AI-Assisted Dataset Discovery with DATASCOUT

Rachel Lin
Bhavya Chopra
Wenjing Lin
Shreya Shankar
Madelon Hulsebos
Aditya Parameswaran

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2025-113
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-113.html

May 16, 2025

AI-Assisted Dataset Discovery with DATASCOUT

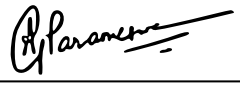
Rachel Lin

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

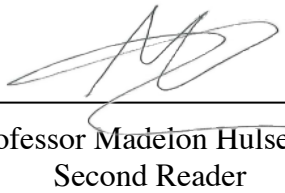


Professor Aditya G. Parameswaran
Research Advisor

May 16, 2025

(Date)

* * * * * * *



Professor Madelon Hulsebos
Second Reader

7 May 2025

(Date)

AI-Assisted Dataset Discovery with DataScout

by

Rachel Lin

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science, Plan II

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Aditya G. Parameswaran, Chair
Professor Madelon Hulsebos

Spring 2025

AI-Assisted Dataset Discovery with DataScout

Abstract

AI-Assisted Dataset Discovery with DataScout

by

Rachel Lin

Master of Science, Plan II in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Aditya G. Parameswaran, Chair

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., dataset topic and 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 and reformulation challenging. To bridge these gaps, we introduce DataScout, a tool that proactively steers 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 that are dynamically generated based on the user-specified task. A within-subjects study with 12 participants comparing DataScout to keyword and semantic dataset search tools reveals that users uniquely employ DataScout's features not only for structured dataset explorations, but also as a means to glean feedback on their search queries and build conceptual models of the dataset search space.

Contents

Acknowledgments

Firstly, I would like to thank my advisor, Aditya G. Parameswaran, for his guidance and mentorship over the past couple of years. His insightful ideas and advice have been invaluable to the projects I pursued under his supervision. I am also deeply grateful to Shreya Shankar for her continued support and for connecting me with Madelon Hulsebos, which made this project possible. I would like to give a special shoutout to Bhavya Chopra, who joined the project later but played a major role in its development. I sincerely appreciate the helpful feedback provided by my mentors—Aditya, Shreya, Madelon, and Bhavya—throughout the course of this project, which helped shape my path at UC Berkeley. Finally, I am grateful for all my friends and family as I would not be who I am without them today.

Chapter 1

Introduction

1.1 Dataset Discovery: Background and Motivation

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 [7]. This problem of *dataset search* is only growing more urgent: estimates suggest the amount of data on the internet will reach 180 zettabytes by the end of 2025 [49]—with organizations often accumulating tens of thousands of tables in their data lakes [20]. 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 [48]. Second, users rarely know exactly what they are looking for [18]. 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 [57]. For example, the Olio system [44] 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. Additionally, 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 [18]. 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.

In this thesis, we explore the design of dataset search systems that can proactively support users' iterative discovery process. In a recent research survey of data professionals (n=89), Hulsebos et al. [18] revealed that characteristics such as data granularity and freshness significantly impact search relevance judgments. While their study identifies key requirements for future dataset search systems—such as progressive refinement, hybrid querying that combines keyword and semantic approaches, task-driven search, and result diversity—several questions remain about the strategies and workarounds users employ to overcome barriers in dataset search workflows. Building on related work and the findings of Hulsebos et al. [18], we further investigate these underexplored areas in Chapter 2.

To do so, we conducted a formative study (n=8) to identify aspects of users' dataset search workflows that could be amenable to automated assistance (Section 2.2). Based on our findings, we derived the following design considerations (DCs) to guide the development of our dataset search system (Section 2.3); these are described in detail in Chapter 2.

1. **(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.
2. **(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.
3. **(DC3) Dataset Suitability Assessment** Facilitate sensemaking of dataset relevance and result inclusion criteria in context of the user-specified search query and filters.
4. **(DC4) Guide Query Reformulation** Bridge the gap between user's search queries and underlying dataset landscape to overcome overly selective or irrelevant results.

1.2 Research Goals and Contributions

We aim to address users' need for dataset search interfaces that proactively provide feedback and assistance to support iterative query reformulation, result interpretation, and navigation of the dataset search space. To this end, we introduce DataScout, a dataset search system

that leverages LLMs to actively support users throughout the dataset discovery process. By accounting for both the user's task and the dataset search space, DataScout helps surface relevant datasets and facilitates more effective exploration. An overview of the interface and an illustrative walkthrough are provided in Chapter 3.

To enable these interactions, we split DataScout's workflow across offline and online components—balancing a trade-off between semantic expressiveness and latency. We precomputed embeddings, indexes, and inferred metadata where possible (e.g., for semantic dataset and attribute searches), while relying on LLM-in-the-loop workflows for dynamic features requiring search context (i.e., generating query reformulations, semantic filtering suggestions, and task-specific relevance indicators). A more detailed outline of DataScout's architecture is provided in Chapter 4. This hybrid architecture allows DataScout to deliver rich, personalized assistance without incurring prohibitive latency, and reflects a broader systems-level challenge of designing intelligent interfaces that combine responsiveness with semantic assistance.

Finally, to evaluate DataScout, we conducted a within-subjects study with 12 participants; comparing DataScout's novel features with traditional keyword search as well as semantic search interfaces, described in Chapter 5. Our findings show 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. We additioinally evaluate the latency of various components of DataScout in Chapter 6. In summary, we make the following contributions:

1. Design considerations for semantic dataset discovery interfaces, derived from prior work and a formative study we conduct with 8 participants;
2. The design and implementation of DataScout, a dataset discovery interface to proactively steer users towards desirable dataset search results; and
3. Empirical findings from a within-subjects user study ($n = 12$) demonstrating how users uniquely leverage DataScout's suggestions and assistance for sensemaking.

This work was led by Rachel Lin and Bhavya Chopra. Other authors include Wenjing Lin, Shreya Shankar, Madelon Hulsebos, and Aditya G. Parameswaran.

Chapter 2

Related Work and Formative Study

In this chapter, we present an overview of related work. We begin by discussing models of information seeking and search interfaces, including those in the context of web search and information retrieval. Next, we summarize key challenges and recommendations from prior work on dataset search. We then review existing dataset search mechanisms and objectives, highlighting their strengths and limitations and the ways in which they informed our design. Finally, we present our formative user study—motivated by gaps in the literature—which led to four design considerations that guided the development of DataScout.

2.1 Related Work

2.1.1 Information Seeking Models and Interfaces

Information seeking has a long history of theories and successful interfaces [15]. Traditional information seeking theories can be broken down into the steps of query specification, examination of the results, and then reformulation, with the cycle repeating until the need is satisfied [32, 45]. Other classical models, such as that by Pirolli and Card [39], frame information seeking as a process of foraging, where information seekers move from one promising "patch" of information to another, based on "information scents." This framework has also been extended to encompass a subsequent stage of sensemaking where the information collected is synthesized and extended [40]—sensemaking helps users understand what they are finding along the way and contextualize it with their own objectives [43]. Other models, such as the berry-picking model [1], explicitly capture the benefits of information seeking as a series of learnings along the way, as opposed to just eventually meeting a desired information seeking target. In our context of dataset discovery, the notions of information foraging and sensemaking as intertwined processes, with progressive discovery of dataset characteristics and goals "along the way," play a key role. Ideal dataset discovery systems need to 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 been applied in various search contexts, most notably in web search. Modern web search interfaces routinely support keyword search. They also provide users feedback via query auto-suggestions and related queries—to aid rapid query reformulation—and empower users to filter results based on attributes such as time and file type, among others. Web search in other contexts, such as e-commerce or travel search, additionally support filtering based on faceted metadata [52], which are multiple orthogonal categories (or facets) predefined by the search interface designer. For example, a flight search interface might include facets such as airline, travel start time, and number of stops. Faceting has been studied in various search contexts [27, 47, 52, 28]. For dataset search, Koesten et al. [23] uncover characteristics that users commonly look for when assessing dataset suitability. These include assumptions about data distributions, granularity, quality, possible questions the data can answer, and creation details. We leverage these insights to surface LLM-generated semantic relevance indicators for each dataset, with added focus on these dimensions of dataset sensemaking.

Recent work on web search and information retrieval continues to build on the aforementioned theories. For example, Palani et al. [38] posit how users' objectives can be influenced by the search results, and their search directions evolve as they gather more information about a problem area, especially when they are exploring a new space with ill-defined information seeking goals. This finding is relevant in dataset search, since in the early stages users may still be learning domain-specific vocabulary and assessing possibilities—as opposed to knowing the precise dataset of interest upfront. Query reformulation and semantic suggestions grounded in space of surfaced search results are especially important, enabling us to bridge the gap between the user's query and available datasets. Tools like Sensecape and CoNotate also provide suggestions for web search queries grounded in the user's context to close information gaps [50, 37], while other recent work explores how to best support the sensemaking process in a lightweight in-context manner [26, 25, 34]. Additionally, Liu et al. [30] show how users of online search interfaces benefit from LLM-generated relevance indicators based on criteria they previously found helpful and referred to for decision-making. We leverage this insight to surface dynamic dataset relevance indicators adaptive to the user's query.

