



HAL
open science

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

Sihem Amer-Yahia, Georgia Koutrika, Martin Braschler, Diego Calvanese, Davide Lanti, Hendrik Lücke-Tieke, Alessandro Mosca, Tarcisio Mendes de Farias, Dimitris Papadopoulos, Yogendra Patil, et al.

► **To cite this version:**

Sihem Amer-Yahia, Georgia Koutrika, Martin Braschler, Diego Calvanese, Davide Lanti, et al.. INODE: Building an End-to-End Data Exploration System in Practice Authors' Copy. 2021. hal-03379639

HAL Id: hal-03379639

<https://hal.science/hal-03379639v1>

Preprint submitted on 15 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Authors' Copy

Sihem Amer-Yahia², Georgia Koutrika¹, Martin Braschler⁹, Diego Calvanese⁸, Davide Lanti⁸, Hendrik Lücke-Tieke³, Alessandro Mosca⁸, Tarcisio Mendes de Farias⁷, Dimitris Papadopoulos⁴, Yogendra Patil², Guillem Rull⁶, Ellery Smith⁹, Dimitrios Skoutas¹, Srividya Subramanian⁵, Kurt Stockinger⁹

(1) Athena Research Center, Greece (2) CNRS, University Grenoble Alpes, France (3) Fraunhofer IGD, Germany (4) Infil, Greece (5) Max Planck Institute, Germany (6) SIRIS Academic, Spain (7) SIB Swiss Institute of Bioinformatics, Switzerland (8) Free University of Bozen-Bolzano, Italy (9) ZHAW 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 [33]. Conversely, existing work that leverages ML for DM [35] is nascent and covers the use of ML for query optimization [18] or for database indexing [17]. 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*

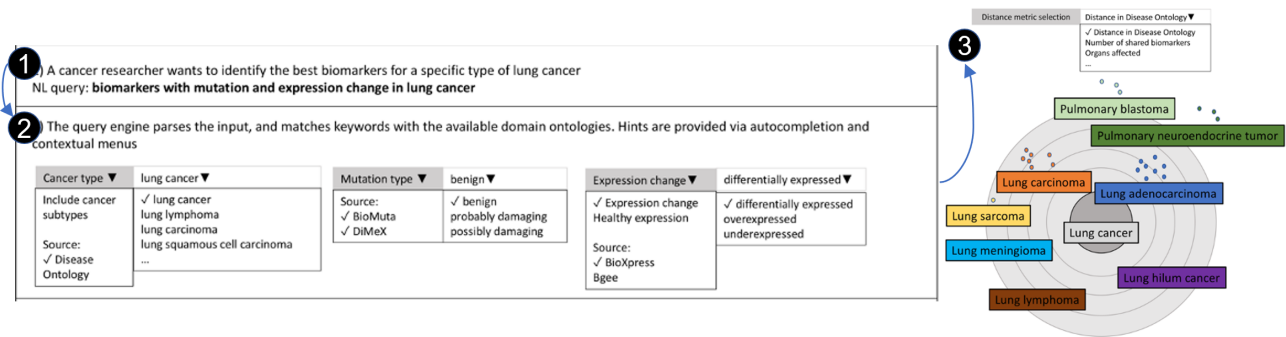


Figure 1: Natural language query interface with user assistance. Step 1: User enters a natural language query. Step 2: System parses query and matches keywords against the available ontology to enable term disambiguation; the user iterates the process. Step 3: System visualizes various cancer types that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

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 [2], SQL-to-NL [16] or both [14] (see a summary in [1]). Recommendation strategies can be leveraged to guide users [21]. 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 INODE¹, 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 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,

¹<http://www.inode-project.eu/>

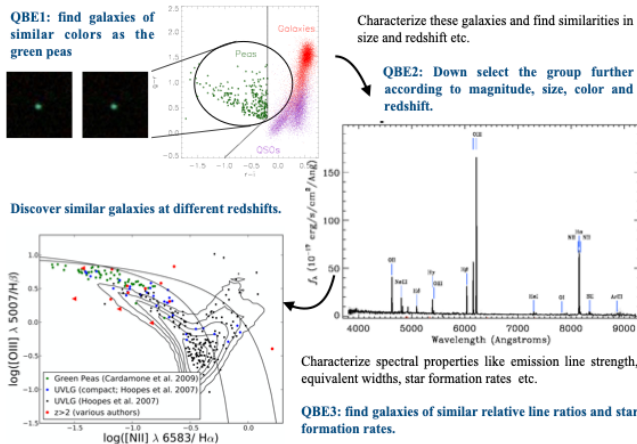


Figure 2: Exploring astrophysics data.

explores astronomical objects in SDSS, a large sky survey database². 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.

