Table Discovery in Data Lakes: State-of-the-art and Future Directions

Grace Fan
Northeastern University
fan.gr@northeastern.edu

Jin Wang
Megagon Labs
jin@megagon.ai

Yuliang Li
Megagon Labs
yuliang@megagon.ai

Renée J. Miller
Northeastern University
miller@northeastern.edu

ABSTRACT

Data discovery refers to a set of tasks that enable users and downstream applications to explore and gain insights from massive collections of data sources such as data lakes. In this tutorial, we will provide a comprehensive overview of the most recent table discovery techniques developed by the data management community. We will cover table understanding tasks such as domain discovery, table annotation, and table representation learning which help data lake systems capture semantics of tables. We will also cover techniques enabling various query-driven discovery and table exploration tasks, as well as how table discovery can support key data science applications such as machine learning and knowledge base construction. Finally, we will discuss future research directions on developing new table discovery paradigms by combining structured knowledge and dense table representations, as well as improving the efficiency of discovery using state-of-the-art indexing techniques, and more.

CCS CONCEPTS

• **Information systems → Top-k retrieval in databases**; *Mediators and data integration*; *Database design and models.*

KEYWORDS

Dataset Discovery, Data Lake, Data Integration, Unionable Tables

ACM Reference Format:
Grace Fan, Jin Wang, Yuliang Li, and Renée J. Miller. 2023. Table Discovery in Data Lakes: State-of-the-art and Future Directions. In *Companion of the 2023 International Conference on Management of Data (SIGMOD-Companion '23), June 18–23, 2023, Seattle, WA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3555041.3589409

1 INTRODUCTION

We have witnessed in the last decades a drastic growth in the number of open and shared datasets coming from governments, academic institutes, and private companies. These massive collections of structured or unstructured datasets, which we refer to as *data lakes*, open up new opportunities for innovation, economic growth, and social benefits [39]. As of 2022, the size of US open data (data.gov) has reached a total of 335k datasets contributing to $3 trillion of the US economy [9]. WebDataCommons [30] has also made 233M tables extracted from the open web publicly available. Data lakes store a wide variety of open-domain knowledge from sources such as Wikipedia, news articles, and other online sources, as well as data that is private to an organization or purchased from data brokers. However, while data lakes are invaluable data sources, their large size and complexity can make it difficult for users to find and access the specific data that is required by specific downstream applications. To address this issue, the data management community has been building *data discovery systems* [3, 6, 18, 20, 34, 40, 48], with table discovery from data lakes as the primary application. Data discovery systems help users quickly and easily explore and analyze large datasets, enabling them to gain insights and knowledge that can support decision making and other activities.

This tutorial aims at providing a comprehensive overview of the most recent developments from the database community that enable the aforementioned data discovery methodologies and systems. While there are some previous studies about discovery from other data formats, such as images and JSON [36, 55], we focus on the discovery from tabular data, which is a primary data format in data lakes. We will organize our tutorial by taking an architectural view of table discovery systems depicted in Figure 1. The architecture highlights key system components (in green) and sub-tasks (in blue) that are essential building blocks of an end-to-end data discovery pipeline from raw data lake tables to end-users and downstream applications.

Firstly, we will have an overview of several table understanding tasks. Given an input data lake, table discovery systems typically enrich the raw data lake tables via automatically annotating and indexing techniques to enable better understanding of table semantics. Annotations such as semantic column types can be further used to improve the performance of downstream applications (e.g., table search). We will focus on the tasks of (1) *table annotation* [34] for generating useful meta-data about tables, (2) *domain discovery* [33, 45] for identifying novel domains beyond standard DB data types, (3) machine learning techniques for learning *embeddings* [14, 50] of data items in tables (including columns, rows, and entire tables).

