SemExplorer: A User Interface for Semantic Approach to Customized Dataset Search

Zixin Wei
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
zixinwei1@link.cuhk.edu.cn

Jun Han
The Hong Kong University of Science and Technology
Hong Kong, China
hanjun@ust.hk

Xiaolin Han
The Northwestern Polytechnical University
Xi'an, China
xiaolinh@nwpu.edu.cn

Chenhao Ma*
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
machenhao@cuhk.edu.cn

Abstract

In response to the increasing complexity of data and the widespread availability of diverse datasets for modeling complex relationships, there is a crucial need for efficient and customizable search capabilities. Many data scientists and researchers still need to manually browse through extensive catalogs to find suitable datasets for their studies. To address this challenge, we introduce SemExplorer, a framework designed to enhance semantic analysis and information retrieval for user-defined needs. Our system processes and interprets complex queries in natural language using large language models (LLMs), converting them into vector representations and statistical filters. When presenting search results, our system uses context dependencies to highlight important information, helping users quickly locate the results they need. Overall, SemExplorer is a robust tool that not only improves the efficiency of finding needed datasets but also enhances the precision of search outcomes by leveraging semantic analysis to deeply understand and filter network data. A demo of our system is available online[1].

CCS Concepts

• **Information systems** → **Search interfaces**; • **Computing methodologies** → *Natural language processing*.

Keywords

Natural Language Processing, Large Language Models, Information Retrieval, Semantic Search, Datasets

ACM Reference Format:
Zixin Wei, Jun Han, Xiaolin Han, and Chenhao Ma. 2025. SemExplorer: A User Interface for Semantic Approach to Customized Dataset Search. In *Companion of the 2025 International Conference on Management of Data (SIGMOD-Companion '25), June 22–27, 2025, Berlin, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3722212.3725133

*Chenhao Ma is the corresponding author.
[1]http://ns2.nilou.top

1 Introduction

Datasets lie at the heart of research in many domains, yet effectively locating the needed datasets from vast repositories remains a persistent challenge—especially when various dataset types must be considered [8]. In this paper, we focus on network datasets as a primary example. These datasets capture complex relationships across diverse areas, including social networks [5], e-commerce platforms [7], and biological systems [1], where different graph models (e.g., directed, undirected, temporal, bipartite) serve as powerful representations. Data scientists and researchers have been building myriad graph-based analytical solutions for tasks such as community detection, recommendation systems, and protein-protein interaction analysis.

Network datasets vary significantly by scale—ranging from a few hundred to billions of vertices—and by domain, spanning social networks, protein-protein interactions, road networks, and beyond. Accordingly, a large, heterogeneous repository of network datasets demands an efficient way to retrieve the most relevant items. Although several websites host large quantities of network data, their search functionality often proves insufficient. For instance, KONECT [9] contains over a thousand networks but lacks a search feature, while Network Repository [14] supports keyword queries only. Traditional approaches like keyword matching are insufficient for addressing complex or nuanced user requirements, as they fail to capture semantic relationships between dataset properties and user intentions.

Beyond networks, other structured data, such as web tables, also require more advanced, context-sensitive search mechanisms. In such cases, researchers might prioritize features like the presence of headers or indexing formats, illustrating a different set of constraints compared to network data [3]. These varying needs further underscore the importance of flexible, extensible search solutions capable of handling diverse data models.

To address these challenges, SemExplorer was developed as a semantic search system to help data scientists navigate vast collections of datasets more intuitively. Firstly, we employ a schema-based approach, which summarizes datasets sharing the same model (for graphs, tables, etc.) using a common schema that captures key properties. Then, moving beyond straightforward keyword matching, SemExplorer utilizes advanced natural language processing techniques. It embeds textual descriptions and user queries using models like word2vec [11], enabling more accurate matching based on semantic similarity, rather than mere keyword overlap.

For scalability, SemExplorer integrates approximate nearest-neighbor algorithms [2], providing a balance of speed and accuracy. Its user-friendly interface supports natural language queries and filtering, significantly reducing manual effort required to locate suitable datasets.

