INODE: Building an End-to-End Data Exploration System in Practice

Sihem Amer-Yahia
CNRS, University Grenoble Alpes, France

Georgia Koutrika
Athena Research Center, Greece

Martin Braschler
Zurich University of Applied Sciences, Switzerland

Diego Calvanese
Free University of Bozen-Bolzano, Italy

Davide Lanti
Free University of Bozen-Bolzano, Italy

Hendrik Lücke-Tieke
Fraunhofer IGD, Germany

Alessandro Mosca
Free University of Bozen-Bolzano, Italy

Tarcisio Mendes de Farias
Swiss Institute of Bioinformatics, Switzerland

Dimitris Papadopoulos
Infili, Greece

Yogendra Patil
CNRS, University Grenoble Alpes, France

Guillem Rull
SIRIS Academic, Spain

Ellery Smith
Zurich University of Applied Sciences, Switzerland

Dimitrios Skoutas
Athena Research Center, Greece

Srividya Subramanian
Max Planck Institute for Extraterrestrial Physics, Germany

Kurt Stockinger
Zurich University of Applied Sciences, Switzerland

ABSTRACT

A full-fledged data exploration system must combine different access modalities with a powerful concept of *guiding* the user in the exploration process, by being *reactive* and *anticipative* both for data discovery and for data linking. Such systems are a real opportunity for our community to cater to users with different domain and data science expertise.

We introduce INODE - an end-to-end data exploration system - that leverages, on the one hand, Machine Learning and, on the other hand, semantics for the purpose of Data Management (DM). Our vision is to develop a classic unified, comprehensive platform that provides extensive access to open datasets, and we demonstrate it in three significant use cases in the fields of Cancer Biomarker Research, Research and Innovation Policy Making, and Astrophysics. INODE offers sustainable services in (a) data modeling and linking, (b) integrated query processing using natural language, (c) guidance, and (d) data exploration through visualization, thus facilitating the user in discovering new insights. We demonstrate that our system is uniquely accessible to a wide range of users from larger scientific communities to the public. Finally, we briefly illustrate how this work paves the way for new research opportunities in DM.

1. INTRODUCTION

The Data Management (DM) community has been actively catering to Machine Learning (ML) research by developing systems and algorithms that enable data preparation and flexible model learning. This has resulted in several major contributions in developing ML pipelines, and formalizing algebras and languages to facilitate and debug model learning, as well as designing and implementing algorithms and systems to speed up ML routines [23]. Conversely, existing work that leverages ML for DM [25] is nascent and covers the use of ML for query optimization [14] or for database indexing [13]. This paper makes the case for democratizing *Intelligent Data Exploration* by leveraging ML for DM.

Traditionally, database systems assume the user has a specific query in mind, and can express it in the language the system understands (e.g., SQL). However, today, users with different technical backgrounds, roles, and tasks are accessing and leveraging voluminous and complex data sources. In many scenarios, they are only partially familiar with the data and its structure, and their user information needs are not well-formed. In such settings, *expanding traditional query answering to data exploration* *is a natural consequence and requirement and with it comes the need to redesign systems accordingly.* This need translates to several challenges at different levels.

(*Interaction*). Regarding interaction with the system, the biggest challenge is to enable the user to express her needs through a variety of **access modalities**, ranging from SQL and SPARQL to natural language (NL) and visual query interfaces, that can be used and intermingled depending on the user needs and expertise as well as the data exploration scenario. The second challenge is that of **user guidance**, i.e., users should be allowed to provide *feedback* to the system, and the system should leverage that feedback to improve subsequent exploration steps.

(*Linking*). Once a user need has been formulated and sent to the system, a search is executed over a (fixed) data set. Users may be aware which additional data sets could be of interest. However, they do not always know how to correctly link, integrate, and query more than one data source to generate rich information. This introduces the challenges of **data linking**, so that new data sources can be added to the system, as well as **knowledge generation**, so that queries over unstructured data can be supported. Both of these aim at enabling the continuous expansion of the "pool" of available data sources, thus making more data available to users.