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*

²<https://www.sdss.org/>

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. A detailed description of each component is given in the component release⁴. 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 [37]. 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 *GAU mappings* [19]. 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* [30] describing both the explicit entities and relationships constituting the schema and the implicit ones inferable 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 [25]. The relational triples are further linked with domain-specific ontology concepts before being integrated into the knowledge graph.

³A prototype implementation of the major system components can be found at: <http://www.inode-project.eu/pendatadialog/>

⁴The INODE components as of September 2021 <http://www.inode-project.eu/blog/second-component-release/>

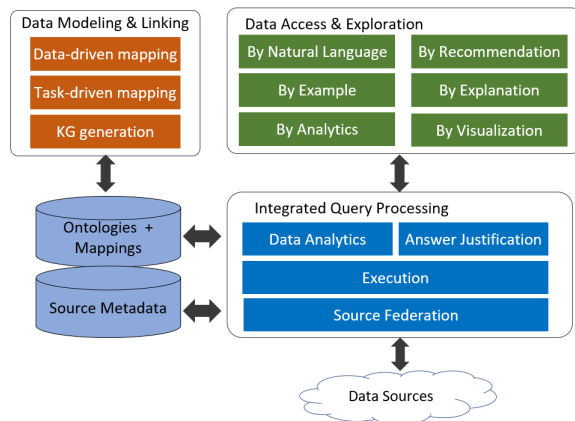


Figure 3: Major components of the INODE architecture.

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 [39], a the state-of-the-art OBDA system. Ontop allows the users to formulate queries in terms of *concepts* and *properties* of 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* [39] and *optimizations of left-joins* [38].

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 [39] 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 [2] 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 [32], 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* [16]. 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) [20] 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 relational backend in which individual operators are translated into SQL. The other one is an in-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. [12]). In the latter case, the system can leverage the user's interactions (queries) to show possible next queries (e.g., [11]).

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* [9] (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 [22]. 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 [34].

Our processes for user requirements elicitation confirms our goals stated above and is based on the *User Centered Design* standard [13]. 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 OPEN RESEARCH CHALLENGES BEYOND INODE

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.

4.1 Learning about Data Sources

There are several data integration challenges that can be tackled with ML. The first one is to *automate data integration* [10], in the spirit of existing work that improves entity matching using transformer architectures [3]. The second challenge is to *automatically generate knowledge graphs*, in the spirit of neural-network systems such as Snorkel [28]. Another powerful concept to automate knowledge graph construction is to combine user dialogs with graph construction [24]. The idea is to augment the knowledge graphs by learning concepts that are commonly queried but do not exist in the graph. In summary, there has been a large amount of research on automatic knowledge base construction. However, the combination of knowledge base construction with natural language query processing has been largely untapped.

4.2 Understanding Users and Queries

Understanding user interests and expertise is a vital component for enabling intelligent data exploration. For instance, the general public interested in black holes has different expectations from an experienced astronomer with a vast knowledge of astrophysics. The challenge is to avoid overwhelming a novice user while providing interesting and relevant information to an expert user.

The system should constantly improve its behavior by learning and adapting to the user from task to task. Our operators are a great opportunity to learn and adapt to users, as they provide the ability to choose between utility and novelty, two dimensions that have not been explored together in the past. Additionally, they enable collecting user feedback at the level of individual operators and of a DEP. While ML methods for learning user profiles exist in the context of individual systems for web search or recommendations, they have not been studied before in the context of determining which operator caters for which user in the next step. A simple start is to use regression methods to determine the weight of utility and novelty when exploring data.

[The state-of-the-art NL-to-SQL systems use supervised ML \[4, 15\]](#). A new opportunity here is to train a neural network for sequence-to-sequence prediction [4, 36] for translating from NL-to-SQL and vice versa. The key research challenge is how to use the feedback provided by users to disambiguate queries and feed the gained knowledge back into ML models to improve learning with semantic information for building ML models to tackle disambiguation and context modeling [31].

4.3 Generating Data Exploration Pipelines

Understanding queries and users helps to provide guidance in generating DEPs. This challenge can be approached in different ways depending on the [availability of logs and on the user's expertise and willingness to provide feedback](#). DEPs could be given either [partially or fully](#). In both cases, they are either provided by the user

or learned. In a scenario where a DEP is given (see example in Section 2), the problem could be cast as finding the right parameters for each query in the DEP. In a scenario where the user is providing the next query, it could be seen as a query completion problem. In a scenario where the user does not write exploration queries and only provides feedback on results, it could be seen as the problem of learning the user’s DEP. All these cases result in partially-guided or fully-guided exploration.

