in dataset discovery workflows across study conditions.

All participants (n=12) found DataScout's interface to be more *"expressive"* and *"flexible"*, giving them a *"greater sense of control"* over their search task. They appreciated the description summaries and consolidated single-page view—reducing context-switching and scrolling. **Participants rated DataScout highly on the ease of use of the interface on a 5-point Likert scale (μ=4.75, σ=0.45), and were mostly satisfied with the relevance of search results** (μ=3.67, σ=0.78) (Table 4). On the other hand, while using Kaggle, participants echoed sentiments in-line with our formative study findings—being unable to freely express their dataset search intents, finding it restrictive (P2–P4, P6, P10). DataScout also enabled more efficient exploration: participants explored more datasets (μ=6.02), and spent less time in assessing dataset suitability (μ=37s). They also found relevant datasets sooner (μ=5.1 mins). Overall task success was highest with DataScout (10 of 12 participants found a relevant dataset), compared to Kaggle (7 of 12) and semantic baseline (6 of 12) (Table 4).

Table 4: Task performance and subjective ratings (5-point Likert scale) across study conditions.

Condition	Ease-of-use Ratings[1]	Relevance Ratings[2]	# Queries	# Datasets Explored	Time to assess suitability (s)	Time to first target (mins)	# Successes
(A) Kaggle	μ=3.08; σ=0.51	μ=3.25; σ=1.05	μ=3.5; σ=3.6	μ=3.33; σ=1.44	μ=134; σ=47	μ=7.0; σ=5.4	7 *of* 12
(B) Semantic Baseline	μ=3.75; σ=0.45	μ=3.25; σ=0.86	μ=1.9; σ=0.6	μ=4.25; σ=1.5	μ=115; σ=28	μ=7.5; σ=5.5	6 *of* 12
(C) DataScout	μ=4.75; σ=0.45	μ=3.67; σ=0.78	μ=1.8; σ=3.4	μ=6.02; σ=2.46	μ=37; σ=12	μ=5.1; σ=1.7	10 *of* 12

[1,2] A Friedman test revealed significant differences in ease-of-use ratings across conditions (χ^2=23.13, p<0.00001), but no significant differences in relevance ratings (χ^2=4.42, p=0.11). Pairwise Wilcoxon tests showed that all ease-of-use comparisons were significant (p<0.002, C > B > A). For relevance, only a marginal difference was observed between conditions B and C (p=0.047, C > B).

Across all study sessions, participants used DataScout's query reformulation suggestions 15 times (11 of 12 participants), search and filter through column concepts 30 times (12 of 12 participants), and data-granularity filters 3 times (2 of 12 participants). We also observed differences in the perceived usefulness of DataScout's features to be dependent on the order in which participants were exposed to the conditions. When exposed to DataScout before either of the baselines, participants missed the presence of semantic attribute filters the most (P3, P4, P7, P8, P10)—which is the most used feature across sessions (30 invocations); and when exposed to DataScout after the baselines, they appreciated the presence of task-specific relevance indicators the most (P2, P6, P11, P12)—which significantly expedited participants' sensemaking and relevance judgments. To examine whether exposure to different conditions influenced user behavior, we analyzed session logs for signs of learning effects. We found that users did not fixate on previously discovered successful datasets; instead, they continued to explore and identify new ones. Notably, when users experienced the control conditions (A or B) first and then transitioned to DataScout (C), they discovered 12 unique, unseen target datasets (9 of 12 participants). Conversely, when users started with DataScout and moved to A or B, they still uncovered 7 unseen target datasets (6 of 12 participants).

On the other hand, we observed differences in dataset search workflows across conditions. First, participants wrote longer and more expressive queries with both DataScout and the semantic baseline. For example, P2 searched for *"images that are artworks with the names of the artists"* on DataScout, versus a shorter *"art history"* on Kaggle. Second, Kaggle often returned overly selective results (5–20 results), while the semantic baseline returned too many loosely relevant ones (50–100 results). In contrast, DataScout helped participants start broad with 50+ dataset results, and narrow down to 10–12 datasets effectively using semantic filters, supporting both exploratory and targeted dataset search workflows. Lastly, participants frequently downloaded datasets in the baseline conditions for deeper inspection. With DataScout, this need diminished due to in-situ feedback from relevance indicators. All participants noted the usefulness of such indicators, and 8 of 12 commented on their soundness and credibility.

In what follows, we present qualitative findings from the user study, organizing them around two key capabilities DataScout unlocked for users: first, their ability to steer and refine their search through interactive features (addressing RQ1); and second, their ability to adapt to search results and learn during exploration (addressing RQ2).

7.1 DataScout Unlocked Users' Ability to Steer and Adapt Their Dataset Search

DataScout enabled participants to adopt more deliberate and informed dataset search strategies (P1, P4, P5, P7, P8, P10, P12). Compared to the baselines, users learned to steer system feedback to their advantage (P2, P3, P6, P8, P10), and encountered learning moments that enhanced their sensemaking and search behavior—even beyond DataScout's immediate environment (P4, P6, P8, P9). We describe these distinctive strategies below.