(Guidance). Traditionally, the system will return to the user a set of tuples that concludes the search. There is a lot of work on how to improve performance for query workloads (predict future queries, build indices adaptively, etc.), but still the system has a rather passive role: anticipating or at best trying to predict the next query and then optimize its performance accordingly. Hence, the challenge of **system proactiveness** arises. The output is not only the set of results but also recommendations for subsequent queries or exploration choices. In our vision, the system guides the user to find interesting, relevant or unexpected data and actively participates in shaping the query workload.

In a nutshell, a full-fledged data exploration system must *combine different access modalities with a powerful concept of guiding the user* in the exploration process. It must *be reactive and anticipative*; co-shaping with the user the data exploration process. Finally, while data integration has been around for a while, *the ability to tie together data discovery and linking is a central question in an intelligent data exploration system.*

(Evaluation). An essential part of our proposal is the development of an evaluation framework to enable the end-to-end assessment of an intelligent data exploration system. This requires to formalize system metrics and human metrics that are necessary for data linking and integration, multi-modal data access, guidance, and visualization.

Related Work. Several systems address components of our vision. A number of them address interaction by enabling NL-to-SQL [3], SQL-to-NL [12] or both [11] (see a summary in [1]). Recommendation strategies can be leveraged to guide users [17]. Work on interactive data exploration aims at helping the user discover interesting data patterns based on an integration of classification algorithms and data management optimization techniques [6]. Each of the above-mentioned systems tackles specific data management challenges as so-called *insular solutions*. However, these insular solutions have not been integrated to tackle the end-to-end aspect of intelligent data exploration targeted at a wide range of different end users.

Combining all the challenges above requires an elaborated system whose multi-aspect behavior and functionality is the result of a synergy between disjoint technologies, and integrates them into a new ensemble. This gives rise to multiple approaches that vary in computational complexity, and raises new challenges that can benefit from recent advances in ML.

In summary, this paper makes the following contributions: One-size-does-not-fit-all when building a full-fledged data exploration system. For instance, the exploration operators are not all the same across different domains since exploring health data requires different semantics than exploring galaxies. Our aim is to encapsulate that semantics in higher level constructs, e.g. exploration by example, by natural language and by recommendation. Similarly, our aim is to build the components necessary for a full fledged system. We illustrate the need for intelligent data exploration with relevant use cases (Section 2). We describe INODE[1], a system that we are currently building as part of a project funded by the European commission (Section 3). To fully complete our vision, we provide open research challenges to be addressed at the intersection of DM and ML (Section 4).

2. USE CASES

In this section, we describe two of our three use cases - cancer research and astrophysics - and show how INODE can tackle them. The system is targeted for domain scientists as well as the general