Furthermore, since DEPs bring together several data access modalities, which may be initiated by the user (e.g. a user query) or by the system (recommendations or explanations), the system needs to learn how to use its options to help the user in meaningful and unobtrusive ways. While there has been work on each of these capabilities individually (e.g., recommendations or query explanations), these efforts only focus on small parts of the problem lacking a holistic understanding of the behavior and dynamics of a multi-aspect system. **ML approaches such as Active Learning (AL) and Reinforcement Learning (RL) appear to be relevant to the question of training models that represent DEPs.**

Partial Guidance with AL. AL is claimed to be superior to faceted search when the goal is to help users formulate queries. Systems like AIDE [6] and REQUEST [8] assist users in constructing accurate exploratory queries, while at the same time minimizing the number of sample records presented to them for labeling. Both systems rely on training a decision tree classifier to build a model that classifies unlabeled records. A bigger challenge is to leverage AL in generating and refining queries that go beyond SQL predicates.

Full-Guidance with RL. In exploratory data analysis, logs are rarely available making RL and Deep RL [23] the methods of choice for training DEPs based on a simulated agent experience [7, 29]. In [7], a Deep RL architecture is used for generating notebooks that show diverse aspects of a data set in a coherent narrative. In [29], we used RL to generate an end-to-end exploration policy to find a set of users in a collection of groups. Both frameworks accept a wide class of exploration actions and do not need to gather exploration logs. In [26, 27], we developed DORA THE EXPLORER, a first prototype that addresses the problem of generating DEPs by leveraging Deep Reinforcement Learning. An open question is the applicability of this framework to specific data sets and the transferability of learned policies across data sets.

5 ACKNOWLEDGEMENTS

This project has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 863410. We also thank all the remaining members of the INODE Team: <http://www.inode-project.eu/consortium/>