SIGMOD-Companion '23, June 18–23, 2023, Seattle, WA, USA

ACM ISBN 978-1-4503-9507-6/23/06…$15.00
https://doi.org/10.1145/3555041.3589409

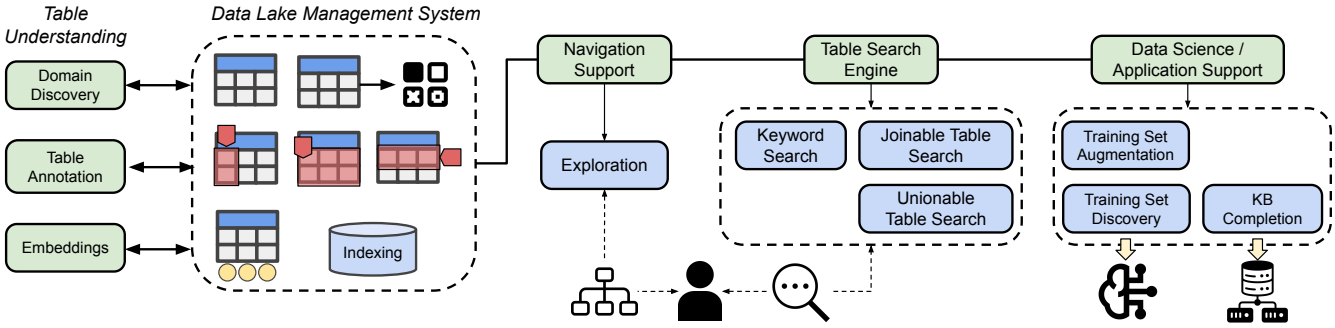

Figure 1: Overview of table discovery in data lakes.

The table discovery system serves the data lake tables to the end-users (e.g., data scientists) by enabling them to query and explore tables that are relevant to their downstream analytic tasks. This is typically achieved via *query driven discovery* [43]: given a user query, retrieve the top-k relevant tables from the data lake. Alternatively, some discovery systems support *data navigation* allowing a user to navigate through the tables in a data lake using a (discovered) organization for the tables [42] or by exploring individual tables followed by tables related through a foreign key or other relationships [18]. And query and navigation can be combined to create organizations over the set of tables retrieved by search [46] in an online manner.

For query driven discovery, we will survey the classic query types includingkeyword search or meta-data search, which often use table schema including attribute names [4, 47, 54] as supported by popular systems such as Google dataset search [3, 21]. Importantly, rather than keywords or table schema, an emerging trend in table discovery systems is to support querying data lakes using table values (e.g., columns or an entire table). Previous studies have identified two types of important search queries: joinable table search [12, 15, 16, 48, 59, 60] and unionable table search [2, 12, 17, 27, 44]. Joinable table search aims at augmenting a query table with additional attributes, while unionable table search is motivated by the need to find tables with new tuples to extend a query table. We will be covering the classic schema-based and value-based approaches, as well as the latest approaches based on knowledge graphs [27] and pre-trained language models [17] that achieve state-of-the-art performance.

We will also cover an emerging trend of performing data discovery to support a variety of data science tasks. One typical example is the creation and augmentation of training datasets. High-quality training datasets are often times a prerequisite for the success of ML models. Data lakes provide a unique opportunity for harvesting training data. Among those techniques, we will primarily introduce (1) data augmentation [8] that aims at finding new features to improve the ML models' performance and (2) training set discovery/construction [58] which focuses on constructing training data from a fixed set of tables via learning representations.

Finally, we will present challenges and future research directions for table discovery. We will cover four main aspects. First, for table understanding which is core to table discovery, we will discuss the possibility of combining two lines of already successful approaches: *knowledge* based approaches [27, 53] and *machine learning* based approaches [14, 17, 23]. We will challenge the community to find synergies between the two approaches by bringing together both rich structured knowledge from KB's and highly effective dense representations from ML. Second, we will further discuss the challenge of boosting the scalability of discovery systems, especially the issues related to indexing. We will elaborate on why indexing is still a challenging unsolved problem for data lakes and how some recently developed techniques on data structures [25], sketching [48], and vector indices [38] can potentially help. Finally, we will discuss the challenge of query-time table annotation and a potential research direction for connecting table discovery with graph mining.

2 TUTORIAL OUTLINE

In this section, we give an overview of the topics that will be covered in the tutorial. They are depicted in Figure 1 as the main components of a data discovery system (in green) and applications (in blue) that are related to table discovery.

