

Intent-aware Join Discovery using Natural Language

Mir Mahathir Mohammad
University of Utah
Salt Lake City, UT, USA
mahathir.mohammad@utah.edu

El Kindi Rezig
University of Utah
Salt Lake City, UT, USA
elkindi.rezig@utah.edu

Abstract

We present NLDisc, a system for intent-aware join discovery over data lakes using natural language (NL) queries. Unlike existing data discovery systems that rely on structured inputs (e.g., example tables) or fail to support join operations for NL queries, NLDisc enables users to describe their desired datasets in natural language—automatically discovering relevant join paths across multiple tables, NLDisc addresses key challenges in NL-based data discovery by supporting interactive intent disambiguation. We demonstrate NLDisc’s capabilities across diverse real-world data lakes, including benchmarks and public datasets. A companion video is available at [2].

CCS Concepts

• **Information systems** → **Information integration; Mediators and data integration; Query representation.**

Keywords

Data Discovery, Join Discovery, Semantic Join, Natural Language Query, Join Path, Data Lake, Semantic Type

ACM Reference Format:

Mir Mahathir Mohammad and El Kindi Rezig. 2026. Intent-aware Join Discovery using Natural Language. In *Companion of the International Conference on Management of Data (SIGMOD Companion '26)*, May 31–June 05, 2026, Bengaluru, India. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3788853.3801605>

1 Introduction

Data lakes are now pervasive, from public repositories (e.g., US Government Open Data [1]) to private platforms (e.g., Academic Institute Data Warehouse [3]). Yet, building datasets from them remains a major obstacle. Unlike traditional databases, data lakes lack explicit schemas (e.g., primary/foreign keys), forcing users to manually interpret table contents and relationships. Despite advances in data discovery tools [8, 10, 11], current systems rely on query tables or structured queries, lacking natural language (NL) interfaces for discovering *joined datasets* and failing to model ambiguity in user intent.

Example: Consider Figure 1(1), where a data scientist, Lou, writes a natural language query describing the dataset they want to build. Ideally, Lou expects the following requirements:

(R1) Query by natural language: Because Lou is unfamiliar with the data lake’s values, they use a natural language interface

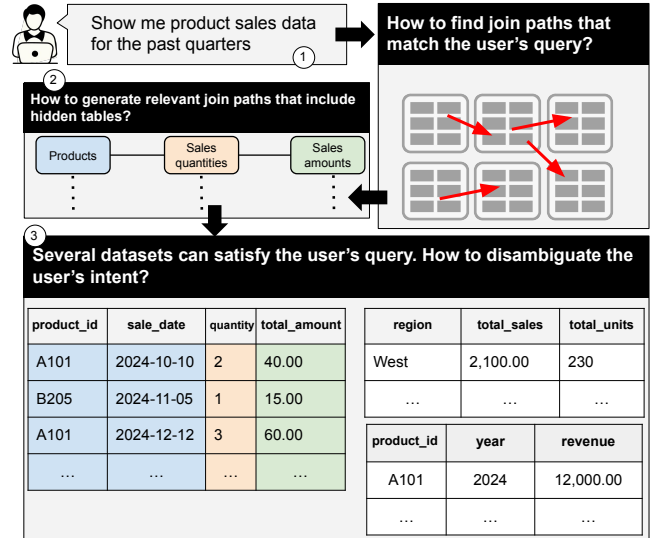


Figure 1: In this motivating example, users compose a natural language query (1) and get multiple join paths that match the semantics of the query (2) and (3).

describing only the target dataset they want to build. Existing NL-enabled discovery systems [15] are limited to finding individual tables and do not perform join discovery.

(R2) Intent disambiguation: Because NL queries are inherently ambiguous, the system should allow Lou to clarify ambiguities. For instance, in Figure 1(3), all three tables could be valid interpretations of the query.

(R3) Diversification of Join Paths: Existing approaches [10] focus exclusively on content overlap with the query table, often producing semantically redundant results with duplicate content. A system must account for semantic similarity among join paths without incurring the cost of materialization, returning a diverse set relevant to user needs.

Contributions. We present NLDisc, which extends SEMDisc [10] with NL query support, enabling interactive join path exploration. Specifically, NLDisc: (1) automatically discovers relevant join paths from NL queries; (2) unlike NL2SQL approaches, translates NL queries into a structured, interpretable representation (a query table) that users can refine to better express their intent; and (3) diversifies join paths based on semantic similarity.