In recent years, conversational search has emerged as a new search paradigm, leveraging clarifying questions as promising approaches to help users refine their search intents through mixed-initiative interactions, potentially addressing the iterative nature of search tasks [54, 41, 56, 42, 33]. This approach has also been adopted by some dataset search tools like Olio [44] and MetaM [13]. However, these methods still rely on users to identify and formulate their dataset requirements as queries or questions, providing limited proactive guidance to them.

2.1.2 Dataset Search Challenges and Recommendations

Dataset search presents a number of unique challenges relative to 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 rapid advancements in understanding natural language intents, challenges persist in dataset search workflows. Various user-centered studies show how users continue to struggle with: incomplete and inconsistent metadata [11, 48], expressing information seeking needs as structured search constraints [24], and assessing dataset relevance to their task [22, 48]. When engaging in dataset search workflows, users are faced with the gulfs of execution (difficulty in translating their intentions to the dataset search interface) and evaluation (difficulty in interpreting if the system perceived their search task correctly, and if their intent is reflected by the surfaced dataset search results) [35].

More recently, Hulsebos et al. [18] conducted a survey with data practitioners and posit that in addition to the above-mentioned challenges, users employ trial-and-error search refinements to overcome barriers in search workflows. They suggest that future dataset search interfaces must more explicitly support iterative refinement, as well as focus on helping users meet analytical goals. Recent work also emphasizes the need for better query assistance, dynamic metadata filtering, and clearer descriptions to aid sensemaking [57]. We extend the research in this area by conducting a formative study (Section 2.2) by actually observing users' search workflows using modern dataset search interfaces to identify points of friction and guide the implementation of DataScout.

2.1.3 Dataset Search Mechanisms and Objectives

Popular dataset search and discovery tools employ various approaches for finding the most relevant datasets, in terms of the space of input datasets, as well as the underlying search mechanisms. Dataset repositories such as Kaggle and HuggingFace support dataset search based on matching keywords to dataset descriptions. Other tools opt for semantic approaches. For example, Google Dataset Search, which indexes datasets from repositories and individual dataset pages [5, 48] employs a semantic search approach based on dataset descriptions. Tools like Databricks Search and Snowflake Universal Search incorporate semantic search abilities in addition to supporting keyword search [51, 55]. None of these approaches go beyond a few fixed metadata filters (e.g., based on date), support iterative exploration by helping users reformulate questions, or provide hints for *why* a particular dataset was relevant to a query.

From the standpoint of objectives, recent work [7] argues that dataset search needs to cover two separate steps: task-based dataset search—identifying an initial dataset for a given task; and join and union dataset search—enriching an already-identified dataset via dataset joins or unions. For the former, the input query is a keyword search expression, while for

the latter, the input query is a table targeted for enrichment.

Recent papers have focused on various aspects of task-based dataset search [16, 2, 6, 12], such as efficiency, privacy, and scalability. Similarly, on the problem of join and union dataset search, various approaches have been proposed that identify semantically equivalent attributes for "join" operations, or aligned schemas for "union" operations to enrich the previously identified dataset [21, 9, 10, 29, 2, 17]. As a concrete example, Metam [13] employs heuristics to identify join and union datasets that maximize utility for user-specified analytical goals (such as improving prediction accuracy) to reduce manual effort needed to shortlist relevant datasets. However, they do not focus on interface elements. Overall, qualitative findings from multiple studies suggest that needs for the former step of task-based dataset search are still largely unsupported [18, 24]. With DataScout, we aim to explore the role of proactive dataset search interfaces for task-based search.

Perhaps most closely related to our work is the semantic question-answering interface, Olio [44]. Olio combines dynamically generated visualizations with a library of pre-authored data visualizations to support question-answering over dataset collections—enhancing exploratory search and enabling users to glance over visualizations to assess dataset relevance. While Olio interprets users' natural language queries to surface datasets, we aim to 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. Olio assumes a predefined question for which a visualization answer exists in the data, while we support the iterative process of discovery of task requirements and the search space. Olio, for example, does not consider filtering based on dataset content, such as the space of columns in a dataset.

2.2 Formative User Study

To identify users' dataset discovery workflows amenable to automated assistance in the context of challenges discussed by prior work (Section 2.1), we conducted a formative study with 8 participants (F1–F8), and identified four design considerations (DC1–DC4) to inform the design of DataScout.

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 (formerly Twitter), inviting participants who have searched for a dataset or a benchmark in recent past. All participants voluntarily consented to participate in the study and agreed to have their screen-sharing sessions recorded for transcription and analysis. Table 2.1 reports participant background and formative study tasks.

Each participant took part in a 40-minute contextual inquiry session via video-conferencing.

Table 2.1: Formative study of participants' backgrounds, tasks, and choice of platforms.

ID	Background	Task	Platform(s)
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.*

We began with a round of introductions, and observed participants perform a dataset search task of their own choice with any preferred tool(s) (Table 2.1), as they thought-out-loud about their actions for the remainder of the session. 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 detailed notes documenting participant actions. Two authors performed reflexive thematic analysis through open coding of the transcripts, notes, and screen recordings, followed by identifying broader themes through axial coding [4, 3]. The authors subsequently performed a second iteration of axial coding to further refine the themes and motivate design considerations for DataScout.

2.3 Design Considerations for DataScout

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.

2.3.1 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 explicitly 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 these implicit criteria could have been specified as metadata filters, participants preferred to keep their search results 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.

2.3.2 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. [18]. 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. [38]. 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.

2.3.3 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] an AI-powered search engine and chatbot, 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.

2.3.4 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 [32]—

[1] `https://www.perplexity.ai/`

while still failing to identify relevant datasets. Prior work has also identified how gauging the dataset search space is overwhelming for users [18].

(DC4) Guide Query Reformulation

Bridge the gap between user's search queries and underlying dataset landscape to overcome overly selective or irrelevant results.

Chapter 3

DataScout Interface and Walkthrough

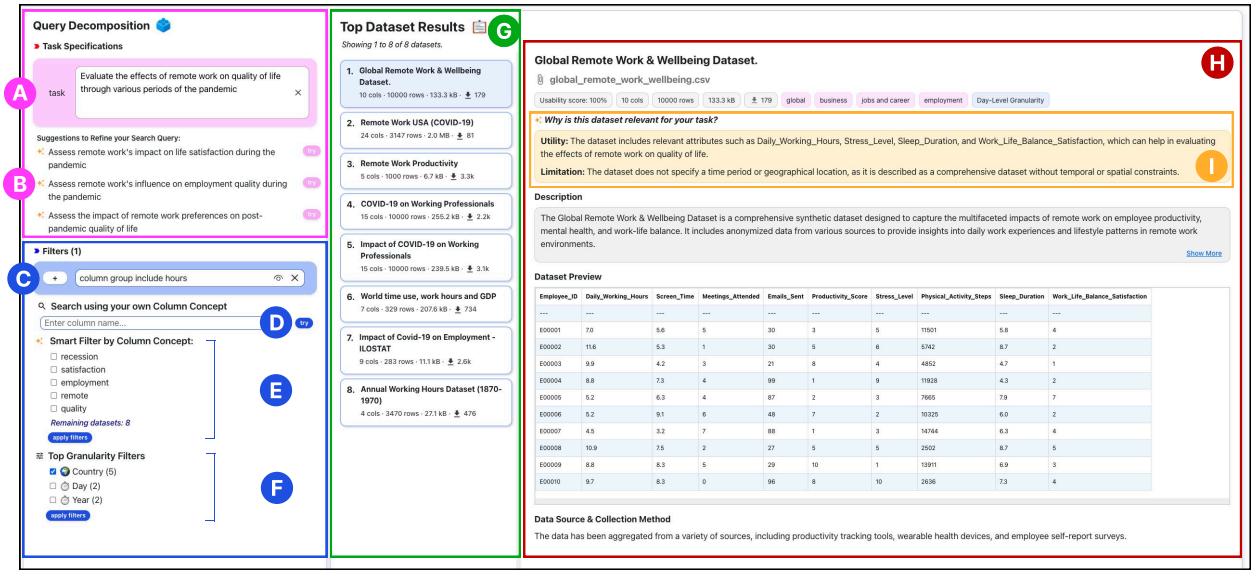


Figure 3.1: DataScout—a proactive dataset discovery interface. (A) Users begin by specifying their dataset discovery query as keywords, phrases, or complete sentences. (B) DataScout provides proactive query reformulation suggestions to bridge the gap between the user's query and datasets available in 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 the benefits and limitations of the dataset.

3.1 Interface

We present DataScout, a dataset search tool that proactively steers users through the process of dataset discovery (Figure 3.1). DataScout assists users in finding target datasets by being cognizant of both the user-specified task as well as the underlying space of dataset search results. DataScout offers three key semantic assistance features that all leverage LLMs: *(i)* **proactive query reformulation (Figure 3.1B)** to bridge the gap between users' search queries and the underlying search space, *(ii)* **semantic search (Figure 3.1D)** and **filtering** based on dataset content, including **attributes (Figure 3.1E)** and **granularity (Figure 3.1F)** to help users appropriately narrow down the search space, and *(iii)* **semantic relevance indicators (Figure 3.1I)** that are generated on-the-fly based on the user-specified task to help them assess dataset relevance rapidly.