A video for our demo is available online[2], and our code is available on Github[3].

2 SemExplorer

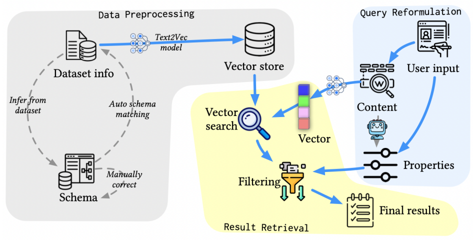


Figure 1: Workflow of the System

2.1 Related Work

Semantic search has significantly evolved by integrating pre-trained language models and vector databases, enabling deeper context comprehension and more intuitive search functionalities [13, 15]. These methods transcend keyword-based approaches by comparing the semantic meaning of query terms, rather than matching words verbatim. However, current solutions primarily focus on text-based tasks and lack support for dataset discovery that requires alignment with specific properties (e.g., graph structure or relational schema). SemExplorer addresses this gap by incorporating property constraints into a semantic search paradigm, bridging the divide between free-text interpretation and structured dataset attributes.

2.2 Overview of SemExplorer

SemExplorer is an innovative tool designed to search vast collections of datasets using natural language efficiently. As shown in Figure 1, the workflow is structured around three major stages: *Pre-prosessing* of datasets, *Query Reformulation* from user input, and *Result Retrieval*. Each stage is critical to the system's ability to interpret and process user queries effectively. A detailed description of each stage is provided in the subsequent sections.

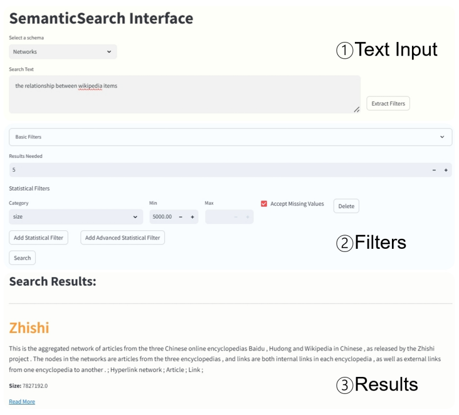
2.3 Data Pre-processing

Effective data pre-processing underpins SemExplorer's ability to handle natural language descriptions during dataset search. First, each dataset's metadata is split into two parts: (1) a set of structured properties stored in a relational table, and (2) a textual description consolidated from various sources. Textual descriptions are converted into dense vectors via embedding models like word2vec [11], allowing SemExplorer to semantically match these descriptions against user queries—also transformed into dense vectors—in a shared embedding space.

A key challenge arises when integrating data from diverse sources. To address this, SemExplorer uses an automatic schema-matching process. A master schema encompasses all potential metadata fields, and each new attribute is checked for matches within this schema. If no exact match is found, spaCy [6] measures semantic similarity with existing attributes; attributes surpassing a similarity threshold are considered matches. If the threshold is not met, a Large Language Model is consulted, using a structured JSON response to identify the best fit or to flag unmatched attributes. This approach streamlines data integration from repositories like SNAP [10], KONECT [9], and Network Repository [14].

For efficient retrieval, the system employs Annoy [2] to build approximate nearest neighbor indices, enabling rapid identification of datasets that align semantically with user queries. These indexing and matching strategies ensure that SemExplorer can scale effectively, maintaining speed and accuracy as new datasets added.

2.4 Query Reformulation


SemanticSearch Interface

①Text Input

Select a schema: Networks

Search Text: the relationship between wikipedia items

Extract Filters

Basic Filters

Results Needed: 5

Statistical Filters

Category: size | Min: 5000.00 | Max: | ✓ Accept Missing Values | Delete

Add Statistical Filter | Add Advanced Statistical Filter

②Filters

Search

Search Results:

Zhishi

This is the aggregated network of articles from the three Chinese online encyclopedias Baidu , Hudong and Wikipedia in Chinese , as released by the Zhishi project . The nodes in the networks are articles from the three encyclopedias , and links are both internal links in each encyclopedia , as well as external links from one encyclopedia to another . ; Hyperlink network ; Article ; Link ;

Size: 7827192.0

③Results

Read More

Figure 2: Web Interface Overview

To enhance the user experience with complex queries, SemExplorer employs a *Query Reformulation* stage that divides user input into two key components: *Content* and *Properties*. The *Content* describes the thematic focus of the dataset (e.g., "relationship between users on social media"), while the *Properties* capture structural or quantitative attributes (e.g., dataset size, maximum degree), which are usually stored as boolean, integer, or float variables. In practice, a dataset may expose numerous properties—over 80 in our network dataset example—making it impractical for users to manually navigate and configure each constraint.