7.1.1 Users learned to "prompt-engineer" queries to control DataScout's relevance indicators. Participants learned through interaction that the dimensions of feedback highlighted by the relevance indicators was dependent on their query and filters (P1–P3, P6, P8–P12). As they gained increased familiarity with DataScout, some participants began treating their queries as "knobs" they could use to manipulate the dataset relevance indicators (P2, P3, P6, P8–P10)—adjusting their task descriptions to elicit more targeted and informative feedback from the system. For instance, P2 needed information about image use rights for datasets containing links to artwork images. They hypothesized that modifying the query with this request would affect the relevance indicators, and added—"I need to know what the image rights are (e.g. if it is public domain, CC0, if attribution is required, etc.)." Thereafter, the relevance indicators began surfacing image licensing details for each dataset.

Similarly, P3 mentioned their preference for "non-synthetic" datasets in their query—with the objective of having relevance indicators pin-point dataset sources upfront. This contrasts with our formative study findings, where participants held unspoken dataset relevance criteria and felt restricted by the dataset search interfaces. By making relevance indicators visible and responsive, DataScout successfully elicited hidden preferences—promoting a reflective search process for other participants as well (P6, P8–P10).

7.1.2 DataScout empowered users by enabling fine-grained queries over dataset attributes and granularity levels. Participants used DataScout's features (query reformulations, and attribute and granularity filters) to systematically broaden or refine their search (P1, P4, P5, P7, P8, P10, P12). P7 began with the query: "land use in USA," which returned mostly irrelevant results, and then used DataScout's query reformulation suggestion—"land distribution across countries"— to consciously broaden the scope. This surfaced more relevant, but geographically non-localized datasets. With this broader scope, DataScout also suggested the `country-level` granularity filter, enabling P7 to narrow results back down to the desired resolution, albeit requiring some pre-processing to filter out all non-U.S. records. This tandem-use of query reformulation suggestions and semantic granularity filters exemplifies how DataScout supports exploration followed by targeted narrowing. We observed similar workflows with DataScout supporting concerted refinement efforts for P1, P5, P7, P10, and P12. Notably, each of these participants had embarked on discovering geographical data with varied levels of granularity.

Through using DataScout's semantic attribute search, participants were able to not only narrow down the search space, but also stumble across previously latent datasets (P1, P2, P12). For instance, P2 had been deeply invested in their search for art history datasets prior to our evaluation study, and described extensively using Kaggle for this task. P2 used the semantic attribute search—a new dataset search modality surfaced by DataScout—to intentionally look for datasets with the `"artist bio"` column, leading them to discover a previously unknown dataset (Carnegie Museum Collections) that was highly relevant to their work. They appreciated the system's semantic matching, noting, *"it's great that it is not only exact matching the column name but it gets the vibes."* We observe how DataScout can surface useful datasets even for other experienced participants working in familiar domains (P1, P12).

7.2 DataScout Helped Users Make Sense of Dataset Availability

Participants frequently repurposed DataScout's features to gain feedback on their queries (P1, P3, P4, P7, P8, P10, P12), build conceptual models of the search space (P4, P9, P10, P12), and sanity-check their progress (P2, P5, P7, P8, P12). Users actively interpreted DataScout's proactive reformulation and semantic filtering suggestions—turning them into implicit system feedback to reason about dataset availability, recalibrate expectations, and steer their search strategy.

7.2.1 Relevance indicators triggered "aha" moments that changed how users judged datasets. Beyond immediate task success, DataScout prompted meaningful learning moments that shaped users' dataset suitability assessment strategies. For some participants, learning moments emerged as a byproduct of expediting sensemaking through dataset relevance indicators, making connections or limitations apparent upfront. For example, P6 initially dismissed a dataset surfaced by the semantic baseline as irrelevant. However, when the same dataset appeared in DataScout, they reviewed the system's utility explanation and reconsidered its fit. The system had highlighted 'joinable' columns relevant to P6's knowledge graph task, helping them realize the applicability of the dataset. P6 noted, *"it provides reasoning and is quite responsive... it [utility indicators] helped me understand what to expect from the dataset."* This illustrates how transparent, in-context explanations can change user perceptions. P6 then continued looking for datasets with a renewed lens for dataset applicability. We observed similar patterns with P4 and P9. Notably, each of these participants' tasks were geared towards finding datasets that would serve as inputs to algorithms they have authored themselves—offering some flexibility in how the dataset or their algorithm can be adapted to each other.

Interestingly, for one participant (P8), the LLM generated relevance indicators enabled a learning moment by filling an information retrieval need. P8 began with a clear objective: "predicting NBA game outcomes based on LaMello Ball's three-point shots." While reviewing a dataset from 2008–2014, DataScout's relevance indicators surfaced a limitation: `"LaMelo started playing for Charlotte Hornets in 2020, while the time-span of this dataset predates LaMelo's NBA career."` This insight helped P8 quickly rule out the dataset and refine their assessment criteria for the remainder of the study—while carrying this learning over to Kaggle, where they began checking dataset upload dates more deliberately.