2.1 Overview

A popular methodology for table discovery is query-driven discovery [12, 43, 60], where the user starts with a query and aims to find dataset(s) relevant to the query. The query could be either keywords or tables, while the query type could be exact match, joinable, unionable, etc. Existing studies support keyword search over the metadata of tables for users who intend to find tables relevant to a topic (Section 2.3). However, the schemas of tables in data lakes are often unreliable, either missing or incomplete often due to data being shared in primitive formats like CSV [43]. Even when metadata is available, the tables often come from a huge variety of sources meaning the metadata may be inconsistent using different vocabularies and naming conventions. Thus, two important lines of work have emerged. The first we call *Table Understanding*, which are tasks that seek to recover some of the semantics of tables. The second is data-driven table discovery based on a query table (rather than metadata), specifically joinable table search and unionable table search. Yet, data-driven table discovery still has ambiguity in the definitions, which have evolved over time. For example, early work [12] defines related tables as schema-complements and entity-complements based on a defined subject attribute that explains

the table entities. Valentine [28] uses four different notions of joinability and unionability depending on the alignment between the schema of returned tables with that of the query table. Bogatu et al. [2] simultaneously finds joinable and unionable tables by employing five different metrics. Thus, definitions of finding related tables can be very broad, leading to many variations of joinable and unionable table search, which are discussed in Sections 2.4 and 2.5, respectively. Also, there is another aspect of table discovery called navigation [46], which can return a collection of tables in a hierarchical structure that includes not only the contents but also the relationship of tables in the results.

Besides, the outcome of table discovery can benefit many related data science applications. For instance, in the problem of tabular data classification, joinable table search [15] can help extract richer features from the data lakes to enrich the original tables in the training data. And table stitching [29] has been proven rather useful in supporting knowledge base completion.

2.2 Table Understanding

An early influential table discovery system [34] makes use of a variety of table annotations to perform search. These include annotating cells with entities (from an ontology) that the cell mentions, annotating columns with ontology types, and annotating pairs of columns with ontology relationships [34]. The approach uses a supervised probabilistic graphical model to collectively learn all types of annotations. Venetis et al. [51] achieves similar goals by leveraging a proprietary ontology to label columns and binary relationships. More recent work has used labeled training data and supervised learning to label columns with a set of types. These studies start from a set of tables annotated with a fixed set of 78 types and train a machine learning model to apply on unlabeled tables. Sherlock [23] uses a feature-based method to learn column representations for semantic type detection; while Sato [56] makes further improvements by incorporating topic modeling methods (that use row context) for semantic type detection.

Domain discovery is a related problem that aims to identify sets of terms that represent instances of a semantic concept or domain from a given collection of tables. Rather than annotating columns with types, domain discovery collects all values that belong to the same domain. Existing solutions [33, 45] are unsupervised and leverage co-occurrence information to cluster domain terms. In addition Li et al. [33] select a representative as the domain or semantic type.

Recently there is a new trend of utilizing pre-trained language models like BERT to support table annotation applications [14, 50, 52]. The idea is to sequentialize tabular data items by inserting special tokens and use this representation of tables as input to perform fine-tuning over pre-trained language models. This has been done for a variety of annotation applications, including column type detection, relation extraction, cell filling, row population and schema augmentation [14]. Notably, such studies achieve state-of-the-art performance on a variety of benchmarking tasks due to the superior power of language models. We will revisit this observation when we talk about future challenges (Section 3).

2.3 Keyword Search

For users who intend to find tables about a certain topic, some data discovery systems support keyword search to satisfy this need. Keyword search was first used to find tuples within databases [10, 22, 26], but this soon developed into keyword search to find relevant tables, especially in a data lake setting. In this case, a user provides topic keywords and the data discovery system finds relevant tables where different systems have defined relevance by proposing different similarity measurements. The OCTOPUS [4] system supports keyword search over web tables using document keyword search to find pages and then computes the correlation between the keywords and each element (cell) in any table extracted from the document. OCTOPUS returns clusters of tables that share the same schema. In addition, it infers additional attributes by joining tables. In contrast, Pimplikar and Sarawagi [47] use keyword queries describing columns and identify a mapping between the columns of a table and query keywords using table headers, the text surrounding a table on a webpage, and the data within a table. To better facilitate keyword search, Google Dataset Search [3, 21] focuses on integrating and normalizing metadata of tables from various data sources. It works on cleaning, standardizing, and inferring the metadata of each dataset and the relationships among different datasets, while organizing them into a knowledge graph to facilitate retrieval. Keyword search is done only over the metadata, not the data within a table.