3.2 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.[1] 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—in an attempt 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 3.2A), 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 3.2B) 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.

[1]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/`

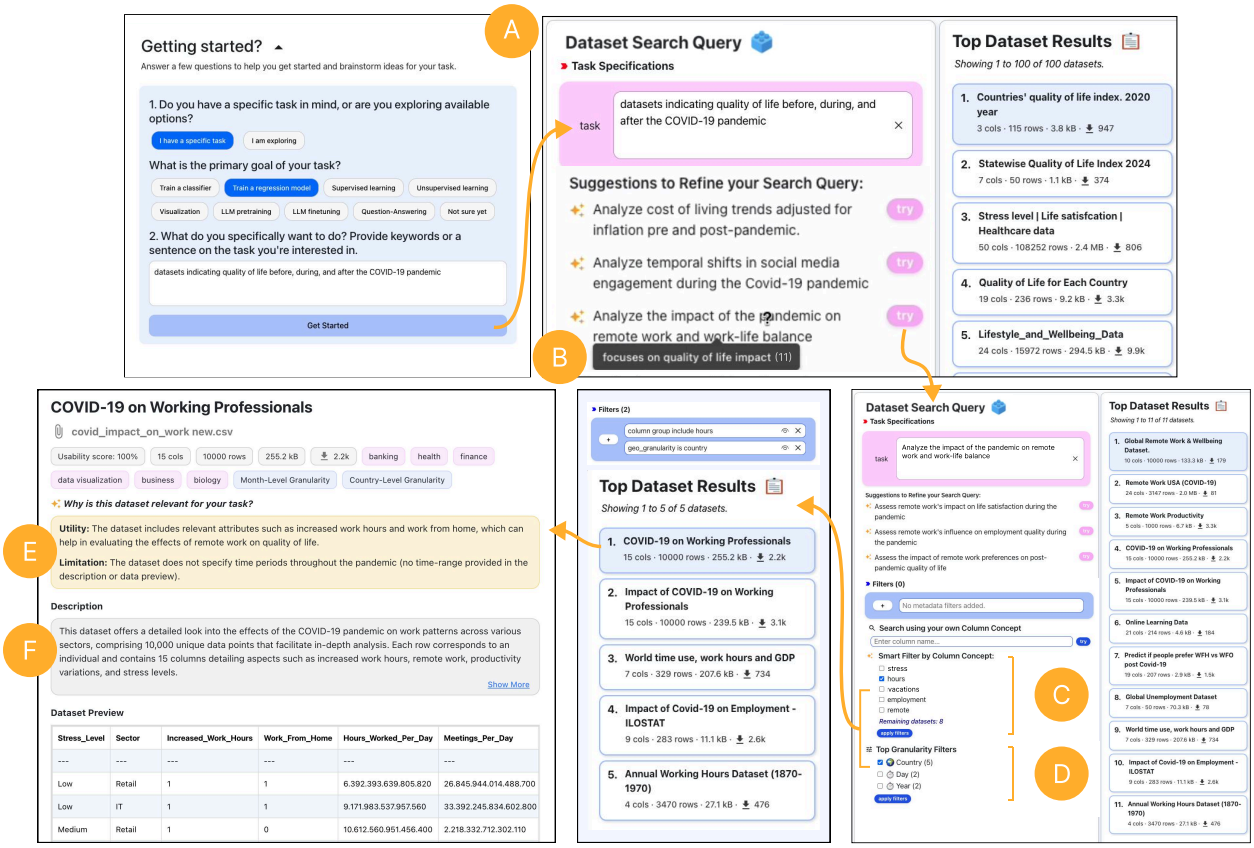


Figure 3.2: Walkthrough of DATASCOUT. Dana expresses her dataset search intent using the (A) getting started card. DATASCOUT retrieves results. Dana reviews (B) query reformulation suggestions and hovers over them to view explanations. She clicks on the third suggestion—refreshing the search results. Dana uses the semantic (C) attribute and (D) granularity filter suggestions to narrow down her search to datasets that contain logged employee hours and country-level data resolution. She inspects dataset relevance using the (E) dynamic task-specific relevance indicators, and (F) dataset description summaries.

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 3.2E) 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 3.2C). 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 3.2D), setting "country" as the data granularity level (supporting **DC2**). Dana applies these filters and continues to iteratively evaluate dataset suitability.

Having walked through how Dana interacts with DataScout to iteratively refine her dataset search, we describe the implementation of DataScout in the next chapter.

Chapter 4

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 metadata filters over dataset size, shape, title, description, and tags/keywords.

DataScout is designed and implemented such that it 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 4.1). 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 4.2)—generating: *(i)* query reformulation suggestions; *(ii)* semantic data content-based attribute and granularity filter suggestions; and *(iii)* dataset relevance indicators on-the-fly. This hybrid architecture enables DataScout to overcome prohibitive latencies and still provide in-situ and personalized assistance. In the following subsections,

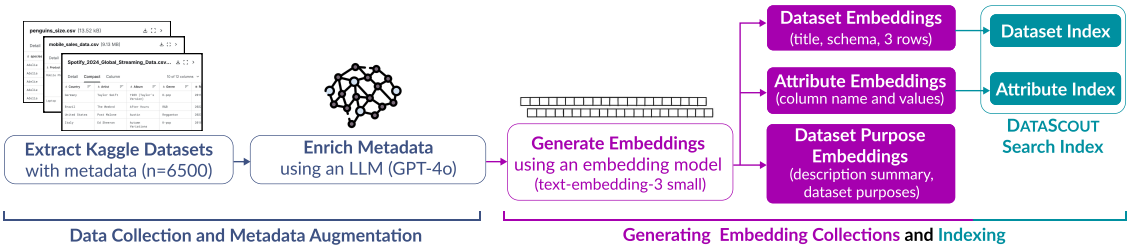


Figure 4.1: Offline dataset collection, augmentation, embedding generation and indexing for DataScout.

Table 4.1: Collected and precomputed metadata and embeddings, along with their downstream uses.

Collected Metadata	**Used For**
• Title + filename + tags	Dataset Cards (Fig. 3.1H)
• Dataset Size	Dataset Cards (Fig. 3.1H)
• Number of downloads	Dataset Cards (Fig. 3.1H)
• Dataset Description	Dataset Embeddings, Dataset Cards (Fig. 3.1H)
• Dataset Sample (10 rows)	Dataset Embeddings, Attribute Embeddings, Dataset Cards (Fig. 3.1H)
Generated Metadata	**Used For**
• Description summaries	Purpose Embeddings, Dataset Cards (Fig. 3.1H)
• Attribute descriptions	Attribute Embeddings, Dataset Cards (Fig. 3.1H)
• Data source/collection	Dataset Cards (Fig. 3.1H)
• Granularity tags	Granularity Filters (Fig. 3.1F), Dataset Cards (Fig. 3.1H)
• Dataset purposes	Purpose Embeddings
Precomputed Values	**Used For**
• Dataset Embeddings	Dataset Index for semantic dataset search (Fig. 3.1A)
• Attribute Embeddings	Attribute Index for search & filtering (Fig. 3.1D, E)
• Purpose Embeddings	Query reformulation suggestions (Fig. 3.1B)

we detail our offline data collection and indexing stages, and online feature-specific implementation details.

4.1 Offline Data Collection and Indexing

Figure 4.1 and Table 4.1 provide an overview of our data collection, preprocessing and indexing pipeline. We collected datasets from Kaggle using the Kaggle API, obtaining over 6,500 unique tables (belonging to over 3150 Kaggle 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 extracted 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, temporal analysis, etc.) (**DC3, DC4**). The prompts used to

generate these additional dataset metadata are detailed below.

Dataset Metadata Augmentation Prompt

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 and 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}] }
```

Then, to support previously identified design considerations, we generated three different sets of embeddings[1] 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**).

[1]Embeddings are compressed vector representations of the data; with similarity of two embedding vectors being a proxy for semantic similarity.

- **Dataset Purpose Embeddings:** Using the previously generated dataset description summary and list of purposes as embedding inputs, to support proactive query reformulations (**DC4**).

Lastly, we stored the augmented and pre-processed dataset collection with all generated embeddings in a PostgreSQL database. We created two HNSW indexes [31]: *(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`.[2] Here, given a dataset schema (or an attribute name), the dataset (or attribute) HNSW index returns k most semantically similar datasets (or attributes).

4.2 Semantic Dataset Search Engine

DataScout leverages the search indexes (Section 4.1 & Figure 4.1) to support semantic dataset search (**DC1**). Figure 4.2 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 below). 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 4.1). 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 at the same time. We generate three hypothetical schemas to balance retrieval effectiveness and response time.

[2] `https://github.com/nmslib/hnswlib` (with `m=16` and `ef_construction=64`)

Hypothetical Schema Generation Prompt

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]} ]
```

4.3 Supporting Dynamic and Contextualized Assistance

DataScout aims to leverage the semantic abilities of LLMs to facilitate contextualized dataset discovery. Figure 4.2 highlights DataScout's online assistance features, and the following sections provide corresponding implementation details.