Related work. Various data discovery techniques have explored different interfaces for automatic data lake discovery. DICE [11] discovers equi-joins using an example table; Aurum [8] uses a



This work is licensed under a Creative Commons Attribution 4.0 International License. *SIGMOD Companion '26, Bengaluru, India*
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2450-3/2026/05
<https://doi.org/10.1145/3788853.3801605>

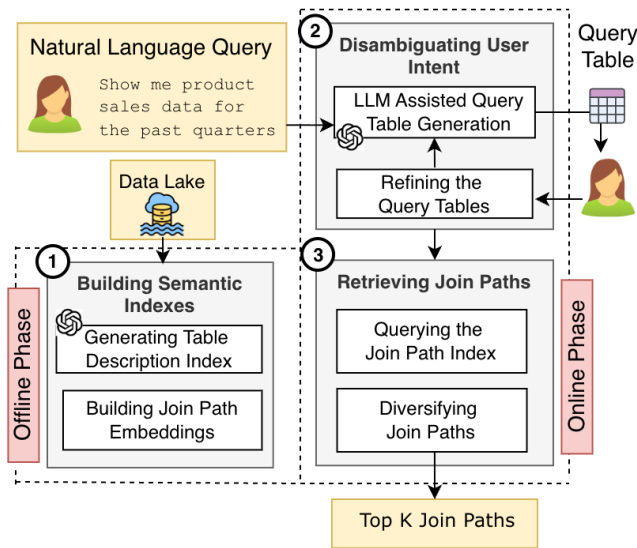


Figure 2: System overview of NLDisc, combining offline semantic indexing with online intent disambiguation and diversified join path retrieval.

structured query language for equi-joins across columns; and DeepJoin [6], WarpGate [4], and SemDisc [10] discover semantic join paths given a query column. However, none support NL queries. A related line of work is NL2SQL [7], which generates SQL for a given NL query but (1) is not designed for schema-less data lakes, and (2) does not support intent disambiguation. Finally, Wang et al. [15] proposed an NL discovery system, but unlike NLDisc, it (1) only queries single tables (no joins), and (2) does not allow intent disambiguation.

2 System Overview

Figure 2 illustrates the workflow of NLDisc. NLDisc operates in two phases: (1) *Offline Phase*: NLDisc extracts descriptions from data lake tables and constructs a table description index to retrieve candidate tables via semantic similarity with the user’s natural language query (Figure 2 ①). Additionally, NLDisc builds join path embeddings based on their semantic types. (2) *Online Phase*: When a user submits a natural language query, NLDisc first suggests candidate query tables for the user to assess; the user may select, modify, and resubmit until a satisfactory query table is obtained (Figure 2 ②). Once the user’s intent has been disambiguated through the final query table, the system retrieves the top-k join paths from a join path index and diversifies the results based on join path embeddings (Figure 2 ③).

2.1 Building Semantic Indexes

This section describes the offline phase of NLDisc, which extracts semantic table descriptions and constructs a join path index for efficient retrieval.

Generating Table Description Index: To identify candidate tables semantically similar to a natural language query, NLDisc uses an

LLM (GPT-5.1 [12]) to generate a single-sentence description per table. The prompt is structured as follows:

Instruction: A directive to generate a natural language table description.

You are given a table name, column semantic types, and example values.

Table name: regional_sales

Table Metadata: <Table Metadata>

Write a single, concise, natural-language sentence that explains what the table contains. The description should be human-readable. Keep each description to one sentence and avoid bullet points or numbering.

Output Format: An NL description of the table.

The table contains the regional sales of units.

Input Table Metadata: Columns listed one per line with identifier, semantic type, and sample values.

column 1 Semantic Type: Region Names

Example Values: West Region, East Region, ...

column 2 ...

NLDisc then computes MPNet [13] embeddings for all descriptions and indexes them using an HNSW [9] index for fast approximate nearest neighbor search. NLDisc also leverages SEMDisc’s [10] join detection to identify equi-joins and semantic joins by extracting semantic types and computing Jaccard similarity between column sketches. The detected relationships form a join graph (nodes = tables, edges = joinable column pairs), from which NLDisc enumerates all simple join paths and encodes them as a binary matrix: rows are tables, columns are paths, and 1 in a cell indicates that the corresponding table is present in the respective path. See SEMDisc [10] for details.

Building Join Path Embeddings: During the online phase, NLDisc diversifies returned join paths by filtering out those with overlapping tables and ranking the rest so diverse paths appear first in the top-K. Unlike [10], NLDisc constructs a semantic representation per path: it extracts MPNet embeddings for the semantic types of all columns in the path and computes their mean vector. These embeddings enable diversity-aware ranking and avoid redundant results.