7.2.2 Users adapted their queries when query reformulation suggestions hinted at unavailable data. Participants learned early on that the query reformulation suggestions were dependent on the search results yielded by DataScout (P1, P3, P4, P6–P8, P10, P11). Some used these suggestions to verify whether their queries contained enough detail (P1, P7), while others used them to make bets on the presence of relevant datasets, probe the search space, and adapt their expectations (P3, P4, P8).

For instance, P3 originally searched for non-synthetic money transfer datasets on Kaggle. However, DataScout and baseline did not have any real-world money transfer datasets as part of their dataset collection, leading to irrelevant results based on synthetic sources. This mismatch led them to question the reliability of the results: *"I started to lose faith in the results and their ranking"*. However, the reformulation suggestion `"Analyze anomalies in real-world income tax datasets"` hinted at not only the absence of money transfer datasets, but the abundance of real-world income tax anomaly datasets; helping P3 pivot their task to income tax datasets—realigning their goals to match the available search space. Other participants refined their geographic or demographic focus without changing their broader goals. For example, P12 used reformulation suggestions to scope financial inclusion data down to agricultural workers in Rwanda.

Relevance indicators also played a role in helping participants evaluate the viability of their queries (P4, P9, P10, P12). When one or more top-ranked datasets indicated `"No significant utilities"` (highly ranked datasets showing poor task adherence)—prompted participants to reformulate their queries.[5] On facing this conflict, P10 said, *"No significant utilities higher up in the search results means that I should change my query, seems like there is not a lot in the search space to begin with."*

7.2.3 Seeing the "right" semantic filter suggestions gave users confidence they were on track. Participants also experientially learned that the suggested semantic attribute and granularity filters depended on the search results (P2, P5, P7, P8, P12). Over time, these filter suggestions became feedback signals or sanity checks that participants used to validate their current direction. Seeing the *"right"* filter suggestions reassured participants that they were on the right track, and within their intended space of dataset search results. For instance, P12 noted, *"Seeing* `[agriculture, income, credit]` *is*

[5]While participants in our formative study also encountered irrelevant top-ranked results in using semantic dataset search engines (like Google Dataset Search), they typically skipped to the next entry without reflecting on the mismatch between ranking and task relevance. We believe that DataScout's relevance indicators prompted users to re-express intent, enabling more iterative and reflective searching. We hypothesize that the presence of relevance indicators but facilitate ***meta-cognition***—helping users reason not only about what they see, but also about their next steps, as discussed in the Cognitive Fit theory by Vessey [60].

affirmative of my intent—it tells me I am still in the right space." In contrast, when filter suggestions seemed off, participants interpreted that as a sign to revise their query. P5, searching for "intergenerational facilities," initially saw unrelated filters like `[emissions, source, insurance, url]`, prompting them to rethink their query phrasing. After revising the query, more aligned filters appeared, such as `[daycare, address, age, cost]`, reinforcing their revised direction.

Similarly, P7 said, *"I see emissions, energy, land, population, and water, along with a year-level filter suggestion. This is giving me confidence that your system is understanding my prompt correctly."* P8 also supported our observation, mentioning how these acted as early cues: *"even before I look at the search results, the smart column filters are giving me some clue about the kind of data in the search results."* DataScout's semantic filters suggestions served as both, conceptual scaffolds, and lightweight progress markers during open-ended search tasks.

8 Discussion

We reflect on our findings in context of sensemaking and information-seeking literature, and discuss opportunities to extend DataScout.

8.1 Impact of Relevance Indicators on Sensemaking

Our findings show how DataScout supported sensemaking through relevance indicators, helping users assess dataset suitability (see Section 7.2). We interpret these findings through Kaur et al. [24]'s framework on sensible AI explanations, which emphasizes understanding not just the content of explanations, but their cognitive timing and alignment with user goals as well.

*Relevance indicators support **Identity Construction** by affirming users' intents.* Relevance indicators helped users quickly identify datasets that aligned with their stated goals and preferences. Echoing Kaur et al. [24], we found that participants gravitated towards cues that affirmed their own reasoning—using them to either confidently shortlist datasets, or skip them without further inspection—speeding up their workflow (as seen in Table 4).

*Relevance indicators disrupt **Retrospective Sensemaking**.* Kaur et al. [24]'s framework argues that offering explanations before users have had a chance to reflect on information themselves negatively affects their sensemaking. In our case, DataScout immediately surfaces task-specific relevance indicators upon inspecting a dataset—often leading to quick decision-making. Surfacing such cues too early sometimes disrupted users' independent judgment of dataset suitability, and short-circuited their exploratory and sensemaking processes. Complementary to this argument, P2 and P5 voiced concerns about the subjectivity in LLM interpretations, preferring to view the *"raw data"* and *"hard cold facts,"* over *"narratives around the data."* This skepticism echoes prior work on interactive ML systems, where Groce et al. [16] observed users heavy reliance on visible system cues while remaining wary of subjective or opaque feedback.