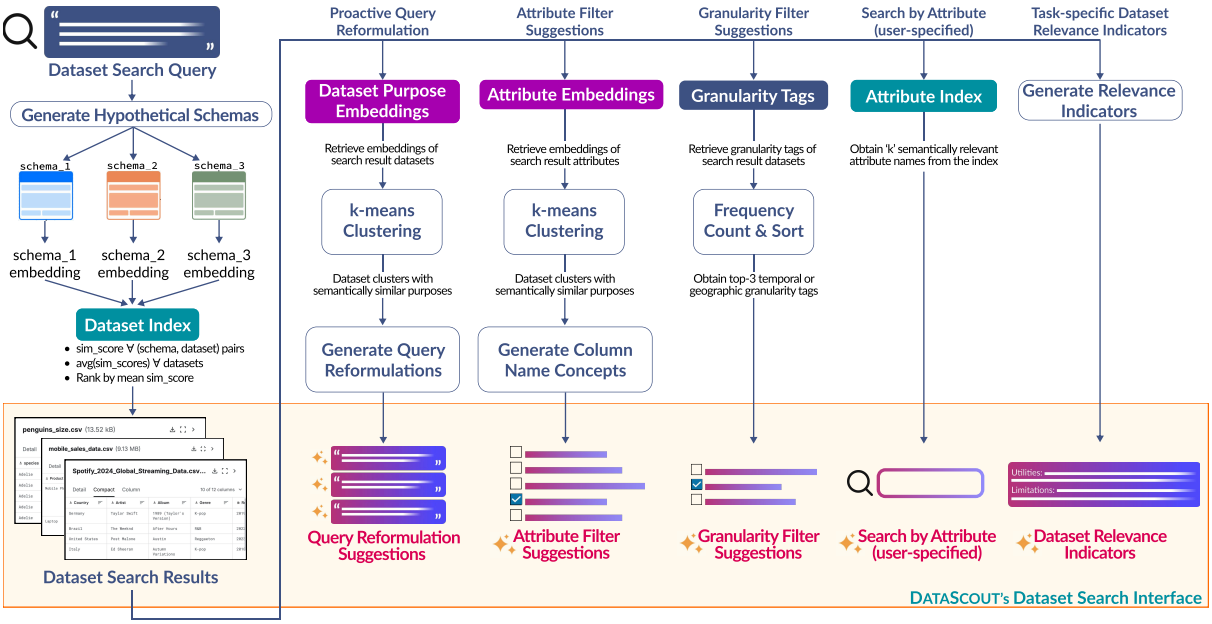


Figure 4.2: DATASCOUT's online dataset search assistance. User's search query is used to generate hypothetical schemas projecting a target dataset. Search results are displayed through retrieval from the Dataset Index (Section 4.1). DATASCOUT proactively generates query reformulation suggestions, semantic attribute and granularity filter suggestions, and dataset relevance indicators—all grounded in the dataset search results and the user's search query. Users may choose to accept a query reformulation suggestion, apply filters, search by attributes, and inspect relevance indicators.

4.3.1 Proactive Query Reformulation Suggestions

To support **DC4**, DataScout surfaces query reformulation suggestions in an attempt to bridge the gap between user specified dataset search queries and the search space of available datasets (Figure 3.2B). 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 4.1) belonging to the surfaced results—semantically grouping together datasets that cover similar topics or have similar intended purposes. DataScout then picks three clusters that are most relevant to the user's original search query, and uses an LLM to surface three corresponding query reformulation suggestions (prompt detailed below), e.g., Figure 3.2B 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 leads DataScout to refresh dataset search results.

Generate Query Reformulations Prompt

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}`

4.3.2 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 3.1D)—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 4.1) 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 below), e.g., `[stress, hours, vacations, employment, remote]` (shown in Figure 3.2C).

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**).

Generate Column Name Concepts Prompt

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]`

4.3.3 Semantic Granularity Filter Suggestions

As detailed in Section 4.1, 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 of datasets (**DC2**). DATASCOUT also proactively inspects search results to recommend the three most frequently seen **temporal** and **spatial** granularity tags as filters to users (Figure 3.2D). Users may select a filter to view datasets at the required resolution and level of detail.

Temporal & Spatial Granularity Annotation Prompt

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`}

4.3.4 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 3.2E). 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 below); 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.

Generate Relevance Indicators Prompt

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`}

Chapter 5

User Evaluation

To understand how users might leverage DataScout's semantic search, filtering, and relevance assessment features, we conducted a within-subjects repeated-measures study 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 5.4.1)
- **(RQ2)** How do DataScout's capabilities support users' data discovery and sensemaking workflows? (Section 5.4.2)

5.1 Participants

We recruited 12 participants by emailing formative study participants for a follow-up study, and through a mailing list of data science professionals maintained by our research group. Four formative study participants (F2 as P8, F4 as P2, F5 as P10, and F8 as P7) took part in the evaluation study. Once again, all participants had expertise in data science and analytics. All participants voluntarily consented to taking part in the study, and agreed to have the sessions recorded for transcription and analysis. Participants were asked to provide details on a search task they would like to perform during the study in the sign-up form. Table 5.1 reports participants' backgrounds and their self-chosen study tasks.

5.2 Procedure

We conducted a within-subjects repeated-measurements study with three conditions:

(A) **Kaggle Dataset Search**: Baseline supporting keyword search (Figure 5.1)—chosen for being representative of traditional keyword dataset search tools, as well as for pro-

Table 5.1: 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

viding 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 fixed metadata filters (Figure 5.2), chosen for an experience representative of semantic search tools like Google Dataset Search and Olio's semantic dataset retrieval [44]; and

(C) **DataScout**: Complete version with semantic search, query reformulation, filtering, and assessment features (Figure 3.1).

Participants completed their dataset search task using all three conditions in a randomized order to take experiential learning effects into consideration. We recorded two observations for each ordering. The study began with a round of introductions and demographic questions. Each session was 60 minutes long, with participants spending 15–18 minutes per condition. They were encouraged to think-out-loud. We asked them follow-up questions about their satisfaction with search results and ease of use of the interface upon the completion of each study condition. We marked a dataset search to be successful when the participants expressed a given dataset to pass their initial round of inspections, or expressed interest in downloading a dataset for further exploration.

Since we built our search engine by indexing a subset of 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