[1] http://www.inode-project.eu/

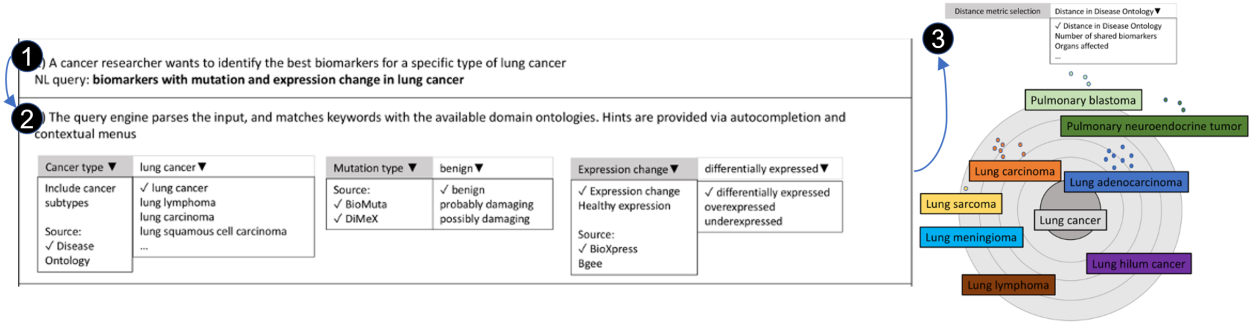

1
) A cancer researcher wants to identify the best biomarkers for a specific type of lung cancer
NL query: **biomarkers with mutation and expression change in lung cancer**
2
) The query engine parses the input, and matches keywords with the available domain ontologies. Hints are provided via autocompletion and contextual menus
Cancer type ▼
lung cancer ▼
Include cancer subtypes
Source:
✓ Disease Ontology
✓ lung cancer
lung lymphoma
lung carcinoma
lung squamous cell carcinoma
...
Mutation type ▼
benign ▼
Source:
✓ BioMuta
✓ DiMeX
✓ benign
probably damaging
possibly damaging
Expression change ▼
differentially expressed ▼
✓ Expression change
Healthy expression
Source:
✓ BioXpress
Bgee
✓ differentially expressed
overexpressed
underexpressed
3
Distance metric selection
Distance in Disease Ontology ▼
✓ Distance in Disease Ontology
Number of shared biomarkers
Organs affected
...
Pulmonary blastoma
Pulmonary neuroendocrine tumor
Lung carcinoma
Lung adenocarcinoma
Lung sarcoma
Lung cancer
Lung meningioma
Lung hilum cancer
Lung lymphoma

Figure 1: Natural language query interface with user assistance. Step 1: User enters a natural language query. Step 2: System parses query and matches keywords against the available ontology to enable term disambiguation; the user iterates the process. Step 3: System visualizes various cancer types that are similar to lung cancer. The distance metric between the diseases can be chosen by the user, e.g. by semantic distance.

public.

Use Case 1: Cancer Research (Natural Language and Visual Data Exploration). Fred is a biologist who studies cancer. His goal is to find which specific biomarkers are indicators for a certain type of lung cancer. He needs natural language exploration.

INODE offers support for NL queries, query recommendations, and interactive visualizations triggered by NL queries (see Figure 1). For instance, Fred starts with **a request in NL** for the topics related to lung cancer but is not sure how to continue after inspecting the results. INODE steps up and **recommends different options**: to expand the search using experimental drugs for treating lung cancer, or to focus on a subset of lung cancer types associated with a certain gene expression. Fred chooses to **expand his search** to one of the recommended topics, and receives a new list of lung cancers, drugs and genes. Additionally, INODE **explains in NL** how results are related. That helps him in selecting experimental drugs for certain gene expressions. After a few such queries, the system **visually analyzes the results** for Fred to study. Fred learns about the similarity between different types of cancer based on distance metrics that he can choose. In order to enable such data exploration, several different databases need to be integrated and potentially be correlated with findings from research papers.

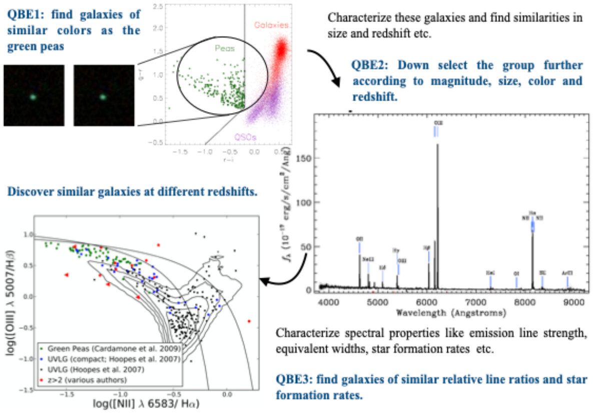
Use Case 2: Astrophysics (Exploration with SQL-Pipelines). In the era of big data, astronomers need to analyze dozens of databases at a time. With the ever increasing number of publicly available astronomical databases from various astronomical surveys across the globe, it is becoming increasingly challenging for scientists to penetrate deep into the data structure and their metadata in order to generate new scientific knowledge. Sri, an astrophysicist, explores astronomical objects in SDSS, a large sky survey database [2]. Sri would like to examine Green Pea galaxies, first discovered in a citizen science project called 'Galaxy zoo', that recently gained attention in astronomy as one of the potential sources that drove cosmic reionization.

Figure 2 shows a sequence of three consecutive processes of analyzing astrophysics data. Sri relies on selected examples at each step and requests to see comparable ones. In the first query, she **asks to find galaxies with similar colors** as Green Pea galaxies. She then **requests objects with similar spectral properties, like emission line measurements, star formation rates etc.,** as those returned in the first step. The last query finds **similar galaxies in terms of their relative ratios and strength of emission lines**. As a result, Sri discovers that green pea emission line ratios are similar to high redshift galaxies.