Yet, users also wanted more visible and persistent indicators (P1, P8).[6] These opposing reactions reflect a fundamental tension: if surfaced too early, sensemaking aids can overly steer users; if surfaced too late, they may lose their utility altogether; as also discussed by Amershi et al. [1]. Future dataset search systems must negotiate this tradeoff carefully, perhaps by layering relevance signals across interaction stages and interface elements.

8.2 Operationalizing Structured Exploration

While DataScout supports dataset discovery through NL intent expression, participants expressed a need for more structured control over their query's interpretation—specifically, the ability to specify binary constraints in NL, rather than loose preferences. This reflects a common tension in semantic search: while NL offers flexible intent expression, it can blur the line between strict filters and preferences, limiting users' ability to precisely steer their search. Participants envisioned interfaces to distinguish between constraints and preferences—P9 suggested separate input fields for the two, prompting reflection on search goals; while P2 proposed an adaptive mechanism that can automatically treat criteria as constraints when results are too broad, and as preferences when too narrow—mirroring the ***Information Diet Model***, where users must balance preferences (easy-to-catch prey) and rigid constraints (hard-to-catch-prey) to optimize search [45].[7] Prior work in exploratory search has emphasized supporting both fluid and rigid filtering modes as well [18, 19, 37, 48].

A complementary direction involves expanding DataScout's query reformulations beyond their current role of narrowing results. Reformulations could also broaden the search space by introducing adjacent and semantically related results, helping users consider alternatives they may not have explicitly articulated, thus supporting robust exploration through both—structured narrowing and expanding of the dataset search space.

8.3 Limitations and Future Work

Our evaluation of DataScout has several limitations. First, the search precision was constrained by our collection of Kaggle datasets, occasionally producing irrelevant results despite augmenting our corpus with ~300 datasets for participant tasks. Second, our prototype lacked basic search functionalities such as result sorting and support for varied ranking criteria (e.g. upload date, downloads, size), which limited participants' ability to explore results systematically. Third, we recorded only two observations per condition order, limiting findings on experiential effects. Finally, we compared only with Kaggle as a keyword-search baseline due to our shared dataset sources, and lacking access to other deployed systems with similar data. This choice allowed for direct comparisons, but narrowed our evaluation scope.

We suggest several directions for extending DataScout. First, users desired a *"birds-eye view"* (P7) summarizing patterns across results—such as covered time periods or geographic regions—to expedite sensemaking and offer feedback on their queries (Section 7.2).

[6] P8 suggested a simple, persistent thumbs-up/down mechanism; and P1 wanted always-on relevance indicators to avoid clicking on each dataset for further inspection.
[7] *"If a predator is too specialized, it will do very narrow searching. If the predator is too generalized, then it will pursue too much unprofitable prey" [45]*

Aggregated overviews, as explored by Ouellette et al. [41], could support this need by presenting bottom-up hierarchical summaries of results. Second, users often wanted to combine data from multiple sources to construct their intended dataset (F2, F5, F7, F8)—via union or joins. While prior work has addressed union/join-based dataset search, future interfaces could better support this with tailored sensemaking tools and visual cues for multi-dataset compositions. Finally, participants wanted visibility into data quality (P4, P10, P12). Building on existing efforts in data quality detection and wrangling [9, 17, 23, 53], future systems could surface these cues as relevance indicators to better inform user decisions.

9 Conclusion

We introduce DataScout—a system that rethinks dataset discovery through proactive AI-assistance, offering query reformulation suggestions, semantic search and filtering based on attributes and data granularity, and task-specific dataset relevance indicators—supporting users in navigating and understanding opaque dataset landscapes. Our study with 12 participants revealed how these features expedited sensemaking and conceptual model building; while eliciting latent search specifications. Our findings also underscore the need for dataset search systems to be designed to support both, exploratory wandering and targeted retrieval—meeting users where they are in their evolving dataset search workflows.

Acknowledgments

We are grateful to HC Moore, Yiming Lin, Sepanta Zeighami, James Smith, and Hila Mor for their valuable feedback on our prototypes and findings. We thank our study participants for their engagement and feedback, helping us identify constructive search workflows and unmet needs for dataset search systems. This work was supported by the National Science Foundation (grants DGE-2243822, IIS-2129008, IIS-1940759, and IIS-1940757), the Dutch Research Council (NWO, grant NGF.1607.22.045), funds from the State of California, an NDSEG and BIDS-Accenture Fellowship, funds from the Alfred P. Sloan Foundation, as well as EPIC Lab sponsors (Adobe, Google, G-Research, Microsoft, PromptQL, Sigma Computing, and Snowflake).

References

[1] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–13.

[2] Marcia J Bates. 1989. The design of browsing and berrypicking techniques for the online search interface. *Online review* 13, 5 (1989), 407–424.

[3] Alex Bogatu, Norman W Paton, Mark Douthwaite, and André Freitas. 2022. Voyager: Data discovery and integration for data science. In *Proceedings 25th International Conference on Extending Database Technology (EDBT 2022)*.

[4] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[5] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health* 11, 4 (2019), 589–597.

