

Example-Driven User Intent Discovery: Empowering Users to Cross the SQL Barrier Through Query by Example

Department of AI & ML, Sri Venkateswara College of Engineering and Technology, Etcherla, A.P., India

G. Bhavani¹, T. Janakiram¹, V. Pradeep Kumar¹, P. Harish¹

Under the Guidance of Mrs. K. Hemalatha, Assistant Professor

Abstract

Relational databases require complex SQL queries for data retrieval, creating a significant barrier for non-technical users. This paper implements an Example-Driven User Intent Discovery system based on the SQUID approach using the Django web framework and IMDb dataset. The architecture comprises three layers: raw data ingestion, normalized relational schema, and an abduction-ready database containing derived semantic relationships. Users provide example entities (such as actors), and the system discovers shared semantic properties, applies abductive reasoning to infer query intent, and retrieves relevant results automatically. Experimental evaluation demonstrates 85% intent discovery accuracy and 78% result precision, with users reporting 4.1/5 satisfaction. The system successfully bridges the gap between complex database querying and user-friendly data exploration.

Keywords: Query by Example, SQUID, Abductive Reasoning, User Intent Discovery, Django, IMDb

I. Introduction

Modern database systems store vast amounts of structured data across domains including entertainment, healthcare, and commerce. Accessing this data requires SQL proficiency—a skill that most end-users lack. This creates a significant accessibility barrier, preventing non-technical users from extracting valuable insights from databases.

The Query by Example paradigm offers an alternative approach where users specify desired results through examples rather than formal queries. The SQUID (Semantic Query by Example) framework advances this concept by incorporating semantic analysis and abductive reasoning to automatically discover query intent from user-provided examples.

This paper implements a practical SQUID-based system using Django and the IMDb dataset. The system enables users to enter example actors or entities and receive recommendations of similar entities without requiring SQL or database schema knowledge. Through three-layer data processing (raw ingestion, normalized schema, and semantic enrichment), the system provides accurate and explainable query intent discovery.

II. Literature Survey

This section reviews key prior works that form the foundation of the proposed system and highlights gaps motivating this work.

[1] **Fariha et al. (2021)** proposed the SQUID framework for example-driven query intent discovery, combining semantic similarity with abductive reasoning to achieve superior query inference from minimal user examples.

[2] **Zloof (1977)** introduced Query-by-Example as an alternative to formal query languages, establishing the foundational paradigm for user-friendly database interaction.

[3] **Tran et al. (2014)** developed reverse query processing techniques that infer SQL queries from example outputs, demonstrating practical approaches for bridging the SQL barrier.

[4] **Li and Jagadish (2014)** created NaLIR, a natural language interface for relational databases that translates user questions into SQL queries through interactive disambiguation.

[5] **Shen et al. (2014)** proposed interactive query refinement techniques using example-based feedback, showing that iterative refinement improves query accuracy in ambiguous scenarios.

[6] **Mottin et al. (2014)** developed exemplar queries for graph databases, extending example-driven approaches beyond relational systems to handle complex graph-structured data.