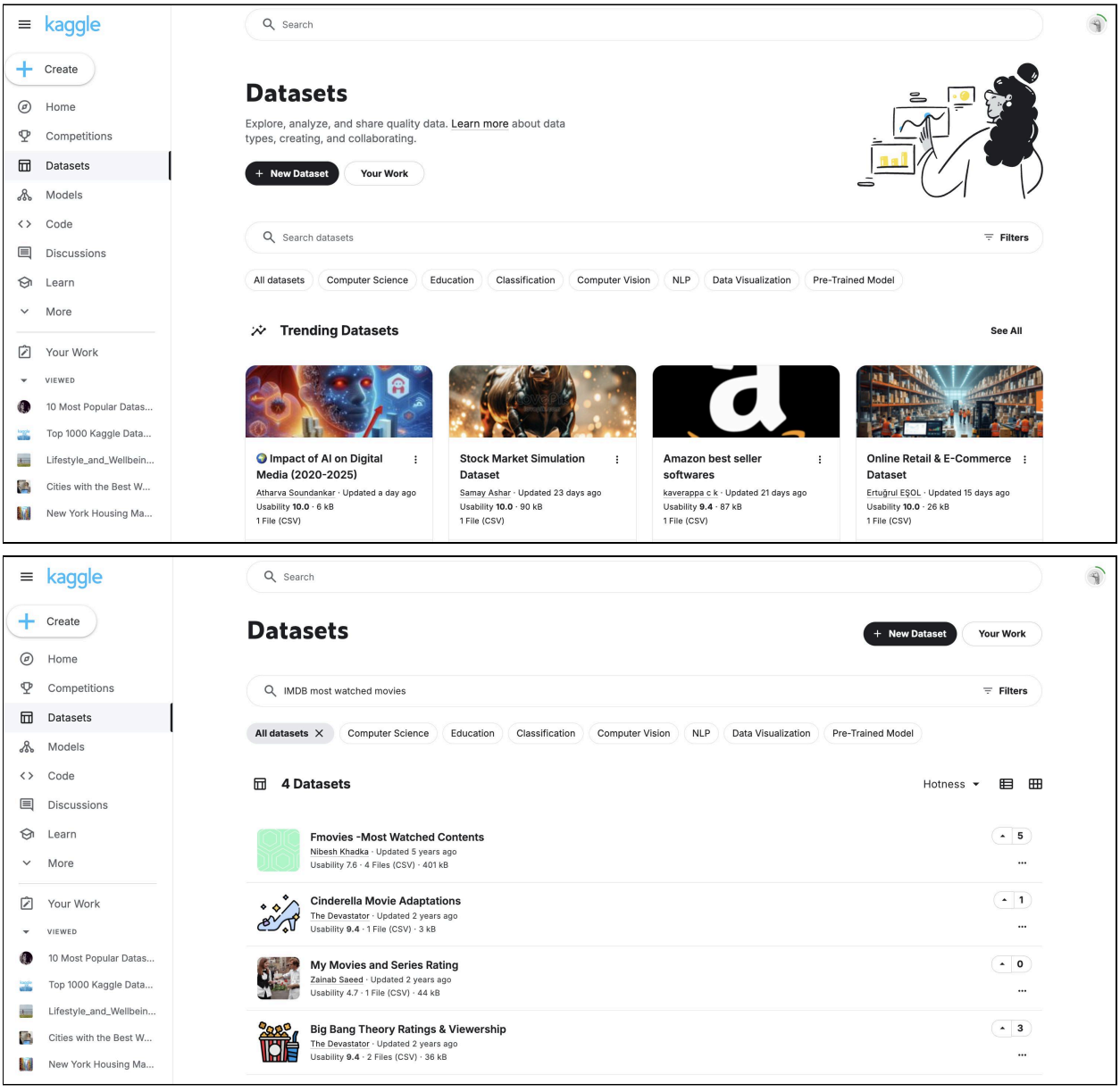


Figure 5.1: (A) Kaggle: Keyword dataset search condition

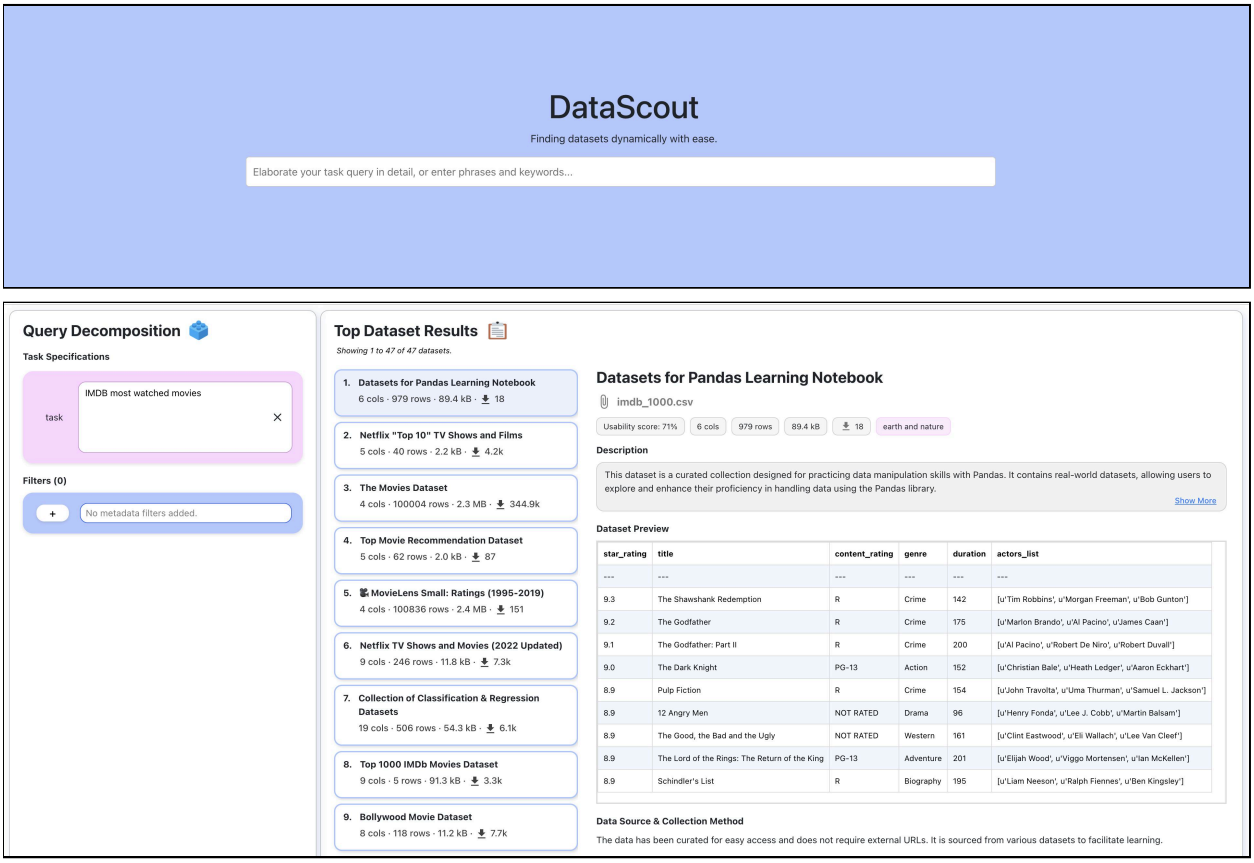


Figure 5.2: (B) Semantic Baseline: A stripped-down version of DataScout supporting only semantic dataset search.

task. All participants were made aware of this limitation, and we purposely did not provide participants with any interface or system walkthrough to glean their raw impressions. This study was approved by our Institutional Review Board (IRB).

5.3 Analysis

We used Zoom's automatic transcription feature to capture session dialogues, which we supplemented with detailed notes documenting participant actions throughout the sessions. Two authors performed thematic analysis through reflexive open coding of the transcripts, notes, and screen recordings, followed by identifying broader themes through axial coding. 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 derived 9 axial-codes. Additionally, we discussed the effects of experiential learning across study conditions, and analyzed emergent patterns due to the underlying order in which participants

Table 5.2: Task Completion and Ratings on 5-point Likert Scale for Ease-of-use and relevance of dataset search results for each study condition.

Condition	Ease-of-use	Relevance	# Successes
(A) Kaggle	μ=3.08; σ=0.51	μ=3.25; σ=1.05	7 *of* 12
(B) Semantic Baseline	μ=3.75; σ=0.45	μ=3.25; σ=0.86	6 *of* 12
(C) DataScout	μ=4.75; σ=0.45	μ=3.67; σ=0.78	10 *of* 12

were exposed to the conditions.

5.4 User Study Findings

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)**. 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 Kaggle's interface to be restrictive (P2–P4, P6, P10). On a 5-point Likert scale, participants expressed neutral-to-mild liking for the baseline interfaces and their search result relevance. Participants had varied success across conditions (Table 5.2).

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.

We observed key differences in dataset search workflows across conditions. First, participants wrote longer and more expressive queries with both DataScout and the semantic baseline compared to Kaggle. For example, P2 searched for *"images that are artworks with the names of the artists"* on DataScout, versus a shorter *"art history"* on Kaggle. However, participants like P6 and P9 noted a higher start-up cost involved in writing elaborate queries. 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.

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).

5.4.1 Finding 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.

5.4.1.1 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).

5.4.1.2 DataScout empowered users by enabling fine-grained queries over dataset attributes and granularity levels

Participants used DataScout's features (query reformulation suggestions, and semantic attribute and granularity filters) to systematically and deliberately broaden or refine their search (P1, P4, P5, P7, P8, P10, P12). For instance, P7 began with the query: "land use in USA," which returned mostly irrelevant results. They 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).

5.4.2 Finding 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.

5.4.2.1 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.

5.4.2.2 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.[1] 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."*

5.4.2.3 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 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.

[1] 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 not only 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 [53].

Chapter 6

Performance Analysis and Future Work

To evaluate the scalability and efficiency of DataScout, we developed two benchmarks aimed at measuring system performance under varying datasize and retrieval conditions:

1. **Dataset Retrieval Benchmark:** Retrieve a fixed number of datasets from databases (i.e., a dataset collection) of increasing sizes to measure how search latency scales with respect to the total number of datasets in the system.
2. **Semantic Filtering Benchmark:** Vary the number of retrieved search results from a fixed-size database to evaluate how the latency of semantic components—such as query reformulation suggestions, column filter suggestions, and HNSW column concept extraction—scales with result size.

6.1 Dataset Retrieval Benchmark

First, we investigated how retrieval latency is affected by database size to evaluate the scalability of the system and determine whether it can maintain efficient performance over larger corpora. To set up this benchmark, we created several databases of varying sizes—100, 200, 500, 1000, 2000, 4000, and 6500 entries—by sampling datasets from the full corpus. For each database size, we retrieved a fixed number of 100 datasets and recorded the retrieval time in seconds. Retrieval time is defined as the time taken to generate hypothetical schemas given the query, perform a cosine similarity search, and render the resulting datasets in DataScout's interface, as described in Section 4.2. We also define a *Subsequent Processing* category, which comprises the processing time of downstream semantic components and additional system-level computations involved in dynamically presenting results.