REFERENCES

- [1] Katrin Affolter, Kurt Stockinger, and Abraham Bernstein. 2019. A comparative survey of recent natural language interfaces for databases. *The VLDB Journal* 28, 5 (2019), 793–819.
- [2] Lukas Blunschi, Claudio Jossen, Donald Kossmann, Magdalini Mori, and Kurt Stockinger. 2012. Soda: Generating Sql for business users. *PVLDB* (2012).
- [3] Ursin Brunner and Kurt Stockinger. 2020. Entity matching with transformer architectures—a step forward in data integration. In *EDBT*.
- [4] Ursin Brunner and Kurt Stockinger. 2021. ValueNet: A Natural Language-to-SQL System that Learns from Database Information. *ICDE* (2021).
- [5] Graham Cormode, Howard J. Karloff, and Anthony Wirth. 2010. Set cover algorithms for very large datasets. In *CIKM*.
- [6] Kyriaki Dimitriadou, Olga Papaemmanouil, and Yanlei Diao. 2016. AIDE: An Active Learning-Based Approach for Interactive Data Exploration. *IEEE Trans. Knowl. Data Eng.* 28, 11 (2016), 2842–2856.
- [7] Ori Bar El, Tova Milo, and Amit Somech. 2020. Automatically Generating Data Exploration Sessions Using Deep Reinforcement Learning. In *SIGMOD*.
- [8] Xiaoyu Ge, Yanbing Xue, Zhipeng Luo, Mohamed A. Sharaf, and Panos K. Chrysanthis. 2016. REQUEST: A scalable framework for interactive construction of exploratory queries. In *BigData*.
- [9] Apostolis Glenis, Yannis Stavrakas, and Georgia Koutrika. 2020. PyExplore: Clustering-based SQL query recommendations. In *under submission*.
- [10] Behzad Golshan, Alon Halevy, George Mihaila, and Wang-Chiew Tan. 2017. Data integration: After the teenage years. In *SIGMOD*.
- [11] Marie Le Guilly, Jean-Marc Petit, and Vasile-Marian Scuturici. 2018. SQL Query Completion for Data Exploration. *CoRR abs/1802.02872* (2018).
- [12] Bill Howe, Garrett Cole, Nodira Khoussainova, and Leilani Battle. 2011. Automatic example queries for ad hoc databases. In *SIGMOD*.
- [13] International Organization for Standardization. 2019. ISO 9241-210:2019 - Ergonomics of Human-System Interaction – Part 210: Human-Centred Design for Interactive Systems. , 33 pages. <https://www.iso.org/standard/77520.html>
- [14] Rogers Jeffrey Leo John, Navneet Potti, and Jignesh M. Patel. 2017. Ava: From Data to Insights Through Conversations. In *CIDR*.
- [15] Hyeonji Kim, Byeong-Hoon So, Wook-Shin Han, and Hongrae Lee. 2020. Natural language to SQL: Where are we today? *PVLDB* (2020).
- [16] Andreas Kokkalis, Panagiotis Vagenas, Alexandros Zervakis, Alkis Simitis, Georgia Koutrika, and Yannis E. Ioannidis. 2012. Logos: a system for translating queries into narratives. In *SIGMOD*.
- [17] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. 2018. The Case for Learned Index Structures. In *SIGMOD*.
- [18] Ani Kristo, Kapil Vaidya, Ugur Çetintemel, Sanchit Misra, and Tim Kraska. 2020. The Case for a Learned Sorting Algorithm. In *SIGMOD*.
- [19] Maurizio Lenzerini. 2002. Data Integration: A Theoretical Perspective. In *PODS*.
- [20] Sanjiang Li and Mingsheng Ying. 2003. Region Connection Calculus: Its models and composition table. *Artificial Intelligence* 145, 1 (2003), 121 – 146.
- [21] Jianjun Liu, Zainab Zolaktaf, Rachel Pottinger, and Mostafa Milani. 2019. Improvement of SQL Recommendation on Scientific Database. In *SDDBM*.
- [22] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. [n.d.]. *Introduction to Information Retrieval*. Cambridge University Press.
- [23] Tova Milo and Amit Somech. 2020. Automating Exploratory Data Analysis via Machine Learning: An Overview. In *SIGMOD*.
- [24] Dat Ba Nguyen, Abdalghani Abujabal, et al. 2017. Query-driven on-the-fly knowledge base construction. *PVLDB* (2017).
- [25] Dimitris Papadopoulos, Nikolaos Papadakis, and Antonis Litke. 2020. A Methodology for Open Information Extraction and Representation from Large Scientific Corpora: The CORD-19 Data Exploration Use Case. *Applied Sciences* 10, 16 (2020).
- [26] Aurélien Personnaz, Sihem Amer-Yahia, Laure Berti-Equille, Maximilian Fabricius, and Srividya Subramanian. 2021. Balancing Familiarity and Curiosity in Data Exploration with Deep Reinforcement Learning. In *aiDM’21 @ SIGMOD*.
- [27] Aurélien Personnaz, Sihem Amer-Yahia, Laure Berti-Equille, Maximilian Fabricius, and Srividya Subramanian. 2021. DORA The Explorer: Exploring Data with Interactive Deep Reinforcement Learning (prototype demonstration). In *CIKM*.
- [28] Alexander J Ratner, Stephen H Bach, Henry R Ehrenberg, and Chris Ré. 2017. Snorkel: Fast training set generation for information extraction. In *SIGMOD*.
- [29] Mariia Seleznova, Behrooz Omidvar-Tehrani, Sihem Amer-Yahia, and Eric Simon. 2020. Guided Exploration of User Groups. *PVLDB* (2020).
- [30] Juan F. Sequeda and Daniel P. Miranker. 2015. 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) (CEUR Workshop Proceedings, http://ceur-ws.org/)*, Vol. 1486.
- [31] Saeedeh Shekarpour, Kemele M Endris, Ashwini Jaya Kumar, Denis Lukovnikov, Kuldeep Singh, Harsh Thakkar, and Christoph Lange. 2016. Question answering on linked data: Challenges and future directions. In *Proceedings of the 25th International Conference Companion on World Wide Web*. 693–698.
- [32] Ana Claudia Sima, Tarcisio Mendes de Farias, Erich Zbinden, Maria Anisimova, Manuel Gil, Heinz Stockinger, Kurt Stockinger, Marc Robinson-Rechavi, and Christophe Dessimoz. 2019. Enabling semantic queries across federated bioinformatics databases. *Database* 2019 (2019).
- [33] Evan R. Sparks, Shivaram Venkataraman, Tomer Kaftan, Michael J. Franklin, and Benjamin Recht. 2017. KeystoneML: Optimizing Pipelines for Large-Scale Advanced Analytics. In *ICDE*.
- [34] Martin Steiger, Jürgen Bernard, Sebastian Mittelstädt, Hendrik Lücke-Tieke, Daniel Keim, Thorsten May, and Jörn Kohlhammer. 2014. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer graphics forum*, Vol. 33. Wiley Online Library, 401–410.
- [35] Ion Stoica. 2020. Systems and ML: When the Sum is Greater than Its Parts. In *SIGMOD*.
- [36] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*.

- [37] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyashev. 2018. Ontology-Based Data Access: A Survey. In *IJCAI*.
- [38] Guohui Xiao, Roman Kontchakov, Benjamin Cogrel, Diego Calvanese, and Elena Botoeva. 2018. Efficient Handling of SPARQL Optional for OBDA. In *ISWC*.
- [39] Guohui Xiao, Davide Lanti, Roman Kontchakov, Sarah Komla-Ebri, Elem Güzel-Kalayci, Linfang Ding, Julien Corman, Benjamin Cogrel, Diego Calvanese, and Elena Botoeva. 2020. The Virtual Knowledge Graph System Ontop. In *ISWC*.