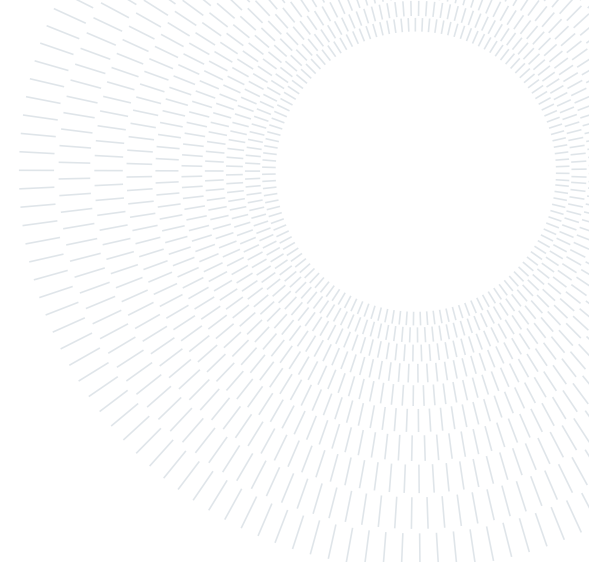




POLITECNICO
MILANO 1863

**SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE**



EXECUTIVE SUMMARY OF THE THESIS

Improving Data Discoverability with LLM: a RAG based method

LAUREA MAGISTRALE IN COMPUTER SCIENCE AND ENGINEERING - INGEGNERIA INFORMATICA

Author: RICCARDO TERRENZI

Advisor: PROF. PIERLUIGI PLEBANI

Co-advisors: PROF. SERKAN AYVAZ, MATTEO FALCONI

Academic year: 2024-2025

1. Introduction

Global data production has reached unprecedented volumes, fueled by pervasive technologies such as the Internet of Things (IoT), Cloud Computing, and the widespread adoption of intelligent systems. In this scenario, businesses and organizations face a crucial challenges on how to leverage a growing amount of heterogeneous data, often distributed and lacking a shared schema. In particular, difficulties arise when organisations want to share data between them.

The Data Space concept, presented in [2] in 2005 as a flexible alternative to classic DBMS, is now back into vogue as a solution to new needs, such as efficient interorganizational data sharing, offering a flexible and decentralized data management model based on the concept of coexistence rather than integration.

A data space allows the management of heterogeneous information sources while maintaining the autonomy of individual datasets, allowing for more flexibility and scalability.

In order to share share and consume efficiently data in a data space, the standard is to adopt the FAIR principles (Findability, Accessibility, Interoperability, Reusability). However, one of the main limitations of such environments lies in

Findability, i.e., the ability of users to efficiently locate datasets relevant to their needs. Current solutions, based mainly on metadata consultation and filtering, are often inadequate: metadata is sometimes incomplete, not very descriptive, or requires specialized terminological skills on the part of the user. This means that datasets within a data space should be at all times findable, accessible (according to their respective access policies), reusable even in domains other than data creation and management, and interoperable, i.e., not restricted by a different source or format for their use.

However, the aspect of Findability, i.e., the ability of users to efficiently locate datasets relevant to their needs in a data space, is still a challenge. Current solutions, based mainly on metadata consultation and filtering, are often inadequate: metadata is often incomplete, not very descriptive, or requires specialized terminological skills on the part of the user.

Can we search for a dataset in a data space relying on the dataset itself and not on metadata? This is the context for this research, whose main objective is to design and implement an intelligent method for Dataset Discovery in data spaces, which overcomes the limitations of metadata and keyword-based techniques and makes dataset search accessible even to users who are

not domain experts. The core of the proposed approach is the integration of Large Language Models (LLMs), known for their semantic and natural language understanding capabilities, within a Retrieval Augmented Generation (RAG) pipeline.

Through the development and evaluation of a prototype system, this research aims to demonstrate that the use of LLMs can bridge the gap between the complex structure of data spaces and the expressive and informational needs of end users. Moreover, our results show that this method is functional and could work as a base for future works aiming at enhancing data spaces using LLMs.