2.2 Disambiguating User Intent

This section describes the online phase of NLDisc, which accepts a natural language query and iteratively disambiguates user intent. NLDisc suggests sample table representations called *Query Tables* and incorporates user feedback to refine them, leveraging data lake tables for increasingly personalized results. Given a query, NLDisc extracts its MPNet embedding and retrieves the most relevant tables from the HNSW index of table description embeddings. These serve as initial candidate tables, provided as context for query table generation.

LLM-Assisted Query Table Generation: NLDisc prompts an LLM (GPT-5.1 [12]) to generate sample query tables that disambiguate user intent. The prompt is structured as follows:

Instruction: A directive to generate a user-specified number of query tables.

You are given a list of context tables, their column semantic types, and sample values. The user has provided a natural language description of a table they want the system to

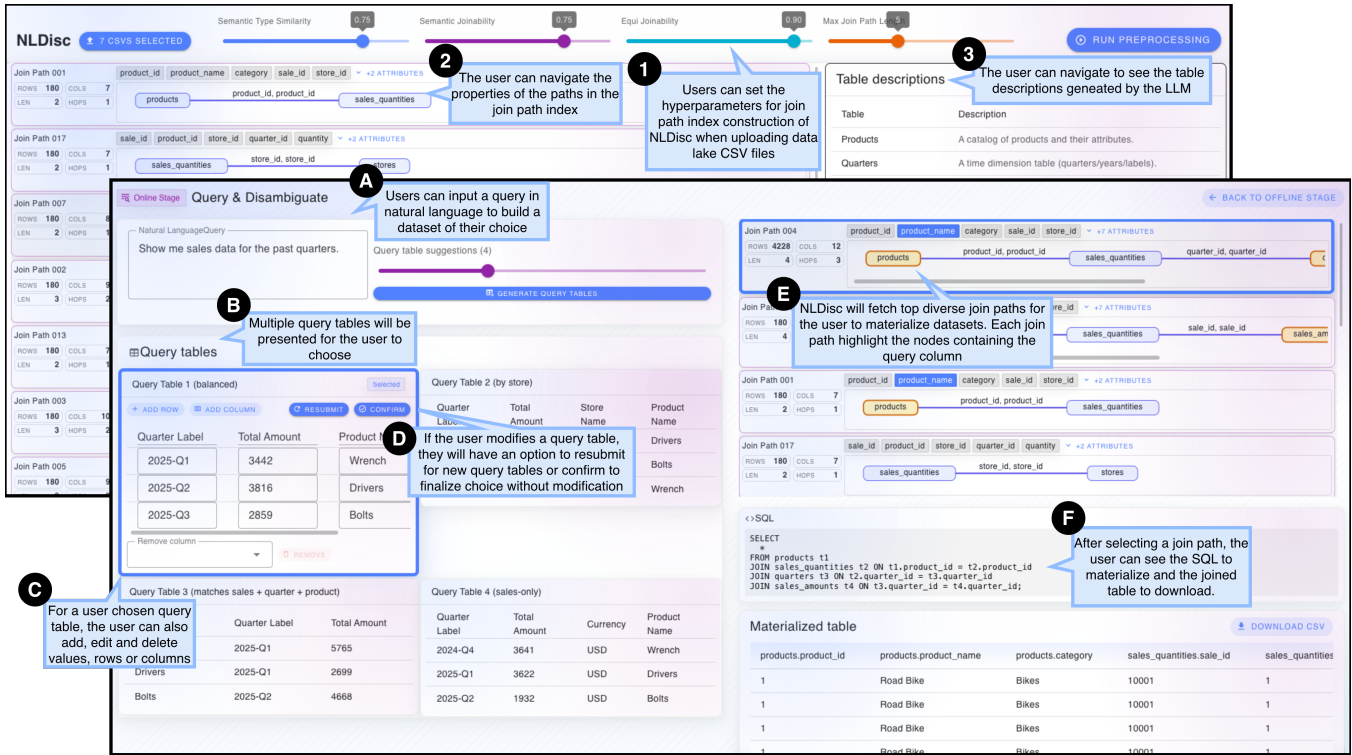


Figure 3: The NLDisc web interface showing the offline phase components (back screenshot: data lake upload, join path index, table descriptions) and online phase workflow (front screenshot: natural language query input, intent disambiguation via query tables, and diverse join path retrieval).

produce from the context tables (by joining multiple tables if necessary).
 Suggest <User-specified number> sample tables with 3 rows that could be produced from the context tables, aligned with the user’s natural language query.
 User Query: <User Query>
 <Context Tables>
 <Output Format>

Context Tables: Provided with their column semantic types and example values.