INODE guides any user in making such new discoveries in an intuitive simpler way, without having to write complicated SQL queries or perform manual analysis of thousands of galaxies. For instance, INODE helps a user **choose among similarity dimensions** rather than rely on her ability to provide them. Additionally, INODE shows to the user **alternative queries** to pay attention to, thus increasing the chances of making new discoveries.

Crucially, INODE can be extended with additional resources which requires close interaction with domain scientists. Detailed user guides are in preparation.

[2] https://www.sdss.org/



QBE1: find galaxies of similar colors as the green peas

Characterize these galaxies and find similarities in size and redshift etc.

QBE2: Down select the group further according to magnitude, size, color and redshift.

Discover similar galaxies at different redshifts.

Characterize spectral properties like emission line strength, equivalent widths, star formation rates etc.

QBE3: find galaxies of similar relative line ratios and star formation rates.

Figure 2: Exploring astrophysics data.

3. CURRENT INODE ARCHITECTURE

The main novelty of INODE is bringing together different data management solutions to enable *intelligent data exploration* (see Figure 3). Although some of these solutions and research challenges have been tackled previously, they have not been combined into such an end-to-end intelligent data exploration system, which in turn opens up new research challenges.

INODE's major components are as follows: (1) *Data Modeling and Linking* enables integration of both structured and unstructured data. (2) *Integrated Query Processing* enables efficient query processing across federated databases leveraging ontologies. (3) *Data Access and Exploration* enables guided data exploration in various modalities such as by natural language, by recommendation, by example or visually. We refer to these as *operators*.[3] Even though INODE is an integrated system, each of the components can also be used independently. The system is targeted for domain scientists as well as the general public.

3.1 Data Modeling and Linking

This component links loosely coupled collections of data sources such as relational databases, graph databases or text documents based on the well-established *ontology-based data access* (OBDA) paradigm [26]. OBDA uses a global ontology (knowledge graph) to *model* the domain of interest and provides a conceptual representation of the information in the data sources. The sources are linked to elements in the global ontology through declarative *GAV mappings* [15]. It is well-known that designing OBDA mappings manually is a time-consuming and error-prone task. The Data Modeling and Linking component of INODE aims at automatizing this task by providing two mechanisms: *data-driven* and *task-driven mapping* generation.

Data-driven Mapping Generation. This mechanism deals with linking novel data sources to the system. The idea is to rely on *mapping patterns* that describe well-assessed and sound schema-transformation rules usually applied in the design process of relational databases. By analyzing (driven by the patterns) the data sources, it is possible to automatically derive a so-called *putative ontology* [20] describing both the explicit entities and relationships constituting the schema and the implicit ones inferrable from the data. From the mapping patterns, one can also automatically derive mappings that link the data sources to the putative ontology.

Task-driven Mapping Generation. This mechanism is applied whenever a task or a query is formulated that uses specific *target* ontology elements that are not yet aligned with the putative ontology. In such scenario, the semantics of the query are used to automatically generate mappings to align the target ontology with the putative ontology.

Knowledge Graph (KG) Generation. This service transforms unstructured information hidden in large quantities of text (e.g. repositories of scientific papers) to an exploitable structured representation through an NLP pipeline. INODE follows an Open Information Extraction (OIE) approach to convert each sentence of the corpus into a set of relational triples, where each triple consists of a subject, an object, and a predicate (relationship) linking them. We leverage a number of preprocessing techniques, including co-reference resolution and extractive summarization to improve the quality of the extracted relational triples. We combine different OIE methods (rule-based, analytics-based and learning-based) to achieve both high precision and high recall [19, 22]. The relational triples are further linked with domain-specific ontology concepts before being integrated into the knowledge graph.

Note that all tasks of the *Data Modelling and Linking* component are executed offline and hence do not require interactivity.