[6] Dan Brickley, Matthew Burgess, and Natasha Noy. 2019. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In *The world wide web conference*. 1365–1375.

[7] Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. 2021. Auctus: A dataset search engine for data augmentation. *arXiv preprint arXiv:2102.05716* (2021).

[8] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. 2020. Dataset search: a survey. *The VLDB Journal* 29, 1 (2020), 251–272.

[9] Bhavya Chopra, Anna Fariha, Sumit Gulwani, Austin Z Henley, Daniel Perelman, Mohammad Raza, Sherry Shi, Danny Simmons, and Ashish Tiwari. 2023. Cowrangler: Recommender system for data-wrangling scripts. In *Companion of the 2023 International Conference on Management of Data*. 147–150.

[10] Mahdi Esmailoghli, Christoph Schnell, Renée J Miller, and Ziawasch Abedjan. 2023. Blend: A unified data discovery system. *arXiv preprint arXiv:2310.02656* (2023).

[11] Grace Fan, Jin Wang, Yuliang Li, and Renée J Miller. 2023. Table discovery in data lakes: State-of-the-art and future directions. In *Companion of the 2023 International Conference on Management of Data*. 69–75.

[12] Grace Fan, Jin Wang, Yuliang Li, Dan Zhang, and Renée Miller. 2022. Semantics-aware dataset discovery from data lakes with contextualized column-based representation learning. *arXiv preprint arXiv:2210.01922* (2022).

[13] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1001–1012.

[14] Sainyam Galhotra, Yue Gong, and Raul Castro Fernandez. 2023. Metam: Goal-oriented data discovery. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2780–2793.

[15] Saul Greenberg and Bill Buxton. 2008. Usability evaluation considered harmful (some of the time). In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 111–120.

[16] Alex Groce, Todd Kulesza, Chaoqiang Zhang, Shalini Shamasunder, Margaret Burnett, Weng-Keen Wong, Simone Stumpf, Shubhomoy Das, Amber Shinsel, Forrest Bice, et al. 2013. You are the only possible oracle: Effective test selection for end users of interactive machine learning systems. *IEEE Transactions on Software Engineering* 40, 3 (2013), 307–323.

[17] Philip J Guo, Sean Kandel, Joseph M Hellerstein, and Jeffrey Heer. 2011. Proactive wrangling: Mixed-initiative end-user programming of data transformation scripts. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 65–74.

[18] Marti Hearst. 2009. *Search user interfaces*. Cambridge university press.

[19] Marti A Hearst. 2006. Clustering versus faceted categories for information exploration. *Commun. ACM* 49, 4 (2006), 59–61.

[20] Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. *arXiv preprint arXiv:2103.12011* (2021).

[21] Zezhou Huang, Jiaxiang Liu, Haonan Wang, and Eugene Wu. 2023. The Fast and the Private: Task-based Dataset Search. *arXiv preprint arXiv:2308.05637* (2023).

[22] Madelon Hulsebos, Wenjing Lin, Shreya Shankar, and Aditya Parameswaran. 2024. It took longer than I was expecting: Why is dataset search still so hard?. In *Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics*. 1–4.

[23] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the sigchi conference on human factors in computing systems*. 3363–3372.

[24] Harmanpreet Kaur, Eytan Adar, Eric Gilbert, and Cliff Lampe. 2022. Sensible AI: Re-imagining interpretability and explainability using sensemaking theory. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 702–714.

[25] Moe Kayali, Fabian Wenz, Nesime Tatbul, and Çağatay Demiralp. 2024. Mind the Data Gap: Bridging LLMs to Enterprise Data Integration. *arXiv preprint arXiv:2412.20331* (2024).

[26] Aamod Khatiwada, Grace Fan, Roee Shraga, Zixuan Chen, Wolfgang Gatterbauer, Renée J Miller, and Mirek Riedewald. 2023. Santos: Relationship-based semantic table union search. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–25.

[27] Laura Koesten, Kathleen Gregory, Paul Groth, and Elena Simperl. 2021. Talking datasets–understanding data sensemaking behaviours. *International journal of human-computer studies* 146 (2021), 102562.

[28] Laura Koesten, Elena Simperl, Tom Blount, Emilia Kacprzak, and Jeni Tennison. 2020. Everything you always wanted to know about a dataset: Studies in data summarisation. *International journal of human-computer studies* 135 (2020), 102367.

[29] Laura M Koesten, Emilia Kacprzak, Jenifer FA Tennison, and Elena Simperl. 2017. The Trials and Tribulations of Working with Structured Data: -a Study on Information Seeking Behaviour. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 1277–1289.

[30] Andrew Kuznetsov, Joseph Chee Chang, Nathan Hahn, Napol Rachatasumrit, Bradley Breneisen, Julina Coupland, and Aniket Kittur. 2022. Fuse: In-situ sensemaking support in the browser. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–15.

[31] Andrew Kuznetsov, Michael Xieyang Liu, and Aniket Kittur. 2024. Tasks, Time, and Tools: Quantifying Online Sensemaking Efforts Through a Survey-based Study. *arXiv preprint arXiv:2411.07206* (2024).