2. LLM-powered dataset discovery method

The main objective of this thesis was to develop an intelligent method capable of improving the Dataset Discovery process within a data space, overcoming the limitations of metadata or keyword-based search mechanisms. Metadata is often incomplete or superficial, thus often leading to unsatisfactory results, that is why we decided to work directly on the datasets using LLMs. Moreover, keyword-based queries can be a limitation for data space users, as they usually do not know the schema of the data sources they are searching for in advance. This means that situations of semantic ambiguity can arise where fields can be expressed in different ways, for example, the *ID* field of a dataset could be specified in different ways, such as *userID*. Another factor we consider is the fact that a data space can contain very different sources belonging to different domains. As a result, there may be users who are interested in datasets but are not experts in the domain of the dataset. For example, a Data Scientist may be looking for specific data on the treatment of diabetes, without being an expert in Medicine or Diabetes. In this situation the user is not expert about a domain and lacks the specific terminology to propose an efficient query. The idea, therefore, is to use the power of LLMs to overcome this semantic ambiguity and allow users to specify queries in natural language, thus specifying their requirements and still obtaining the desired result.

To achieve this goal, a pipeline has been designed that combines the use of Large Language

Models (LLMs) with a Retrieval Augmented Generation (RAG) architecture. The entire system is divided into two main phases: the offline phase and the online phase.

2.1. Offline phase

The Offline phase consists of those components that process all the Datasets in the data space and must be performed once for each resource contained in the data space, as we can observe in Figure 1.

In this phase we prepare and preprocess every dataset. We create a semantic and a statistic profile for each dataset by using an LLM. We use the profiles to create a prompt and use an LLM to generate pseudo-queries about the dataset, which we then store in a Vector DB.

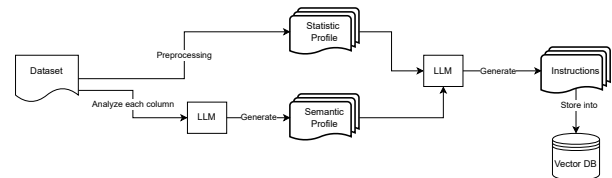


Figure 1: Offline phase of the method.

2.1.1 Preprocessing and Profiles

Data spaces usually contain heterogeneous data, in different formats and from different sources. For time reasons we decided to work only with datasets in CSV format, in order to create a system that could serve as a basis for future expansion to multiple formats. We opted for the CSV format as it is a standard format used by many companies to manage and share data. Moreover, structured data like tabular data are particularly difficult for LLMs to process, so it was particularly interesting to start from this base.

The aim of the project was to develop a method to find datasets in a easy way starting from natural language queries. In particular we are interested in the "meaning" of the whole dataset and not in specific cell-level contents. So, our focus is more on the semantics of the schema and on range and frequency of values, which can give us a perspective of what the dataset contains in general.

Moreover, we want to process datasets using LLMs and it can be problematic to fit a whole dataset in a fixed length prompt. There are feasible options like sampling, but we decided, in-

stead, to generate two profiles, one semantic and one statistic, inspired by the approach used in [5]. Both the profiles are text documents and will be fed to an LLM through prompt in the next steps of the method, in this way we can work with any length datasets.

The statistic profile contains:

- Minimum, maximum, and average values for numeric columns;
- Most frequent values for categorical columns;
- Summary distributions of the data.

We need this profile during the instructions generation phase to provide the LLM with quantitative information based on the dataset. This aspect is particularly interesting for managing ranges of numerical values, most frequent categories, or temporal data. All this without the LLM having direct access to the dataset, for the statistic profile generation we only used a profiling library in Python. The result will be a text document containing the summary.

Then we generate the semantic profile, which includes:

- The description and meaning of each column;
- The implicit relationships between columns;
- A thematic classification of each column and the dataset.

In this case we use an LLM to analyze the schema and the meaning of every column of the dataset. The result will be as well a text document containing the summary. This profile allows the LLM that generates the Instructions to have accurate information about the intrinsic meaning of the dataset and the meaning of each individual column in it.

2.1.2 Instructions

Starting from the profiles we have generated, we now want to generate Instructions, which are pseudo-queries that attempt to represent hypothetical and probable queries that a user might make to find that specific dataset. They also aim to represent the dataset in its entirety, both semantically and quantitatively, as they will be used later on in the method to retrieve the best datasets. The idea is developed from a concept presented in [4], a work in which the authors generate questions from scientific papers in PDF

format and use them to create a RAG pipeline. In our case, however, we generate these pseudo-queries that cover the entire meaning and content of the dataset, then save them in a vector DB and use them for dataset retrieval. This way, we do not need to serialize the dataset in any way, nor do we need to resort to dataset sampling techniques.

2.2. Online Phase

The Online phase consists of those components and processes that accompany the user from the query to the final response, as we can see in Figure 2.

In the online phase, the user interacts with a prompt where they can specify their query in natural language to find the datasets they need. We optimize the query received by the user by Internal Expansion and Decomposition. Then we retrieve the best results and rerank them using an LLM.