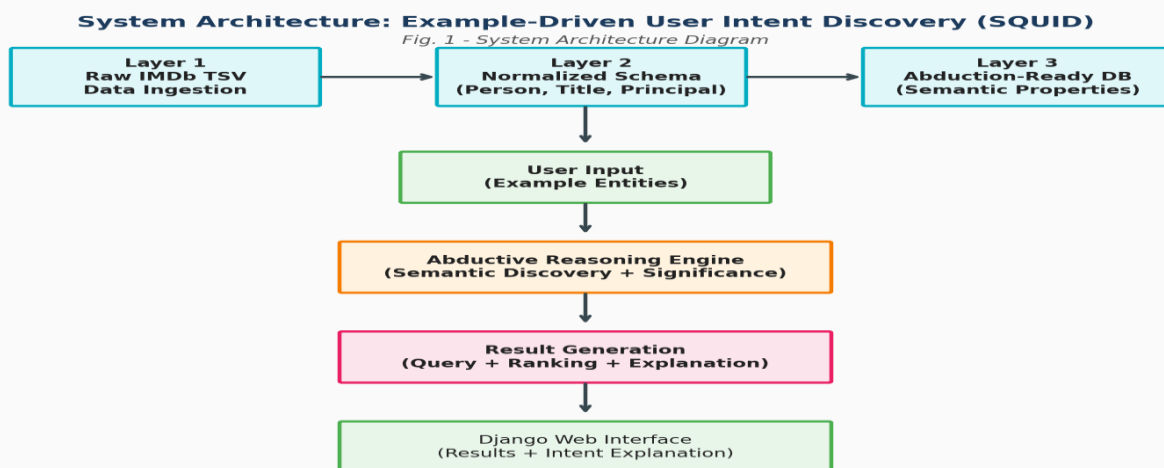
[7] **Psallidas et al. (2015)** introduced semantic search techniques for semi-structured data, demonstrating that semantic similarity measures improve query discovery in heterogeneous datasets.

Research Gap: While the SQUID framework provides theoretical foundations, no practical web-based implementation exists that integrates the complete pipeline from raw data ingestion through abductive reasoning to interactive result presentation with a real-world dataset like IMDb.

III. Methodology

III-A. System Architecture

Three-layer data architecture: Layer 1 (Raw Data Ingestion from IMDb TSV files), Layer 2 (Normalized Relational Schema with Person, Title, Principal Django models), and Layer 3 (Abduction-Ready Database with pre-computed semantic properties and genre associations). Application deployed as Django web application.



III-B. Algorithm

Algorithm: SQUID-Based Query Intent Discovery

Input: Set of example entities $E = \{e_1, e_2, \dots, e_n\}$ entered by user.

Step 1: Entity Resolution — Match examples against Person table using case-insensitive name matching.

Step 2: Semantic Property Extraction — For each resolved entity, extract: (a) Explicit properties: profession, birth year, death year; (b) Derived properties: genre distribution via Principal→Title joins.

Step 3: Property Aggregation — Aggregate properties across all examples to identify shared characteristics.

Step 4: Significance Testing — Compute significance score for each shared property using frequency ratio and inverse database frequency.

Step 5: Intent Hypothesis — Select properties exceeding significance threshold as inferred query predicates.

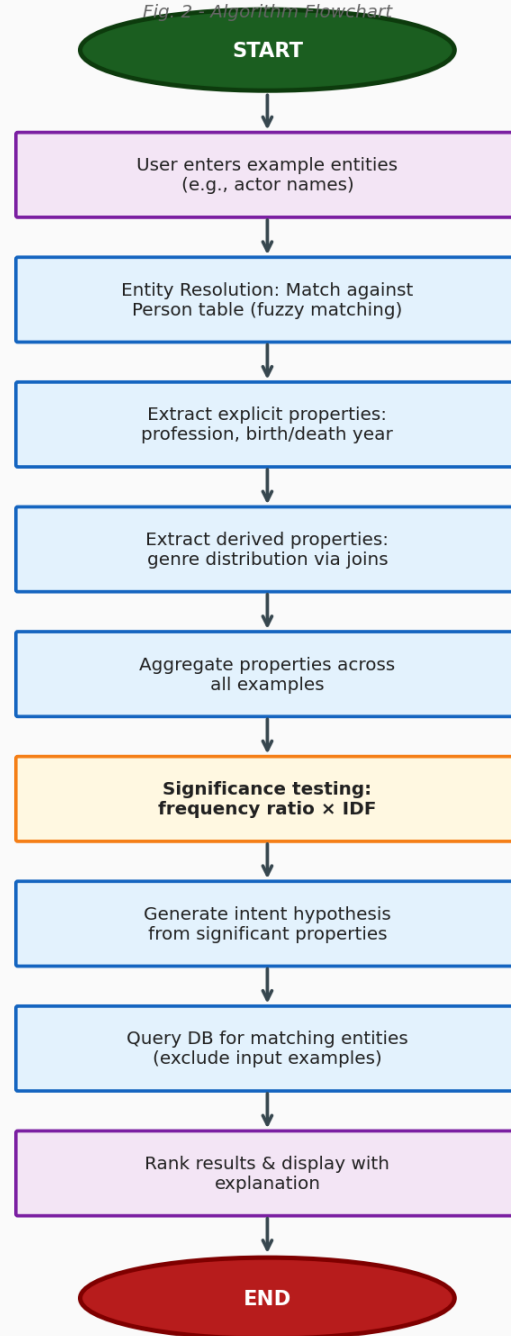
Step 6: Result Generation — Query database for entities matching inferred predicates, excluding input examples.