[32] Bongshin Lee, Mary Czerwinski, George Robertson, and Benjamin B Bederson. 2005. Understanding research trends in conferences using PaperLens. In *CHI'05 extended abstracts on Human factors in computing systems*. 1969–1972.

[33] Bongshin Lee, Greg Smith, George G Robertson, Mary Czerwinski, and Desney S Tan. 2009. FacetLens: exposing trends and relationships to support sensemaking

within faceted datasets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1293–1302.
[34] Aristotelis Leventidis, Martin Pekár Christensen, Matteo Lissandrini, Laura Di Rocco, Katja Hose, and Renée J Miller. 2024. A Large Scale Test Corpus for Semantic Table Search. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1142–1151.
[35] Michael Xieyang Liu, Tongshuang Wu, Tianying Chen, Franklin Mingzhe Li, Aniket Kittur, and Brad A Myers. 2024. Selenite: Scaffolding Online Sensemaking with Comprehensive Overviews Elicited from Large Language Models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–26.
[36] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.
[37] Gary Marchionini. 2006. Exploratory search: from finding to understanding. *Commun. ACM* 49, 4 (2006), 41–46.
[38] Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Jian-Yun Nie. 2024. A survey of conversational search. *arXiv preprint arXiv:2410.15576* (2024).
[39] Meredith Ringel Morris, Jarrod Lombardo, and Daniel Wigdor. 2010. WeSearch: supporting collaborative search and sensemaking on a tabletop display. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. 401–410.
[40] Donald A. Norman. 2002. *The Design of Everyday Things*. Basic Books, Inc., USA.
[41] Paul Ouellette, Aidan Sciortino, Fatemeh Nargesian, Bahar Ghadiri Bashardoost, Erkang Zhu, Ken Q Pu, and Renée J Miller. 2021. RONIN: data lake exploration. *Proceedings of the VLDB Endowment* 14, 12 (2021).
[42] Srishti Palani, Zijian Ding, Stephen MacNeil, and Steven P Dow. 2021. The" Active Search" Hypothesis: How search strategies relate to creative learning. In *Proceedings of the 2021 conference on human information interaction and retrieval*. 325–329.
[43] Srishti Palani, Zijian Ding, Austin Nguyen, Andrew Chuang, Stephen MacNeil, and Steven P Dow. 2021. CoNotate: Suggesting queries based on notes promotes knowledge discovery. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.
[44] Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review* 106, 4 (1999), 643.
[45] Peter L. T. Pirolli. 2007. *Information Foraging Theory: Adaptive Interaction with Information* (1 ed.). Oxford University Press, Inc., USA.
[46] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval*. 117–126.
[47] Corbin Rosset, Chenyan Xiong, Xia Song, Daniel Campos, Nick Craswell, Saurabh Tiwary, and Paul Bennett. 2020. Leading conversational search by suggesting useful questions. In *Proceedings of the web conference 2020*. 1160–1170.
[48] Francesca Rossi, Kristen Brent Venable, and Toby Walsh. 2008. Preferences in constraint satisfaction and optimization. *AI magazine* 29, 4 (2008), 58–58.
[49] Daniel M Russell, Mark J Stefik, Peter Pirolli, and Stuart K Card. 1993. The cost structure of sensemaking. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. 269–276.
[50] Tony Russell-Rose and Tyler Tate. 2013. Chapter 2 - Information Seeking. In *Designing the Search Experience*, Tony Russell-Rose and Tyler Tate (Eds.). Morgan Kaufmann, 23–45. doi:10.1016/B978-0-12-396981-1.00002-1
[51] Vidya Setlur, Andriy Kanyuka, and Arjun Srinivasan. 2023. Olio: A semantic search interface for data repositories. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–16.
[52] Ben Shneiderman. 1987. *Designing The user interface: Strategies for effective human-computer interaction, 4/e (New Edition)*. Pearson Education India.
[53] skrub data. 2025. skrub: A library for data cleaning and preprocessing. https://github.com/skrub-data/skrub. Accessed: 2025-04-07.
[54] Greg Smith, Mary Czerwinski, Brian Meyers, Daniel Robbins, George Robertson, and Desney S Tan. 2006. FacetMap: A scalable search and browse visualization. *IEEE Transactions on visualization and computer graphics* 12, 5 (2006), 797–804.
[55] Katrina Sostek, Daniel M Russell, Nitesh Goyal, Tarfah Alrashed, Stella Dugall, and Natasha Noy. 2024. Discovering datasets on the web scale: Challenges and recommendations for Google Dataset Search. *Harvard Data Science Review* Special Issue 4 (2024).
[56] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. 2024. Luminate: Structured generation and exploration of design space with large language models for human-ai co-creation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–26.
[57] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling multilevel exploration and sensemaking with large language models. In *Proceedings of the 36th annual ACM symposium on user interface software and technology*. 1–18.
[58] Nitya Tarakad. 2024. A Peek Inside: How Snowflake's New Universal Search Feature Was Built. *Snowflake Builders Blog: Data Engineers, App Developers, AI/ML, & Data Science* (February 2024). https://medium.com/snowflake/a-peek-inside-how-snowflakes-new-universal-search-feature-was-built-dfd1188176d0
[59] Daniel Tunkelang. 2022. *Faceted search*. Springer Nature.
[60] Iris Vessey. 1991. Cognitive fit: A theory-based analysis of the graphs versus tables literature. *Decision sciences* 22, 2 (1991), 219–240.
[61] Alexandra Vtyurina, Denis Savenkov, Eugene Agichtein, and Charles LA Clarke. 2017. Exploring conversational search with humans, assistants, and wizards. In *Proceedings of the 2017 chi conference extended abstracts on human factors in computing systems*. 2187–2193.
[62] Chi Zhang. 2024. Adding Intelligence to Databricks Search. *Databricks Blog* (March 2024). https://www.databricks.com/blog/adding-intelligence-to-databricks-search
[63] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management*. 177–186.
[64] Yihang Zhao, Albert Meroño-Peñuela, and Elena Simperl. 2024. User Experience in Dataset Search Platform Interfaces. *arXiv e-prints* (2024), arXiv–2403.