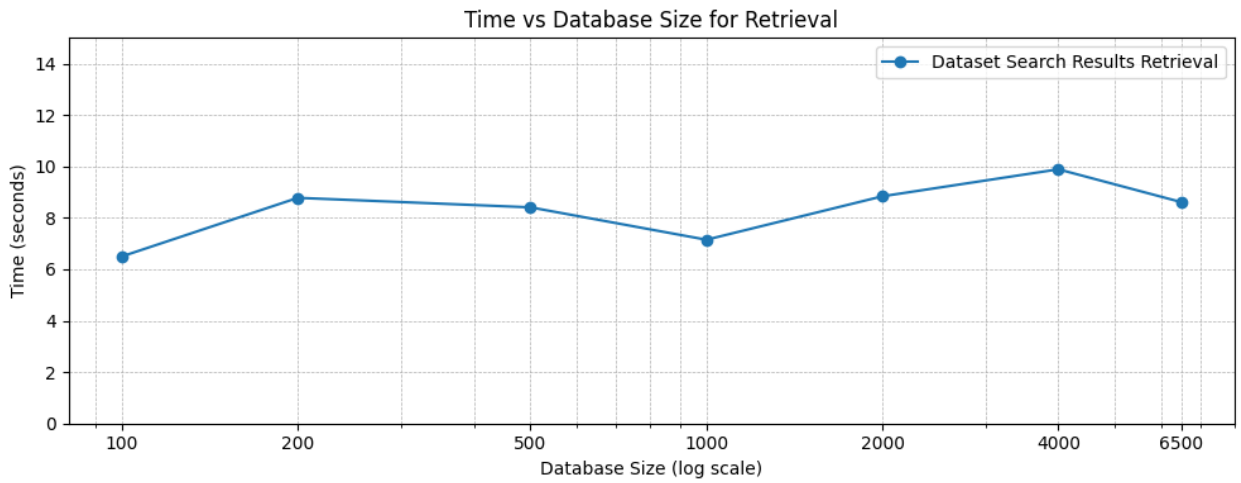


Figure 6.1: Retrieval time (in seconds) for a fixed query size of 100 datasets across databases of increasing size.

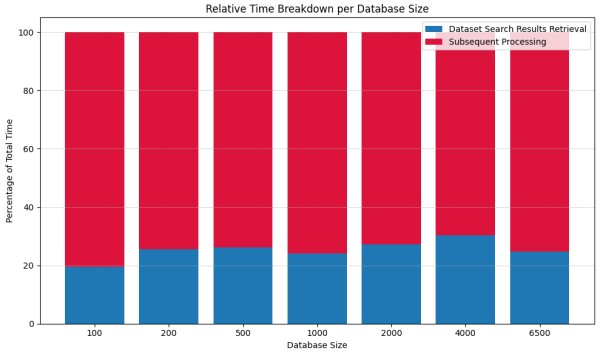


Figure 6.2: Relative breakdown of component latency across varying database sizes.

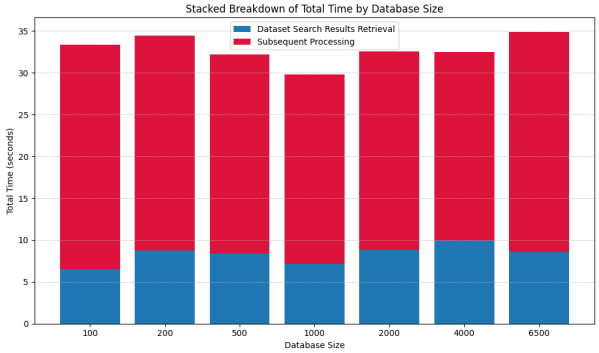


Figure 6.3: Total retrieval time of components across varying database sizes.

Figure 6.1 illustrates the retrieval time across database sizes, with the x-axis plotted on a logarithmic scale. Despite a 65× increase in database size from 100 to 6500 entries, retrieval time remains relatively stable, ranging from approximately 6.5 to 9.9 seconds, with an average of 8.3 seconds across all tested database sizes. This indicates that DataScout's retrieval process scales efficiently with database size, likely due to the HNSW dataset index on the dataset embedding. Minor fluctuations in timing are likely attributable to system-level factors such as runtime load, rather than fundamental limitations of the retrieval mechanism.

To contextualize retrieval within the overall search pipeline, we break down total latency by functional components, as shown in Figure 6.2 and Figure 6.3. As seen in Figure 6.2, retrieval consistently accounts for approximately 20-28% of total processing time across all database sizes. Figure 6.2 reveals that the total processing time remains relatively stable

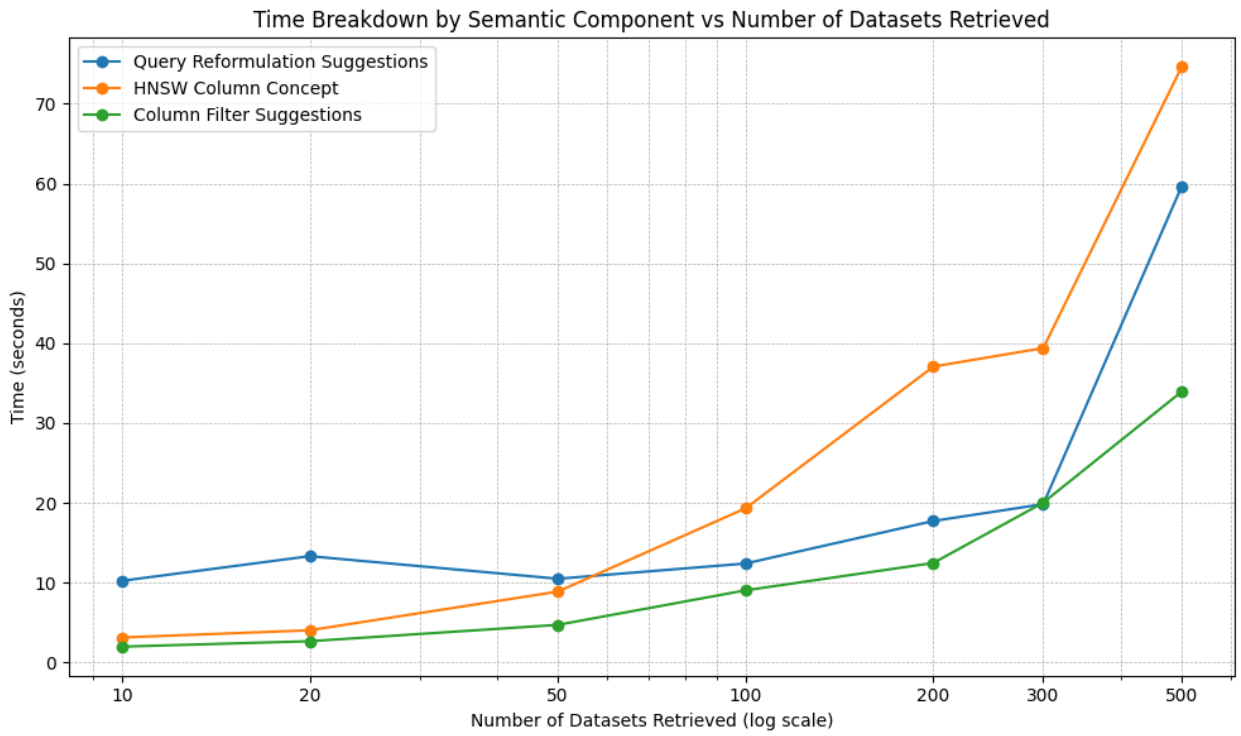


Figure 6.4: Processing time (in seconds) for semantic components as number of datasets retrieved varies, with database size held constant at 6500 entries.

between 30-35 seconds regardless of database size, with semantic components accounting for the majority of execution time. These results suggest that retrieval is not the primary performance bottleneck in DataScout. Instead, semantic components and downstream processing account for most of the runtime. Therefore, future optimization efforts should focus on improving the efficiency of semantic operations to meaningfully reduce overall computation time. Moreover, the stability of retrieval times across increasing database sizes suggests that the HNSW indexing approach on dataset embeddings will likely remain effective for DataScout's task-driven search, even as the system scales beyond the tested range.