3.2 Integrated Query Processing

This component is responsible for the execution of queries using Ontop [28], a the state-of-the-art OBDA system. Ontop allows the users to formulate queries in terms of *concepts* and *properties* of

[3] A prototype implementation of the major system components can be found at: `http://www.inode-project.eu/opendatadialog/`

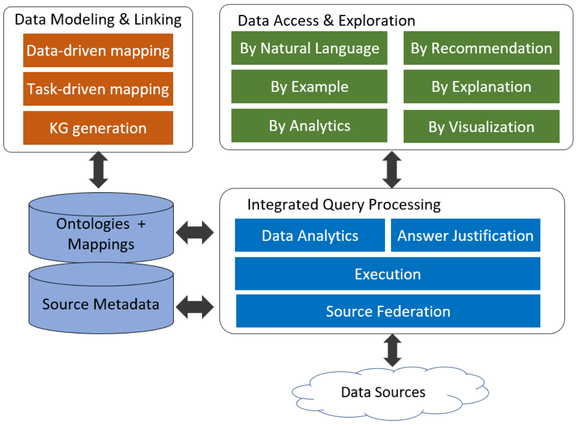


Figure 3: Major components of the INODE architecture.

their domain of expertise (represented in *knowledge graphs*), rather than in terms of table and attribute names used in the actual data sources. Hence, users do not have to be aware of the specific storage details of the underlying data sources in order to satisfy their information needs.

Query Execution. This service provides on-the-fly *reformulation* of SPARQL queries over the domain ontology to SQL queries over the data sources. An approach based on reformulation has the advantage that the data available in the data sources does not need to be duplicated in the query processing system, but can be kept in the data sources as-is. This means that the Query Execution service is guaranteed even in the common scenario where the user does not own the data nor does have the right to copy them. To produce reformulations that can efficiently be executed over the data, in INODE we use optimization techniques such as *self-join elimination for denormalized data* [28] and *optimizations of left-joins* [27].

Source Federation. The Source Federation service deals with distributing the processing of queries over the available data sources. INODE provides *seamless federation* over the SQL data sources.

In seamless federation, users send queries against a unified view of the remote endpoints without the need to be aware of the actual vocabularies used in the federated endpoints. The challenge is to automatically detect to which sources which components of the query need to be dispatched, to collect the retrieved results, and to combine them into a coherent answer. We address this challenge by relying on the knowledge about the sources encoded in the OBDA mappings. Note that in a seamless setting, the end-user interacts with the endpoint as usual, and remains unaware of whether the system will perform a federated query to retrieve the answers. Given that efficiency is a crucial requirement, in this, mostly interactive setting, our approach requires a dedicated cost-model able to minimize the number of distributed joins over the federation layer, in order to favor more efficient joins at the level of the sources.

Data Analytics. The data analytics service exploits novel and efficient query reformulation and optimization techniques [28] to compute complex analytical functions. Such techniques are based on algebraic transformations of the SPARQL algebra tree, rather than on Datalog transformations as traditionally done in the OBDA literature. This shift of paradigm allows for an efficient implementation of analytical functions such as SPARQL aggregates. It is worth noting that INODE, through Ontop, provides the first open-source reformulation-based system able to support SPARQL aggregates.

3.3 Data Access and Exploration Operators

We describe the set of operators currently available individually within INODE.

Exploration by Natural Language. For translating a natural language question into SQL or SPARQL, INODE uses *pattern-based*, *graph-based* and *neural network-based approaches*. For translating from NL to SQL, INODE extends the pattern-based system SODA [3] with NLP techniques such as lemmatization, stemming and POS tagging to allow both key word search queries as well as full natural language questions. In addition, we use Bio-SODA [21], a graph-based system to enable NL questions over RDF graph databases.

Finally, INODE integrates the *neural network-based approach* ValueNet, which leverages transformer architectures to translate NL to SQL [4]. The ultimate goal of INODE is to combine all these techniques into an *intelligent hybrid approach* that improves on the errors of each of the individual systems.

Exploration by Explanation. One of the biggest hurdles in today's exploration systems is that the system provides no explanations of the results or system choices. Nor does the system trigger input from the user, for example, by asking the user to provide more information. In INODE, we enable a conversational setting, where the system can (a) ask for clarifications and (b) explain results in natural language. This interaction assumes that the system is capable of analyzing and understanding user requests and generating its answers or questions in natural language.

One approach used in INODE builds on *Template-based Synthesis* [12]. This approach considers the database schema as a graph and a query as a subgraph. We use templates that tell us how to compose sentences as we traverse the graph and we use different traversal strategies that generate query descriptions as phrases in natural language. Furthermore, to generate NL descriptions that use the vocabulary of a particular database, INODE enriches its vocabulary by *leveraging ontologies* built by the Data Modeling and Linking components. To further improve INODE's explanation capabilities, we are working on an approach to automatically learn templates, which is especially critical for databases with no descriptive metadata, such as SDSS. Essentially, we are using *neural-based methods* to translate from SQL or SPARQL to natural language.