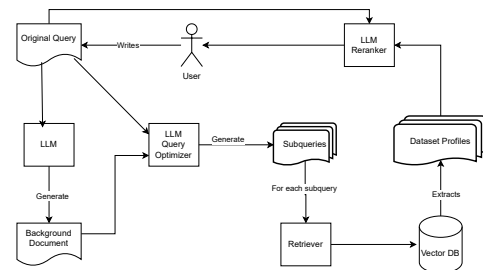


Figure 2: Online phase of the method.

2.2.1 Query Optimization

A natural language query is very convenient for a user, who can easily specify what they want in their own language. However, it can be complex for a system to handle natural language queries. In our case, to better manage the query, we optimize it with an approach based on the work presented in [1]. In particular, we exploit the idea of extracting a Background Document, i.e., a text that contains the information that the LLM intrinsically has about the query topics and considers essential for understanding the query itself.

The optimization is divided into the following phases:

1. **Internal Expansion:** starting from the query, we extract, using an LLM, a Background document that includes essential

information for understanding the query. This document will be included in the prompt used in the second phase of query optimization.

2. **Decomposition:** we break down the original query using an LLM, thus obtaining a list of clear and direct subqueries, which facilitate the retrieval phase. It should be noted that these subqueries will be very similar in form and content to the Instructions we generated in the Offline phase.

This process allows us to effectively manage even the most complex queries. In particular, the use of the background document is extremely helpful when the query is not very specific, as the subqueries generated are enriched with keywords not present in the original query, thus leading to a much more accurate retrieval. This allows even users who are less familiar with a domain to find the specific data they are looking for. Even in the case of more specific, complex queries, thanks to our optimization, we are able to manage them efficiently without losing detail.

2.2.2 Retrieving

The general idea behind our retrieval step is to exploit the similarity between the instructions generated in the Offline phase and the subqueries generated in the Online phase. To make the most of this scenario, we decided to use a vector DB, in which all the instructions generated from the datasets are saved. Each instruction is paired with metadata:

- Dataset Identifier;
- Semantic Profile;
- Statistic Profile;

After the decomposition, each subquery is transformed into a vector and compared with the instructions saved in the offline phase. Then, for each subquery, we extract the most relevant instructions and their metadata, which we will use in the next step of the method.

2.2.3 Reranking

At this stage, we have a list of the best candidates obtained from the retrieval. For each candidate, we have the semantic and statistical profiles available. At this point, an LLM performs a semantic evaluation between the user’s query and each candidate dataset, assigning a

Listing 1 Example of a keyword query.

```
1      'DS1-E-0005': 'births by
      ↪ month'
```

relevance score. For the evaluation of the candidates we use a listwise unsupervised approach, basically we provide a LLM with the query of the user and the whole list of candidates. With a listwise evaluation the length of the prompt can be a problem since we want to provide the LLM with the list of all candidates, but in this case it’s not a problem since we can decide how many candidates we retrieve during the retrieval step. The results are then sorted according to this evaluation, returning a final set of datasets consistent with the user’s information expectations.

3. Evaluation

To validate the effectiveness of the proposed method in improving Dataset Discovery within a data space, a systematic experimental evaluation was conducted. The goal was to measure the system’s ability to return relevant datasets in response to natural language queries.

3.1. Benchmark

The experiment was conducted using the NTCIR-15 Data Search benchmark [3], an established standard for evaluating data search systems. This benchmark includes:

- a corpus of heterogeneous datasets from the statistical and governmental domains;
- a set of keyword queries, see an example in Listing 1;
- a ground truth (human relevance judgments) for each query, which allows for the calculation of objective metrics.

3.2. Evaluation Methodology

The system’s performance was measured using classic information retrieval metrics:

- Precision@k: how many datasets proposed as relevant are actually relevant;
- Recall: how many of the relevant instances were proposed in the response;
- nDCG@k (Normalized Discounted Cumulative Gain): measures how well the system

ordered the proposed datasets, considering both the relevance of each proposed dataset and the position in which it was inserted.

3.3. Experimental setup

To evaluate our system, we had to make some trade-offs, as our system is currently only able to work with tabular datasets in CSV format, while the NTCIR-15 benchmark has many different types of dataset formats. This led us to select a subset of the benchmark to work on. In particular, we selected queries for which the relevant datasets are in CSV format. This allows us to effectively evaluate our system working with queries which have as results only CSV datasets. In this case, we consider that our system, being evaluated on a subset of the benchmark, may perform better than the systems presented in the benchmark since our system has to find the best candidates for a query between significantly less datasets.

For this reason, in order to make the evaluation fairer, we decided to implement a baseline with BM25, a solution widely used in search engines and information retrieval systems. The baseline will be evaluated on the same subset of data as our system.