6.2 Semantic Filtering Benchmark

We next conducted a deeper analysis of the latency associated with semantic components—specifically, query reformulation suggestions, column filter suggestions, and HNSW column concept extraction. Our goal was to analyze how latency scales as we increase the number of retrieved datasets—10, 20, 50, 100, 200, 300, and 500—while keeping the database size fixed at 6500 entries. Figure 6.4 depicts how the processing time for each semantic component

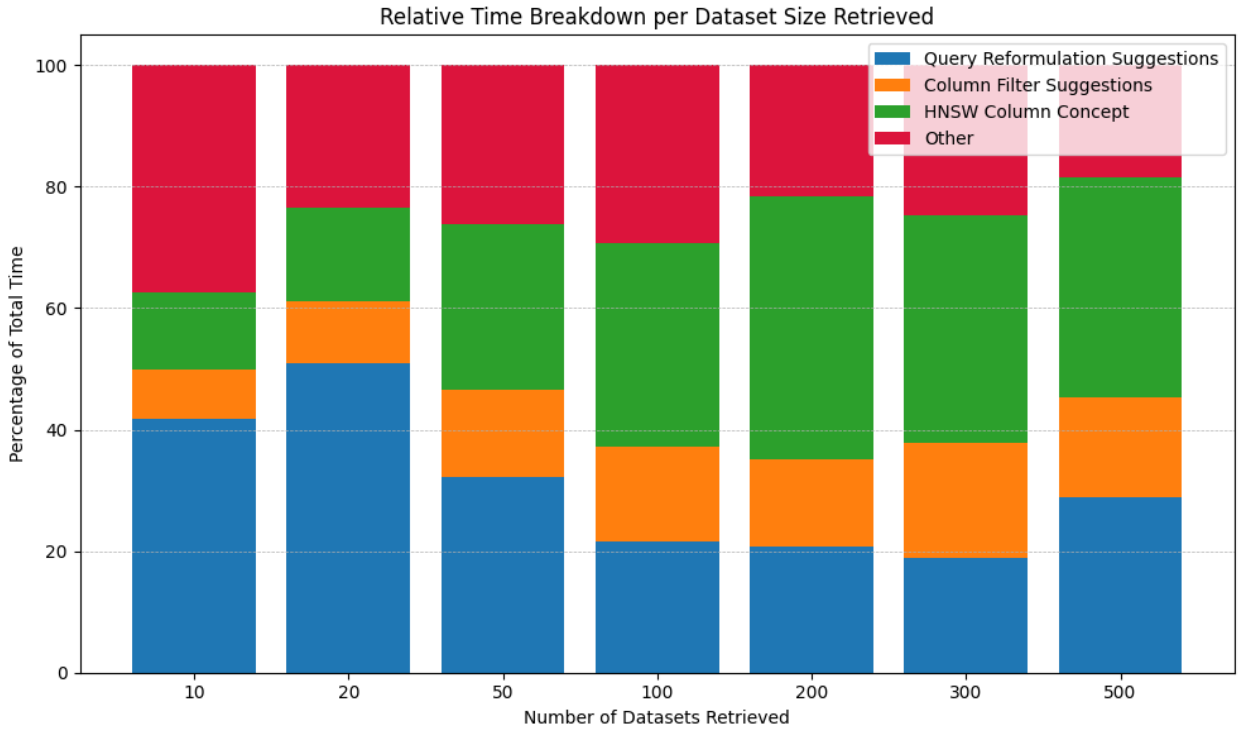


Figure 6.5: Relative breakdown of latency across semantic components with varying numbers of retrieved datasets.

changes with the number of retrieved datasets, with the x-axis plotted on a logarithmic scale.

The results reveal distinct scaling behaviors across the semantic components. The HNSW column concept component exhibits the most dramatic increase in latency, following an almost quadratic growth pattern, as shown in Figure 6.4. This suggests that the underlying semantic vector search becomes increasingly expensive as the number of retrieved results grows, making it the most significant bottleneck in DataScout's search pipeline. Optimizing this component is therefore critical to balancing system responsiveness with high-quality semantic results for users.

In contrast, query reformulation suggestions and column suggested attribute filters show more moderate scaling behavior as shown in Figure 6.5. Here, *Other* refers to the retrieval time of datasets, along with additional system-level computations involved in dynamically presenting the results. As described in Sections 4.3.2 and 4.3.1, both components use KNN clustering to group similar items, followed by an LLM to generate representative names for each group. Their latency generally increases proportionally to the number of retrieved datasets. However, at 500 retrieved datasets, the query reformulation component shows a sharp spike in latency. This spike is likely due to processing a larger number of dataset titles, which results in significantly more tokens compared to processing column names. Figure 6.6

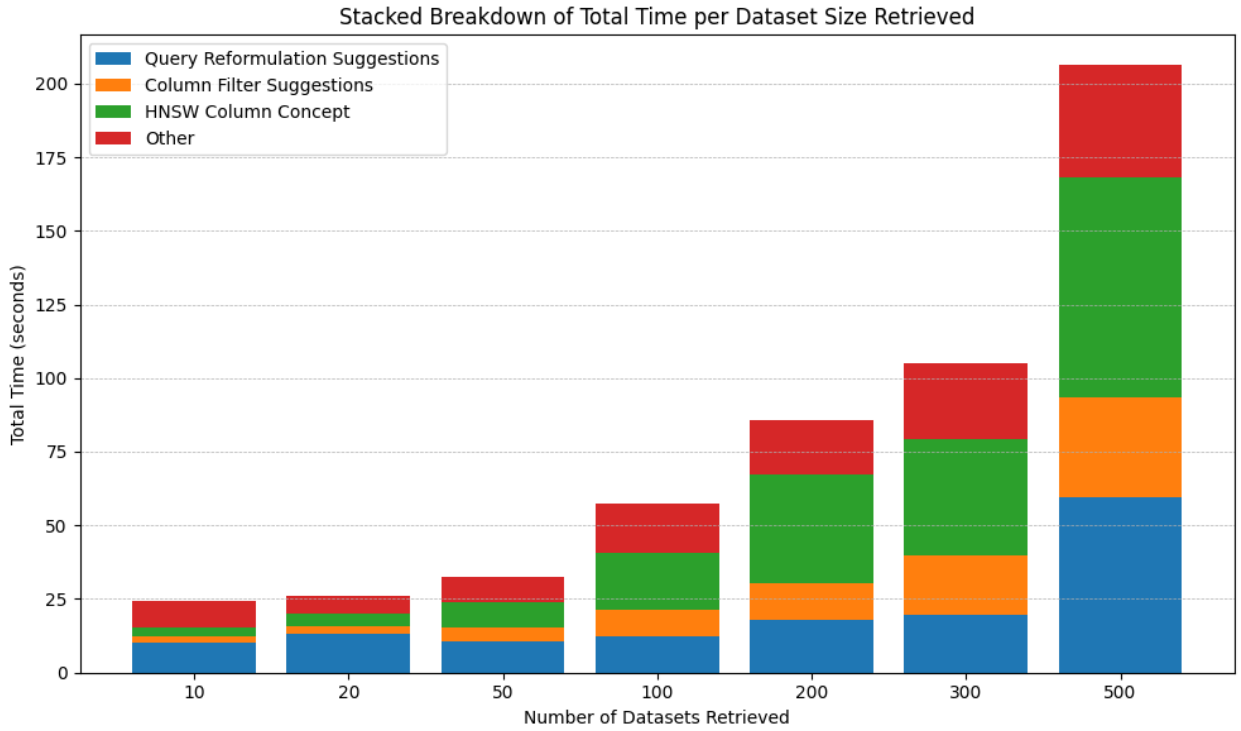


Figure 6.6: Total runtime of semantic components with varying numbers of retrieved datasets.

further provides a complementary view of this scaling behavior, where the sudden increase in computation time for query reformulation suggestions is particularly evident.

6.3 Future Work

The performance analysis across the two benchmarks highlights several opportunities for future optimization and system improvement. While dataset retrieval performance is relatively stable as database size increases, processing time for semantic components becomes a limiting factor in DataScout's performance as the number of datasets retrieved increases. To address the latency introduced by HNSW-based column concept extraction, we plan to explore several optimizations, such as tuning HNSW parameters and parallelizing the semantic search process. These efforts aim to maintain semantic quality while significantly reducing latency for larger result sets.

To reduce the growing runtime of query reformulation suggestions—as clustering and LLM summarization over 500+ dataset titles becomes expensive—we can reduce our reliance on LLMs through lightweight precomputation strategies. For example, we can average the embeddings within each cluster and identify the dataset titles closest to the cluster centroid as representative samples. Then, instead of sending the entire set of titles to the LLM,

summarization can be performed on just the top-k most representative titles per cluster. This approach preserves the diversity of dataset titles while significantly lowering computational costs and improving responsiveness.

Furthermore, based on our user study presented in Section 5.4, we propose several directions for extending DataScout and improving the design of dataset search interfaces. First, users desired a *"birds-eye view"* (P7) summarizing patterns across all search results—such as covered time periods or geographic regions—to expedite sensemaking and offer feedback on their queries (Section 5.4.2). Aggregated overviews, as explored by Ouellette et al. [36], could support this need by presenting bottom-up hierarchical summaries of result sets. Second, users often wanted to combine data from multiple sources to construct their intended dataset (F2, F5, F7, F8)—either 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 navigating multi-dataset compositions. Finally, participants wanted visibility into the quality of key dataset attributes (P4, P10, P12). Building on existing efforts in data quality detection and wrangling [19, 14, 8, 46], future systems could surface these cues as part of relevance indicators to better inform user decisions.

Chapter 7

Limitations and Conclusion

7.1 Limitations

Our evaluation of DataScout has several limitations. First, the precision of search results in our system was constrained by the availability and curation quality of underlying data sources. We relied on a limited dataset collection sourced from Kaggle. This occasionally led to irrelevant or mismatched results, even after augmenting our dataset corpus with ∼300 datasets for participant tasks—many of which were still not surfaced in response to relevant queries. Second, our prototype lacked basic search functionalities such as result sorting and support for varied relevance criteria (e.g. upload date, downloads, and size), which may have limited participants' ability to explore results systematically. Third, we recorded only two observations per ordering of conditions, which may have limited our findings on experiential effects of DataScout. Finally, we compared our system only against Kaggle's search interface as a keyword-search baseline, owing to our shared reliance on Kaggle datasets and the lack of access to other deployed dataset search systems with similar data. This choice allowed for more direct comparisons, but narrowed the scope of our evaluation.