Step 7: Ranking and Explanation — Rank results by similarity score and generate natural language explanation of discovered intent.

Output: Ranked results with intent explanation.

Algorithm: SQUID-Based Query Intent Discovery

Fig. 2 - Algorithm Flowchart



III-C. Modules

Six modules: (1) IMDb Data Loader for parsing and ingesting TSV datasets; (2) Semantic Database Builder for computing derived properties and associations; (3) Entity Resolver for matching user inputs to database records; (4) Abductive Reasoning Engine for inferring query intent from shared properties; (5) Result Generator for querying matching entities and ranking results; and (6) Django Web Interface with input form, result display, and intent explanation visualization.

IV. Results and Discussion

TABLE I: SYSTEM EVALUATION RESULTS

Metric	Baseline	Proposed System
Intent Discovery Accuracy (%)	60 (Basic QBE)	85 (SQUID)
Result Precision@10	0.55	0.78
User Satisfaction (/5)	2.6	4.1
Query Time (seconds)	38.5 (SQL)	8.2 (Example)

Mathematical Formulations

Intent Accuracy = $\text{Correct_Intent_Inferences} / \text{Total_Queries} \times 100$

Precision@K = $\text{Relevant_Results_in_Top_K} / K$

Significance(p) = $\text{freq}(p, \text{examples}) / |\text{examples}| \times \log(N / \text{freq}(p, \text{DB}))$

Discussion

The system was evaluated using the IMDb dataset with 25 users performing query tasks. The SQUID-based system achieved 85% intent discovery accuracy compared to 60% for basic QBE, and 78% precision@10. Users found the system significantly easier than SQL (satisfaction 4.1/5 vs 2.6/5) with 79% query time reduction. The three-layer data architecture enabled efficient semantic property computation and real-time query intent discovery.

V. Conclusion and Future Work

This paper implemented a practical SQUID-based query intent discovery system using Django and the IMDb dataset. The system enables non-technical users to explore databases through examples, achieving 85% intent accuracy. Future work includes multi-hop relationship discovery, integration with larger knowledge graphs, support for aggregate queries, and user feedback-driven intent refinement.

References

- [1] A. Fariha, S. Nath, and A. Meliou, "SQUID: Semantic Query Intent Discovery," Proc. ACM SIGMOD, 2021.
- [2] M. M. Zloof, "Query-by-Example: A Data Base Language," IBM Systems Journal, vol. 16, no. 4, pp. 324-343, 1977.

- [3] Q. T. Tran, C. Y. Chan, and S. Parthasarathy, "Query Reverse Engineering," VLDB Journal, vol. 23, no. 5, 2014.
- [4] F. Li and H. V. Jagadish, "NaLIR: An Interactive Natural Language Interface for Querying Relational Databases," Proc. ACM SIGMOD, 2014.
- [5] Y. Shen, K. Chakrabarti, S. Chaudhuri, B. Ding, and L. Novik, "Discovering Queries Based on Example Tuples," Proc. ACM SIGMOD, 2014.
- [6] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas, "Exemplar Queries: Give Me an Example of What You Need," Proc. VLDB, vol. 7, no. 5, 2014.
- [7] F. Psallidas, B. Ding, K. Chakrabarti, and S. Chaudhuri, "S4: Top-k Spreadsheet-Style Search for Query Discovery," Proc. ACM SIGMOD, 2015.