To streamline this process, users can enter natural language queries in a text box (see Figure 2). Through the Dashscope[4] NLP API, property-related phrases within the query are automatically extracted into filters, leaving only thematic details in the text box. Users remain free to edit or refine these filters after they are generated, ensuring accurate alignment with the intended search criteria.

[2]Click to Access the Video
[3]Click to Access the Code

[4]https://dashscope.aliyun.com/

Below is an illustrative example of the query reformulation process on our network datasets example:

> An undirected network about the relationship between users on social media, with more than 10000 nodes, a max degree between 500 and 2000, and an average degree smaller than 50.

The extracted results are summarized in Table 1, demonstrating the system's ability to delineate and structure user inputs effectively.

Table 1: Extracted Content and Filters

Item	Value
Content	the relationship between users on social media
Directed	FALSE
Size	≥ 10000
Max Degree	≥ 500 and ≤ 2000
Avg Degree	≤ 50

A core challenge in automating query reformulation is ensuring that Large Language Models produce structured outputs. Although fine-tuning can guide LLMs toward specific output formats [16], using existing models is often more practical. In SemExplorer, the LLM's responses are requested in SQL format, capitalizing on the widespread presence of SQL examples in training data. Specifically:

- The prompt contains a `CREATE TABLE` statement detailing all the properties (fields) relevant to a dataset.
- A `SELECT` statement, generated by the LLM, encodes the properties extracted from the query within a `WHERE` clause.

By providing the LLM with a clear SQL template, the system ensures that responses conform to the target schema. These SQL outputs are then parsed to identify the exact filters requested by the user, effectively translating plain-language constraints into actionable property constraints for the search. This structured approach simplifies the integration of complex queries, ensuring both user-friendly input and systematically validated results.

2.5 Result Retrieval

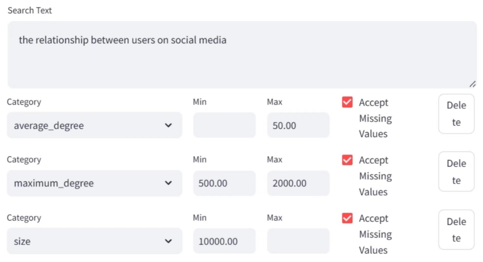
SemExplorer first employs approximate nearest neighbor (ANN) matching to efficiently pinpoint datasets with textual descriptions most similar to the user's query. Next, it filters out any candidates that fail to meet the specified property constraints, returning a final set based on user preferences. Each result provides a hyperlink for quick access, along with its title, description, and key properties. To highlight relevancy, SemExplorer uses spaCy [6] to extract keywords from the user query, then leverages WordNet [4, 12] to locate related or synonymous terms. These terms are emphasized in the dataset descriptions, aiding users in quickly identifying the most pertinent information.

3 Tool Usage

3.1 Searching Interface

SemExplorer offers a web user interface implemented with Streamlit [17] that enables the rapid creation of interactive web applications. Figure 2 shows an overview of a running demo. As shown in the figure, the user interface is structured into three primary fields: ①Text Input, ②Filters, and ③Results.

Field ① provides a text input area where users can describe the substantive query focus. To simplify the process of defining search parameters, when the *Extract Filters* button is clicked, the textual description is parsed into relevant filters and automatically applied in Field ② in 3 to 5 seconds. For instance, in our network example, we enter the previous sample from *Section 2.4* in Field ①. Clicking *Extract Filters* button, the result of extraction is shown in Figure 3.