Table 1: <Table Name>
 Column 1 Semantic Type: <Semantic Type>
 Column 1 Example Values: <List of 10 most frequent values>
 ...

Output Format: The LLM returns query tables as a JSON array:
 Only reply with a response that follows the JSON format specified below:

```
[{"table": "Query 1",
  "content": [
    {"semantic type": <column semantic type>,
     "examples": [<example 1>, <example 2>, ...]}
  ], ...}]...
```

Refining Query Tables: After generation, NLDisc presents query tables to the user, who can add, edit, or remove rows and columns before resubmitting. The system then identifies data lake tables with overlapping semantic types and values, using these as context for subsequent LLM-assisted generation. This iterative process

continues until the user finalizes a query table, simultaneously curating the desired output and refining the candidate tables needed for joining.

2.3 Retrieving Join Paths

Querying the Join Path Index: During the online phase, NLDisc uses the join path index to retrieve paths containing the candidate tables. Once the user finalizes a query table, NLDisc queries the index by identifying columns where all corresponding candidate table rows contain 1s, and retrieves all join paths that involve all candidate tables.

Diversifying Join Paths: NLDisc leverages join path embeddings to diversify the results returned from the index, selecting a final set of K diverse join paths from the initial join path candidates. This diversification follows a two-step process:

(1) Semantic Deduplication: NLDisc initializes an empty list of deduplicated join paths and iterates through all the paths containing the candidate tables. For each path, NLDisc computes the cosine similarity between its embedding and those of all paths already in the deduplicated list. If the cosine similarity exceeds a threshold (~ 0.8) with any existing path, the current path is discarded; otherwise, it is added to the deduplicated list.

(2) Diversification: NLDisc applies K -means clustering with K centroids to the embeddings of the deduplicated join paths. For each cluster, only the join path whose embedding is closest to the

centroid is selected, yielding the final set of K diverse join paths returned to the user.

3 Demonstration plan

We demonstrate NLDisc on the following data lakes: (1) Spider (SP) [16], a Text-to-SQL benchmark, adapted by selecting 156 databases with join queries and removing schema information to test join recovery; (2) LakeBench (LB) [5], a data discovery benchmark whose join subset (OpenData) provides ground truth joinable column pairs; (3) DrugCentral (DC) [14], a curated resource with comprehensive drug-related data; (4) U.S. Fish and Wildlife Service (FWS), containing geographic, behavioral, and ecological records of U.S. flora and fauna; and (5) Centers for Disease Control and Prevention (CDC), including public health datasets such as disease incidence, mortality, and drug usage statistics from data.gov.

Demonstration outline. SIGMOD participants will discover datasets from the above data lakes by interacting with NLDisc, including (1) asking natural language queries and disambiguating intent through suggested query tables; (2) examining returned join paths and their source tables. We present two scenarios illustrating the inner workings of NLDisc.

Scenario 1: Offline Phase. Participants can adjust offline phase settings and observe how NLDisc constructs the required indexes.

① **Data Lake Upload:** Users upload CSV files through the NLDisc web interface (Figure 3 ①), triggering the *Offline Phase*, which constructs the join path index, table description index, and join path embeddings. This step is performed once; participants can configure hyperparameters to customize similarity thresholds for identifying joinable columns.

② **Exploring Curated Join Paths:** Participants can examine all indexed join paths (Figure 3 ②), each displaying its estimated cardinality, attribute list, hop count, and a graph visualization of tables and joinable column pairs.

③ **Table Descriptions:** Users can view LLM-generated table descriptions (Figure 3 ③), each a single natural-language sentence with an MPNet embedding indexed via HNSW.

Scenario 2: Online Phase. Participants load a preprocessed data lake and initiate the online phase with a natural language prompt, constructing a dataset through intent disambiguation.

Ⓐ **Query Specification:** Participants pose natural language queries (Figure 3 Ⓐ), describing the desired table to construct from available data lake tables.

Ⓑ **Intent Disambiguation:** NLDisc presents candidate query tables (Figure 3 Ⓑ) as previews of the final constructed dataset.

Ⓒ **Editing Query Tables:** Users refine candidates to clarify intent—e.g., adding, editing, or deleting values, rows, or columns (Figure 3 Ⓒ). These edits directly influence the tuples and attributes in the materialized table.