Exploration by Example and by Analytics. By example is a powerful operator that encapsulates multiple semantics. It takes a set of examples, such as galaxies or patients, and explores its different facets, filters them, finds similar/dissimilar sets, finds overlapping sets, joins them with other sets, finds a superset, etc. Additionally, by-example operators can be combined with by-analytics to find sets that are similar/dissimilar wrt some value distributions.

By-example and by-analytics operators can be represented in the *Region Connection Calculus* (RCC) [16] and are, in their general form, computationally challenging. For instance, by-subset is akin to solving a set cover problem, which has been extensively studied [5]. Similarly, by-join requires to have appropriate indices. In INODE, we adopt two approaches. One is based on a *r*elational backend in which individual operators are translated into SQL. The other one is an *i*n-memory Python implementation that relies on pre-computing and indexing sets.

Exploration by Recommendation. In a mixed-initiative setting, the system actively guides the user in what possible actions to perform or data to look at next. In INODE, we are interested in recommendations in both *cold-start* (where the user has not given any input) and *warm-start* settings (where the user has asked one or more queries but may not know what to do next). In the former case, the goal is to show a set of example or starter queries that the users could use to get some initial answers from the dataset (e.g. [9]). In the latter case, the system can leverage the user's interactions (queries) to show possible next queries (e.g., [8]).

A big differentiator is the availability of query logs. In case no query logs are available, the system should still provide recommendations. In INODE we are addressing the recommendation problem from different angles, i.e., generating recommendations: (a) based on *data analysis* [7] (b) by *NL-based processing and query augmentation techniques* leveraging knowledge bases (c) by *user log analysis.*

Exploration by Visualization. In information retrieval, search queries result in a list of candidates ranked by their matching score [18]. This also holds true for INODE, as most exploration operators generate multiple potential answers. However, results are not individual items such as documents, but data sets (i.e. sets of items) and have to be communicated to the user differently to support their goals. Not only do users have to decide, which data set contains the answer they are looking for, but also to compare the results, to assess redundancies, discrepancies and other surprising or interesting differences in order to draw hints on how to continue the exploration. The goals of the *by-visualization* data access and exploration interface are two-fold: (1) Enable "explorers" to understand, compare and decide based on the provided results and (2) enable them to interact with the results by enabling indirect query manipulation, identifying and highlighting parts that are of interest for further analysis and guiding them towards interesting regions [24].

Our processes for user requirements elicitation confirms our goals stated above and is based on the *User Centered Design* standard [10]. In addition to that, users emphasized the importance to compare differences as well as similarities of queries and results. As a baseline, we enabled the visualization of multiple tables with direct manipulation capabilities and currently work on an *overview visualization* that spans the result data space.

4. CONCLUSIONS

A full-fledged data exploration system should *learn about data sources*, *learn about users and queries*, and leverage this knowledge to facilitate and *guide users.* All these challenges constitute new opportunities for ML research to contribute to DM which are elaborated in the extended version of this paper [2].

5. ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 863410.

REFERENCES

[1] K. Affolter, K. Stockinger, and A. Bernstein. A comparative survey of recent natural language interfaces for databases. *The VLDB Journal*, 28(5):793–819, 2019.

[2] S. Amer-Yahia, G. Koutrika, F. Bastian, T. Belmpas, M. Braschler, U. Brunner, D. Calvanese, M. Fabricius, O. Gkini, C. Kosten, D. Lanti, A. Litke, H. Lücke-Tieke, F. A. Massucci, T. M. de Farias, A. Mosca, F. Multari, N. Papadakis, D. Papadopoulos, Y. Patil, A. Personnaz, G. Rull, A. C. Sima, E. Smith, D. Skoutas, S. Subramanian, G. Xiao, and K. Stockinger. INODE: building an end-to-end data exploration system in practice [extended vision]. *CoRR*, abs/2104.04194, 2021.

[3] L. Blunschi, C. Jossen, D. Kossmann, M. Mori, and K. Stockinger. Soda: Generating sql for business users. *PVLDB*, 2012.