Search Text: the relationship between users on social media

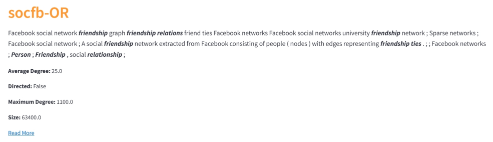
Category	Min	Max	Accept Missing Values	
average_degree		50.00	✓	Delete
maximum_degree	500.00	2000.00	✓	Delete
size	10000.00		✓	Delete

Figure 3: Result of Extracion

Meanwhile, Field ② allows the user to adjust the filters. In *Basic Filters*, the user can select boolean conditions, such as *Directed*. In *Statistical Filters*, the user can add various filters with maximum and/or minimum requirements. In addition, users can choose to accept or exclude results with missing values for each filter.

After clicking the *Search* button, the results will be displayed in Field ③ within 3 to 6 seconds, depending on the complexity of the query. Each displayed result includes a title, description, and the properties the user is concerned about. The keywords inside the description of the dataset are highlighted based on the user's query to facilitate quick and easy identification.

Continuing with the example above, we can cancel all the *accept missing values* checkboxes and click the *Search* button. As shown in Figure 4, the result meets our needs in terms of both content and properties.



socfb-OR

Facebook social network ***friendship*** graph ***friendship relations*** friend ties Facebook networks Facebook social networks university ***friendship*** network ; Sparse networks ; Facebook social network ; A social ***friendship*** network extracted from Facebook consisting of people (nodes) with edges representing ***friendship ties*** . ; ; Facebook networks ; ***Person*** ; ***Friendship*** , social ***relationship*** ;

Average Degree: 25.0

Directed: False

Maximum Degree: 1100.0

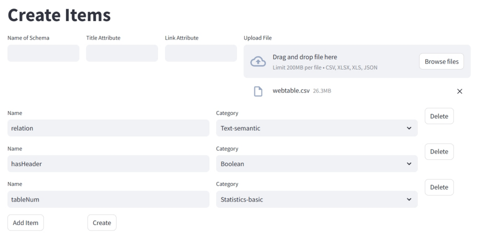
Size: 63400.0

Read More

Figure 4: Result of Searching

3.2 Schema Creation Interface

SemExplorer provides a schema creation interface that allows users to establish a dedicated "searching area" for their own datasets. As shown in Figure 5, users can upload JSON files (which directly store the target schema) or table files (which are analyzed to derive potential attributes). Based on column content, SemExplorer automatically categorizes each attribute as text, statistic, or boolean, and users can revise these assignments before finalizing the schema. This streamlined process ensures swift, accurate schema configuration and integration of new datasets.


Create Items

Name of Schema:
Title Attribute:
Link Attribute:
Upload File: Drag and drop file here — Limit 200MB per file • CSV, XLSX, XLS, JSON — Browse files
webtable.csv 26.3MB

Name	Category	
relation	Text-semantic	Delete
hasHeader	Boolean	Delete
tableNum	Statistics-basic	Delete

Add Item Create

Figure 5: A Sample of Schema Creation Interface

3.3 Datasets Import Interface

After creating a schema, users can upload new datasets. First, the user selects a search area and uploads a local table file. SemExplorer automatically matches the uploaded file's attributes to those defined in the schema, as illustrated in Figure 6. Most attributes align correctly, but users can manually correct any mismatches. Click the *Submit* button to save the dataset info, and the *Upload Embedding* button to process the dataset for incorporation into the vector search index. This two-step approach ensures that both structured data and text embeddings are efficiently integrated for future queries.