A Screenshots of User Study Conditions

Here, we present screenshots for our baseline dataset search conditions, as detailed in Section 6: (A) Kaggle (Figure 5) and (B) Semantic Baseline (Figure 6).

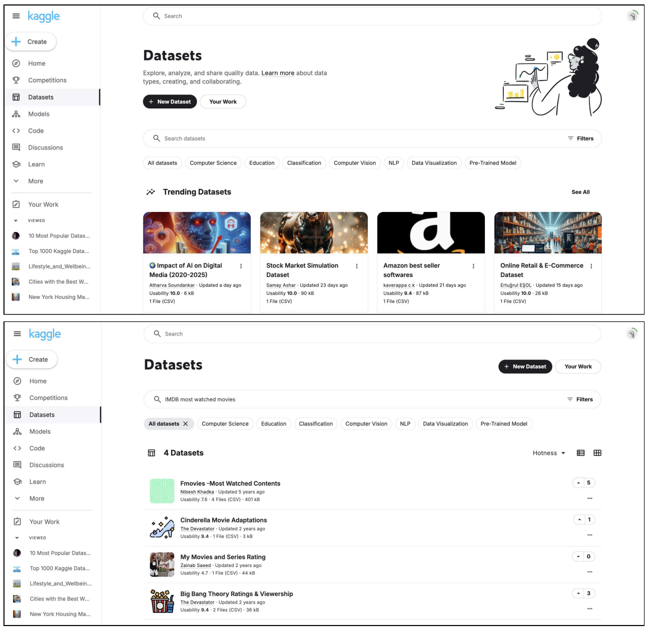

Figure 5: (A) Kaggle: Keyword dataset search condition

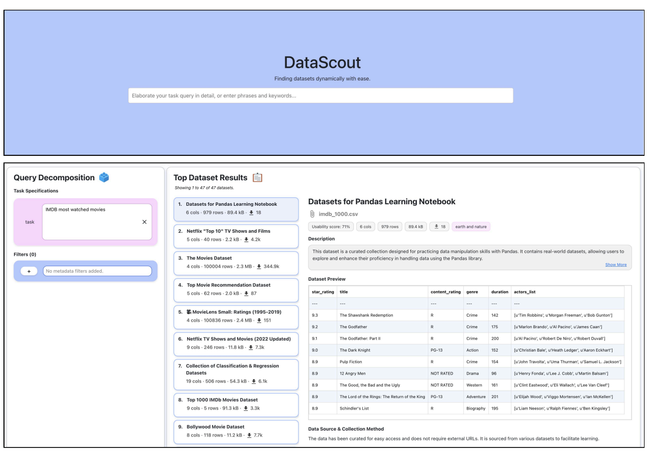

Figure 6: (B) Semantic Baseline: A stripped-down version of DataScout supporting only semantic dataset search.

B Prompts for Offline Metadata Augmentation

We designed two prompts to augment the metadata for each dataset in our collection, as described in Section 5.1 and illustrated in Figure 3. For every dataset, we prompted OpenAI's `GPT-4o` to: *(i)* enrich the metadata with summaries of the dataset's description, purpose, source and collection methods, and attribute descriptions; and *(ii)* annotate the dataset's temporal and spatial granularity. We use the LLM's tool calling functionality to generate structured responses that adhere to the output schema defined for each prompt.