Another interesting aspect we evaluated is the type of query. In fact, in the benchmark, the queries are simply a set of keywords, which is limiting. For this reason, we used an LLM to convert these keywords queries into complex queries in natural language. Then we submitted them to our system, so as to evaluate it in this context as well, which is more similar to how an end user would use it. We can observe the difference between a keyword query and its respective natural language we generated in Listing 2.

In the following section, we can see the results of our evaluation.

3.4. Results

After testing our system on both keyword queries and natural language queries and the baseline on keyword queries, we obtain the results shown in Table 1, where our system is called *OS*, the variant that works with natural language queries is called *OSNL* and the baseline is called *BM25*:

Listing 2 Example of keyword query converted into natural language.

```

1 'DS1-E-0005': 'births by month'
2 'DS1-E-0005': 'Find comprehensive
  ↳ statistical data and detailed
  ↳ analyses on the number of
  ↳ births categorized by each
  ↳ calendar month, including
  ↳ historical trends, regional
  ↳ variations, seasonal
  ↳ patterns, and potential
  ↳ correlations with
  ↳ socio-economic,
  ↳ environmental, or healthcare
  ↳ factors over the past several
  ↳ decades.'
```

Comparison of results

	Precision@10	Recall	nDCG@10
BM25	0.325	0.488	0.455
OS	0.560	0.8	0.679
OSNL	0.54	0.8	0.632

Table 1: Comparison of results of the baseline and our system.

Our system is significantly superior to the baseline in both scenarios, making it an interesting alternative, especially in scenarios where complex natural language queries are used or required.

The results show how a LLM approach to dataset search can overcome the limitations of traditional techniques, which often fail when the user does not use exactly the same keywords as those found in the metadata. The use of offline-generated semantic profiles, combined with the interpretative power of LLMs in the search phase, has resulted in a more flexible, accessible, and accurate system.

3.5. Limitations and Future Work

Despite the positive results, our method has some limitations. First of all, the current approach is able to work only on tabular data in CSV format, so adaptations will be needed to

include other data formats, both structured and unstructured (e.g., PDF, images). This would also lead to the possibility of a more extensive and detailed evaluation, for example using the whole dataset corpus included in NTCIR-15 dataset search.

Furthermore, in a future scenario, it might be interesting to consider solutions where there is direct access to data in order to provide cell-level answers. This would open to queries looking for dataset containing specific values, scenario that at the moment with our proposed method is not possible and in general was out of the scope of this work.

It should also be considered that an approach such as ours, which involves the use of LLMs, has higher costs and latency than, for example, the system implemented for the baseline. This means that a method such as ours is not perfect for every scenario, especially if cost is a problem. It would also be interesting to obtain evaluations from users, through questionnaires, to understand how convenient they find the use of such a system compared to traditional ones.

4. Conclusions

This research has introduced and validated a method for Dataset Discovery in a data space, capable of overcoming the intrinsic limitations of traditional techniques based on metadata and keywords. By integrating Large Language Models (LLM) within a Retrieval Augmented Generation (RAG) pipeline, it was possible to offer a system capable of understanding queries in natural language, significantly improving the accessibility and quality of the results returned.

The proposed approach demonstrated high performance in terms of accuracy, recall, and sorting of results, surpassing an established baseline such as BM25. The system was particularly effective even in the presence of complex queries or those formulated by users who were not experts in the domain.

Despite the constraints related to the type of data processed (CSV format) and the computational resources required by LLMs, the results obtained highlight the validity and potential of the approach. Future extension to heterogeneous formats, performance optimization, and the inclusion of feedback from end users represent the main directions for development.

In conclusion, this thesis provides a concrete contribution towards the construction of more intelligent, inclusive, and semantically aware data spaces, paving the way for new ways of interacting with and enhancing data.

References

- [1] Cong et al. Query optimization for parametric knowledge refinement in retrieval-augmented large language models. (arXiv:2411.07820), November 2024. arXiv:2411.07820 [cs].
- [2] Halevy et al. From databases to dataspace: a new abstraction for information management. *ACM Sigmod Record*, 34(4):27–33, 2005.
- [3] Kato et al. A test collection for ad-hoc dataset retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2450–2456, Virtual Event Canada, July 2021. ACM.
- [4] Mombaerts et al. Meta knowledge for retrieval augmented large language models. (arXiv:2408.09017), August 2024. arXiv:2408.09017 [cs].
- [5] Zhang et al. Autoddg: Automated dataset description generation using large language models. (arXiv:2502.01050), February 2025. arXiv:2502.01050 [cs].