Ⓓ **Resubmit or Confirm Query Table:** Users can click 'Resubmit' to generate new candidates (Figure 3 Ⓓ), iteratively refining suggestions, or *confirm* the selected query table to proceed with join path retrieval.

Ⓔ **Retrieved Diverse Join Paths:** Upon confirmation, NLDisc presents ranked diverse join paths (Figure 3 Ⓔ) with the same

metadata as indexed paths, highlighting nodes containing query table columns in yellow.

Ⓕ **Materializing a Join Path:** Users select a join path to view its SQL query (Figure 3 Ⓕ) and download the materialized table as CSV or SQL.

Demonstration engagement. Since NLDisc is an interactive system requiring human feedback (intent disambiguation), this demonstration will engage SIGMOD participants at various steps of the join discovery process. Participants will be able to observe the unique *ambiguity* challenges that arise in join discovery when using natural language as the query interface.

References

- [1] [n. d.]. The Home of the U.S. Government's Open Data. <https://data.gov/>.
- [2] [n. d.]. NLDisc demo video. <https://youtu.be/QRpM5aWiS8>
- [3] Peter Baile Chen, Fabian Wenz, Yi Zhang, Devin Yang, Justin Choi, Nesime Tatbul, Michael Cafarella, Çağatay Demiralp, and Michael Stonebraker. 2024. BEAVER: an enterprise benchmark for text-to-sql. *arXiv preprint arXiv:2409.02038* (2024).
- [4] Tianji Cong, James Gale, Jason Frantz, H. V. Jagadish, and Çağatay Demiralp. 2023. WarpGate: A Semantic Join Discovery System for Cloud Data Warehouses. In *13th Conference on Innovative Data Systems Research, CIDR 2023, Amsterdam, The Netherlands, January 8-11, 2023*. www.cidrdb.org.
- [5] Yuhao Deng, Chengliang Chai, Lei Cao, Qin Yuan, Siyuan Chen, Yanrui Yu, Zhaoze Sun, Junyi Wang, Jiajun Li, Ziqi Cao, et al. 2024. LakeBench: A Benchmark for Discovering Joinable and Unionable Tables in Data Lakes. *Proceedings of the VLDB Endowment* 17, 8 (2024), 1925–1938.
- [6] Yuyang Dong, Chuan Xiao, Takuma Nozawa, Masafumi Enomoto, and Masafumi Oyamada. 2023. DeepJoin: Joinable Table Discovery with Pre-Trained Language Models. *Proc. VLDB Endow.* 16, 10 (June 2023), 2458–2470. doi:10.14778/3603581.3603587
- [7] Ju Fan, Zihui Gu, Songyue Zhang, Yuxin Zhang, Zui Chen, Lei Cao, Guoliang Li, Samuel Madden, Xiaoyong Du, and Nan Tang. 2024. Combining Small Language Models and Large Language Models for Zero-Shot NL2SQL. *Proc. VLDB Endowment* 17, 11 (2024), 2750–2763. doi:10.14778/3681954.3681960
- [8] Raul Castro Fernandez, Ziawash Abedjan, Famiem Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, ICDE, ICDE, 1001–1012.
- [9] Yu A. Malkov and D. A. Yashunin. 2020. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 4 (April 2020), 824–836. doi:10.1109/TPAMI.2018.2889473
- [10] Mir Mahathir Mohammad and El Kindi Rezig. 2026. Qualitative Join Discovery in Data Lakes using Examples. *Proceedings of the ACM on Management of Data* 4, 1, Article 68 (feb 2026), 28 pages. <https://doi.org/10.1145/3786682>
- [11] El Kindi Rezig, Anshul Bhandari, Anna Fariha, Benjamin Price, Allan Vanterpool, Vijay Gadepally, and Michael Stonebraker. 2021. DICE: data discovery by example. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2819–2822.
- [12] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267* (2025).
- [13] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (Vancouver, BC, Canada) (NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 1414, 11 pages.
- [14] Oleg Ursu, Jayme Holmes, Cristian G Bologna, Jeremy J Yang, Stephen L Mathias, Vasileios Stathias, Dac-Trung Nguyen, Stephan Schürer, and Tudor Oprea. 2019. DrugCentral 2018: an update. *Nucleic acids research* 47, D1 (2019), D963–D970.
- [15] Qiming Wang and Raul Castro Fernandez. [n. d.]. Solo: Data Discovery Using Natural Language Questions Via A Self-Supervised Approach. *SIGMOD* ([n. d.]).
- [16] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 3911–3921. doi:10.18653/v1/D18-1425