Dataset Metadata Augmentation

Given following dataset details, you must extract information about this dataset.
Dataset Details:

- Title: **`{title}`**
- Description: **`{description}`**
- Dataset Preview: **`{example_rows}`**

Directly answer each question, be brief and to the point:
1. Description Summary: In 1–3 sentences, provide a brief and summarized description of the dataset.
2. Purposes: Provide a list of analytical, data science, visualization, or machine learning tasks that can be performed with this dataset. e.g., `["training a regression model", "temporal analysis"]`
3. Dataset Source & Collection Methods: Gather the source(s) of this dataset, which could include names and/or affiliations of persons, website URLs, web-APIs, synthetic sources, human annotations, and so on. If no information is available about the source of the data, output `'N/A'`.
4. Column Descriptions: For each column in the dataset, provide a brief description for the column with its data type.
Output Schema:
`{"description_summary": string, "dataset_purposes": list[string], "dataset_sources": string, "column_descriptions": list[{"column_name": string, "type": string, "description": string}] }`

Temporal & Spatial Granularity Annotation

Given a dataset with the following details, determine the most likely temporal and/or spatial granularity reflected in the dataset.
Dataset Details:

- Title: **`{title}`**
- Description: **`{description}`**
- Dataset Preview: **`{example_rows}`**

Select the **temporal granularity** from the following options: Year, Quarter, Month, Week, Day, Hour, Minute, or Second.
Select the **spatial granularity** from the following options: Continent, Country, State/Province, County/District, City, Neighborhood/Region, Zip Code/Postal Code, Street Address, Residential Address, or Latitude/Longitude.
Identify the temporal and/or spatial granularity only if reflected in the dataset. Leave the respective field(s) empty if the granularity cannot be inferred from the table.
Output Schema:
`{"temporal_granularity": string, "spatial_granularity": string}`

C Prompts for Online Dataset Search Assistance

We designed four prompts to support DataScout 's semantic search and online LLM-in-the-loop workflows, as detailed in Sections 5.2

and 5.3, and illustrated in Figure 4. These prompts take the user's dataset search query, applied filters, and the resulting datasets as inputs, enabling proactive and contextualized assistance. DataScout sends these prompts to OpenAI's `GPT-4o-mini`, once again leveraging its tool calling functionality to generate structured responses that follow the output schema defined for each prompt.

Hypothetical Schema Generation

Given the task of **`{query}`**, generate three dataset schemas to implement the task. Only generate three table schemas, excluding any introductory phrases and focusing exclusively on the tasks themselves. Generate the table names and corresponding column names, data types, and example rows. For example:
Example Task: Datasets to train a machine learning model to predict housing prices
Example Output: (Parts omitted for brevity)

```
[ { "table_name": "Properties",
"column_names": ["id", "num_bedrooms", "num_bathrooms",
"sqft", "year_built", "location", "price"],
"data_types": ["INT", "INT", "INT", "FLOAT", "INT", "TEXT",
"FLOAT"],
"example_row": [101, 3, 2, 1450.5, 2005, "Seattle, WA",
675000.0] },
{ "table_name": "NeighborhoodStats",
"column_names": [...],
"data_types": [...],
"example_row": [...] },
{ "table_name": "PropertySalesHistory",
"column_names": [...],
"data_types": [...],
"example_row": [...] } ]
```

Output Schema:

```
list[ {"table_name": string,
"column_names": list[string],
"data_types": list[string],
"example_row": list[string]} ]
```

Generate Query Reformulations

Generate a dataset search query matching a collection of given dataset names, such that it:

- Incorporates the common theme of these dataset names: **`{cluster}`**
- Relates to the original task: **`{query}`**
- Is specific enough to include both a topic, as well as a clear objective.

Also provide a brief reason (under 10 words) why this query improves upon **`{query}`**.
Example Output:

```
{ "query": "Analyze voter demographics in presidential
elections", "reason": "adds demographic focus" }
```

Output Schema:

```
{"query": string, "reason": string}
```

Generate Column Name Concepts

You are an assistant that returns a flat list of words. The input will be a list with nested elements. For each nested element, return 1 to 2 representative words that best represent the topic of the nested group. The representative word should also make sense in context with the **`{query}`**. The words should be lower case single words without special characters (like hyphens or underscores). The output must be a valid JSON array with no additional formatting, symbols, or repetitions.

Output Schema:

```
list[string]
```

Generate Relevance Indicators

You are an assistant that explains what makes the following dataset search result relevant or irrelevant, given my task and applied search filters.
Dataset Details:

- Description: **`{description}`**
- Example Rows: **`{schema}`**
- Purpose of dataset: **`{purpose}`**
- Dataset Collection Method: **`{source}`**

Dataset Search Specifications:

- Dataset search query: **`{query}`**
- Applied filters: **`{filters}`**

Instructions:
1. Utilities: Identify the strongest factors that make this dataset useful. Look for the presence of relevant attributes, high data quality, and matching intent. If there are no strong advantages, return `"No significant utilities."`
2. Limitations: Identify limitations such as missing relevant attributes, specific geographical locations (e.g., "dataset only contains records of location X"), specific temporal ranges (e.g., "data belongs to X and Y time range"), poor data quality and missing or incomplete data. If no major issues exist, return `"No significant limitations."`
Guidelines:

- Stay factual: Base responses strictly on the provided dataset details. Do not assume information that isn't explicitly stated.
- Be concise: Limit each response to 1–2 sentences.
- Avoid hallucination: If no strong reason exists for relevance or irrelevance, default to `"No significant utilities"` or `"No significant limitations"`.

Output Schema:

```
{"utilities": string, "limitations": string}
```