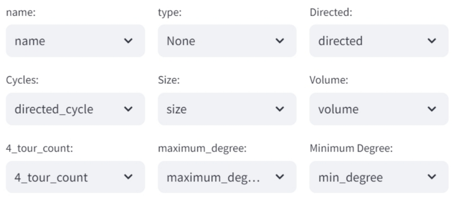

name: name
type: None
Directed: directed
Cycles: directed_cycle
Size: size
Volume: volume
4_tour_count: 4_tour_count
maximum_degree: maximum_deg...
Minimum Degree: min_degree

Figure 6: A Sample of Automatical Matching

4 Limitation

Despite its robust search functionalities, SemExplorer remains dependent on the quality and comprehensiveness of dataset descriptions within its database. Poorly annotated datasets hamper the system's ability to return relevant results, forcing researchers to spend additional time verifying suitability. Moreover, when metadata is incomplete (e.g., missing detailed statistical attributes), the system may struggle to filter datasets effectively.

Another limitation comes from LLMs. While these models excel at interpreting natural language, they can occasionally yield different outputs for the same query. Such variability may affect user confidence in the system's consistency.

To address these issues, future refinements should include:

- *Dataset Quality Control:* Broader, more well-documented datasets and improved data curation processes.
- *Model Consistency:* Techniques to reduce stochastic variations and increase the reproducibility of results.

5 Conclusion

In this paper, we introduced SemExplorer, a semantic dataset search framework that combines embedding-based natural language processing with approximate nearest neighbor retrieval. It translates user queries into actionable constraints, enabling intuitive and efficient dataset discovery.

SemExplorer is deployed as a web-based interface, suitable for both casual and expert users. Its open-source design encourages community contributions, while the *Schema Creation* and *Data Import* features allow easy integration of user-defined datasets. By streamlining large-scale dataset retrieval and providing a more intelligent, customizable search process, SemExplorer unlocks new possibilities for research across multiple domains.

Acknowledgments

This work was supported by the NSFC grant (62302421, 62302422, 62302397).

References

[1] Tero Aittokallio and Benno Schwikowski. 2006. Graph-based methods for analysing networks in cell biology. *Briefings in Bioinformatics* 7, 3 (2006), 243–255. https://doi.org/10.1093/bib/bbl022

[2] Erik Bernhardsson. 2024. *Annoy: Approximate Nearest Neighbors in C++/Python.*

[3] Christian Bizer, Robert Meusel, Oliver Lehmberg, Dominique Ritze, and Sanikumar Zope. 2015. Web Data Commons - Web Table Corpus 2015 / Relational Data. School of Business Informatics and Mathematics, Universität Mannheim. https://doi.org/10.7801/207 Available online: https://madata.bib.uni-mannheim.de/207/.

[4] Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA.

[5] D. Goldenberg. 2021. Social Network Analysis: From Graph Theory to Applications with Python. *arXiv preprint arXiv:2102.10014* (2021). https://ar5iv.labs.arxiv.org/html/2102.10014

[6] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. (2020). https://doi.org/10.5281/zenodo.1212303

[7] Zan Huang, Wingyan Chung, and Hsinchun Chen. 2004. A graph model for E-commerce recommender systems. *Journal of the American Society for Information Science and Technology* 55, 3 (2004), 259–274. https://doi.org/10.1002/asi.10372

[8] Madelon Hulsebos, Wenjing Lin, Shreya Shankar, and Aditya Parameswaran. 2024. It Took Longer than I was Expecting: Why is Dataset Search Still so Hard?. In *Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics.* 1–4.

[9] Jérôme Kunegis. 2013. KONECT – The Koblenz Network Collection. In *Proc. Int. Conf. on World Wide Web Companion.* 1343–1350. http://dl.acm.org/citation.cfm?id=2488173

[10] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (Eds.), Vol. 26.

[12] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[13] Malte Pietsch, Tanay Soni, Branden Chan, Timo Möller, and Bogdan Kostić. 2024. Haystack. https://github.com/deepset-ai/haystack.

[14] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. http://networkrepository.com

[15] Jan Rygl, Jan Pomikálek, Radim Řehůřek, Michal Růžička, Vít Novotný, and Petr Sojka. 2017. Semantic Vector Encoding and Similarity Search Using Fulltext Search Engines. *arXiv preprint arXiv:1706.00957* (2017).

[16] Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction. (2023). https://doi.org/10.48550/arXiv.2310.03668

[17] Streamlit. 2024. *Streamlit.* Snowflake Inc. https://www.streamlit.io