[4] U. Brunner and K. Stockinger. Valuenet: A natural language-to-sql system that learns from database information. *ICDE*, 2021.

[5] G. Cormode, H. J. Karloff, and A. Wirth. Set cover algorithms for very large datasets. In *CIKM*, 2010.

[6] K. Dimitriadou, O. Papaemmanouil, and Y. Diao. AIDE: an active learning-based approach for interactive data exploration. *IEEE Trans. Knowl. Data Eng.*, 28(11):2842–2856, 2016.

[7] A. Glenis, Y. Stavrakas, and G. Koutrika. Pyexplore: Clustering-based sql query recommendations. In *under submission*, 2020.

[8] M. L. Guilly, J. Petit, and V. Scuturici. SQL query completion for data exploration. *CoRR*, abs/1802.02872, 2018.

[9] B. Howe, G. Cole, N. Khoussainova, and L. Battle. Automatic example queries for ad hoc databases. In *SIGMOD*, 2011.

[10] International Organization for Standardization. ISO 9241-210:2019 - Ergonomics of Human-System Interaction — Part 210: Human-Centred Design for Interactive Systems, 2019.

[11] R. J. L. John, N. Potti, and J. M. Patel. Ava: From data to insights through conversations. In *CIDR*, 2017.

[12] A. Kokkalis, P. Vagenas, A. Zervakis, A. Simitsis, G. Koutrika, and Y. E. Ioannidis. Logos: a system for translating queries into narratives. In *SIGMOD*, 2012.

[13] T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis. The case for learned index structures. In *SIGMOD*, 2018.

[14] A. Kristo, K. Vaidya, U. Çetintemel, S. Misra, and T. Kraska. The case for a learned sorting algorithm. In *SIGMOD*, 2020.

[15] M. Lenzerini. Data integration: A theoretical perspective. In *PODS*, 2002.

[16] S. Li and M. Ying. Region connection calculus: Its models and composition table. *Artificial Intelligence*, 145(1):121 – 146, 2003.

[17] J. Liu, Z. Zolaktaf, R. Pottinger, and M. Milani. Improvement of SQL recommendation on scientific database. In *SSDBM*, 2019.

[18] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[19] D. Papadopoulos, N. Papadakis, and A. Litke. A methodology for open information extraction and representation from large scientific corpora: The cord-19 data exploration use case. *Applied Sciences*, 10(16), 2020.

[20] J. F. Sequeda and D. P. Miranker. Ultrawrap Mapper: A semi-automatic relational database to RDF (RDB2RDF) mapping tool. In *Proc. of the 14th Int. Semantic Web Conf., Posters & Demonstrations Track (ISWC)*, 2015.

[21] A. C. Sima, T. Mendes de Farias, E. Zbinden, M. Anisimova, M. Gil, H. Stockinger, K. Stockinger, M. Robinson-Rechavi, and C. Dessimoz. Enabling semantic queries across federated bioinformatics databases. *Database*, 2019, 2019.

[22] E. Smith, D. Papadopoulos, M. Braschler, and K. Stockinger. Lillie: Information extraction and database integration using linguistics and learning-based algorithms. *Information Systems*, 2021.

[23] E. R. Sparks, S. Venkataraman, T. Kaftan, M. J. Franklin, and B. Recht. Keystoneml: Optimizing pipelines for large-scale advanced analytics. In *ICDE*, 2017.

[24] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer graphics forum*, volume 33, pages 401–410. Wiley Online Library, 2014.

[25] I. Stoica. Systems and ML: when the sum is greater than its parts. In *SIGMOD*, 2020.

[26] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, and M. Zakharyaschev. Ontology-based data access: A survey. In *IJCAI*, 2018.

[27] G. Xiao, R. Kontchakov, B. Cogrel, D. Calvanese, and E. Botoeva. Efficient handling of SPARQL optional for OBDA. In *ISWC*, 2018.

[28] G. Xiao, D. Lanti, R. Kontchakov, S. Komla-Ebri, E. Güzel-Kalayci, L. Ding, J. Corman, B. Cogrel, D. Calvanese, and E. Botoeva. The virtual knowledge graph system ontop. In *ISWC*, 2020.