7.2 Conclusion

In this thesis, we introduced 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 from users. 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.

Bibliography

[1] Marcia J Bates. "The design of browsing and berrypicking techniques for the online search interface". In: *Online review* 13.5 (1989), pp. 407–424.

[2] Alex Bogatu et al. "Voyager: Data discovery and integration for data science". In: *Proceedings 25th International Conference on Extending Database Technology (EDBT 2022)*. 2022.

[3] Virginia Braun and Victoria Clarke. "Reflecting on reflexive thematic analysis". In: *Qualitative research in sport, exercise and health* 11.4 (2019), pp. 589–597.

[4] Virginia Braun and Victoria Clarke. "Using thematic analysis in psychology". In: *Qualitative research in psychology* 3.2 (2006), pp. 77–101.

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

[6] Sonia Castelo et al. "Auctus: A dataset search engine for data augmentation". In: *arXiv preprint arXiv:2102.05716* (2021).

[7] Adriane Chapman et al. "Dataset search: a survey". In: *The VLDB Journal* 29.1 (2020), pp. 251–272.

[8] Bhavya Chopra et al. "Cowrangler: Recommender system for data-wrangling scripts". In: *Companion of the 2023 International Conference on Management of Data*. 2023, pp. 147–150.

[9] Mahdi Esmailoghli et al. "Blend: A unified data discovery system". In: *arXiv preprint arXiv:2310.02656* (2023).

[10] Grace Fan et al. "Semantics-aware dataset discovery from data lakes with contextualized column-based representation learning". In: *arXiv preprint arXiv:2210.01922* (2022).

[11] Grace Fan et al. "Table discovery in data lakes: State-of-the-art and future directions". In: *Companion of the 2023 International Conference on Management of Data*. 2023, pp. 69–75.

[12] Raul Castro Fernandez et al. "Aurum: A data discovery system". In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE. 2018, pp. 1001–1012.

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

[14] Philip J Guo et al. "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*. 2011, pp. 65–74.

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

[16] Jonathan Herzig et al. "Open domain question answering over tables via dense retrieval". In: *arXiv preprint arXiv:2103.12011* (2021).

[17] Zezhou Huang et al. "The Fast and the Private: Task-based Dataset Search". In: *arXiv preprint arXiv:2308.05637* (2023).

[18] Madelon Hulsebos et al. "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*. 2024, pp. 1–4.

[19] Sean Kandel et al. "Wrangler: Interactive visual specification of data transformation scripts". In: *Proceedings of the sigchi conference on human factors in computing systems*. 2011, pp. 3363–3372.

[20] Moe Kayali et al. "Mind the Data Gap: Bridging LLMs to Enterprise Data Integration". In: *arXiv preprint arXiv:2412.20331* (2024).

[21] Aamod Khatiwada et al. "Santos: Relationship-based semantic table union search". In: *Proceedings of the ACM on Management of Data* 1.1 (2023), pp. 1–25.

[22] Laura Koesten et al. "Everything you always wanted to know about a dataset: Studies in data summarisation". In: *International journal of human-computer studies* 135 (2020), p. 102367.

[23] Laura Koesten et al. "Talking datasets–understanding data sensemaking behaviours". In: *International journal of human-computer studies* 146 (2021), p. 102562.

[24] Laura M Koesten et al. "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*. 2017, pp. 1277–1289.

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

[26] Andrew Kuznetsov et al. "Fuse: In-situ sensemaking support in the browser". In: *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 2022, pp. 1–15.

[27] Bongshin Lee et al. "FacetLens: exposing trends and relationships to support sensemaking within faceted datasets". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2009, pp. 1293–1302.

[28] Bongshin Lee et al. "Understanding research trends in conferences using PaperLens". In: *CHI'05 extended abstracts on Human factors in computing systems*. 2005, pp. 1969–1972.

[29] Aristotelis Leventidis et al. "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*. 2024, pp. 1142–1151.

[30] Michael Xieyang Liu et al. "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*. 2024, pp. 1–26.

[31] Yu A Malkov and Dmitry A Yashunin. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs". In: *IEEE transactions on pattern analysis and machine intelligence* 42.4 (2018), pp. 824–836.

[32] Gary Marchionini. "Exploratory search: from finding to understanding". In: *Communications of the ACM* 49.4 (2006), pp. 41–46.

[33] Fengran Mo et al. "A survey of conversational search". In: *arXiv preprint arXiv:2410.15576* (2024).

[34] Meredith Ringel Morris, Jarrod Lombardo, and Daniel Wigdor. "WeSearch: supporting collaborative search and sensemaking on a tabletop display". In: *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. 2010, pp. 401–410.

[35] Donald A. Norman. *The Design of Everyday Things*. USA: Basic Books, Inc., 2002. ISBN: 9780465067107.

[36] Paul Ouellette et al. "RONIN: data lake exploration". In: *Proceedings of the VLDB Endowment* 14.12 (2021).

[37] Srishti Palani et al. "CoNotate: Suggesting queries based on notes promotes knowledge discovery". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–14.

[38] Srishti Palani et al. "The" Active Search" Hypothesis: How search strategies relate to creative learning". In: *Proceedings of the 2021 conference on human information interaction and retrieval*. 2021, pp. 325–329.

[39] Peter Pirolli and Stuart Card. "Information foraging." In: *Psychological review* 106.4 (1999), p. 643.

[40] Peter L. T. Pirolli. *Information Foraging Theory: Adaptive Interaction with Information*. 1st ed. USA: Oxford University Press, Inc., 2007. ISBN: 0195173325.

[41] Filip Radlinski and Nick Craswell. "A theoretical framework for conversational search". In: *Proceedings of the 2017 conference on conference human information interaction and retrieval*. 2017, pp. 117–126.

[42] Corbin Rosset et al. "Leading conversational search by suggesting useful questions". In: *Proceedings of the web conference 2020*. 2020, pp. 1160–1170.

[43] Tony Russell-Rose and Tyler Tate. "Chapter 2 - Information Seeking". In: *Designing the Search Experience*. Ed. by Tony Russell-Rose and Tyler Tate. Morgan Kaufmann, 2013, pp. 23–45. ISBN: 978-0-12-396981-1. DOI: https://doi.org/10.1016/B978-0-12-396981-1.00002-1. URL: https://www.sciencedirect.com/science/article/pii/B9780123969811000021.

[44] Vidya Setlur, Andriy Kanyuka, and Arjun Srinivasan. "Olio: A semantic search interface for data repositories". In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 2023, pp. 1–16.

[45] Ben Shneiderman. *Designing The user interface: Strategies for effective human-computer interaction, 4/e (New Edition)*. Pearson Education India, 1987.

[46] skrub-data. *skrub: A library for data cleaning and preprocessing*. https://github.com/skrub-data/skrub. Accessed: 2025-04-07. 2025.

[47] Greg Smith et al. "FacetMap: A scalable search and browse visualization". In: *IEEE Transactions on visualization and computer graphics* 12.5 (2006), pp. 797–804.

[48] Katrina Sostek et al. "Discovering datasets on the web scale: Challenges and recommendations for Google Dataset Search". In: *Harvard Data Science Review* Special Issue 4 (2024).

[49] Statista. *Data growth worldwide 2010-2028*. Accessed: 2025-04-01. 2025. URL: https://www.statista.com/statistics/871513/worldwide-data-created/.

[50] Sangho Suh et al. "Sensecape: Enabling multilevel exploration and sensemaking with large language models". In: *Proceedings of the 36th annual ACM symposium on user interface software and technology*. 2023, pp. 1–18.

[51] Nitya Tarakad. "A Peek Inside: How Snowflake's New Universal Search Feature Was Built". In: *Snowflake Builders Blog: Data Engineers, App Developers, AI/ML, & Data Science* (Feb. 2024). URL: https://medium.com/snowflake/a-peek-inside-how-snowflakes-new-universal-search-feature-was-built-dfd1188176d0.

[52] Daniel Tunkelang. *Faceted search*. Springer Nature, 2022.

[53] Iris Vessey. "Cognitive fit: A theory-based analysis of the graphs versus tables literature". In: *Decision sciences* 22.2 (1991), pp. 219–240.

[54] Alexandra Vtyurina et al. "Exploring conversational search with humans, assistants, and wizards". In: *Proceedings of the 2017 chi conference extended abstracts on human factors in computing systems*. 2017, pp. 2187–2193.

[55] Chi Zhang. "Adding Intelligence to Databricks Search". In: *Databricks Blog* (Mar. 2024). URL: https://www.databricks.com/blog/adding-intelligence-to-databricks-search.

[56] Yongfeng Zhang et al. "Towards conversational search and recommendation: System ask, user respond". In: *Proceedings of the 27th acm international conference on information and knowledge management*. 2018, pp. 177–186.

[57] Yihang Zhao, Albert Meroño-Peñuela, and Elena Simperl. "User Experience in Dataset Search Platform Interfaces". In: *arXiv e-prints* (2024), arXiv–2403.