2.4 Joinable Table Search

Except for keyword search, many table discovery systems use the data values themselves to find tables that are related to the column values of the given query table. One problem that several approaches aim to solve is joinable table search, with the goal of finding relevant tables to join and augment the query table with additional attributes. InfoGather [54] allows augmentation of a query table with new attributes by first performing schema matching and then either augmentation by attribute names or example tuples, or discovery of related attributes to a given set of entities. Das Sarma et al. [12] defines joinable tables as schema-complements, where schema matching is conducted using attribute names, types, and values to find joinable attributes and non-matching attributes to augment with the query. Lehmberg et al. [31] conducts joinable table search on web tables with an aligned subject attributes by measuring the Jaccard similarity between the data values, metadata, and table contexts.

Although early work tended to use Jaccard similarity to measure the syntactic similarity between attributes, Jaccard has been shown to be bias attributes with smaller cardinality [1]. This is impractical in the case of data lake tables, where there is a large skew in the domain sizes. LSH Ensemble [60] studies the problem of domain search to solve the problem of joinable table search. For domain search, it uses set containment to find attributes with high overlap (containment) in values. In this way, it finds joinable tables by finding domains that highly contain the query domain. By employing a Locality Sensitive Hashing (LSH) index [13] and a data partitioning scheme that optimally minimizes false negatives, LSH ensemble is able to find joinable tables on a large scale even among attributes with skewed cardinality distributions. This line of work continues

with JOSIE [59], which uses set containment to assess the equi-joinability between columns and returns the exact top-k joinable tables.

Meanwhile, Juneau [57] supports joinable table search in Jupyter Notebooks by creating data profiles using the data values and domains, then performing schema matching. PEXESO [15] addresses the joinable table search problem using word embeddings. By encoding the columns into high-dimensional vectors and finding the cosine similarity between vectors, it finds tables that can be fuzzy joined using similarity predicates. Santos et al. [48] extends joinable table search to also find top-k tables that are joinable with the query table and contain numerical columns that are correlated with some numerical query column. To assist with the retrieval of tables that are both joinable and correlated, it uses a QCR hashing scheme that estimates the correlations between join keys and numerical values. MATE [16] allows for joins over multiple attributes by hashing each row value and aggregating them into a super key. WarpGate [11] supports semantic join discovery by using embeddings to capture semantic relationships between tables and retrieving the top-k joinable columns to a query column.

2.5 Unionable Table Search

Another data-driven table discovery problem is unionable table search, with the goal of finding tables that can be unioned with the query table to extend it with tuples. However, this problem definition is very vague, leading many systems to create their own interpretation of what it means to be unionable. For example, Das Sarma et al. [12] first defines unionable tables as entity-complements that share a subject attribute and have similar, if not the same schema as the query table. To this end, it makes use of different ontologies, including a proprietary one, and table co-occurrences. Relaxing the strong, and often unrealistic, assumption that unionable tables share schemas, Nargesian et al. [44] defines table union search based on attribute unionability, in which unionable tables have attributes that originate from the same domain. It defines three probabilistic models to measure the likelihood that attributes originate from the same domain and makes use of syntactic similarity through set overlap, semantic similarity using an ontology, and a natural language measure using word embeddings. It then aggregates the column-level scores to a table-level unionability score using bipartite graph matching. Attribute unionability is a much more expensive measure to compute than schema overlap, hence Nargesian et al. [44] use LSH indices to store column embeddings and return most related columns in sub-linear time. This approach has been extended to add in measures that include formatting similarity and attribute names [2].

SANTOS [27] defines table unionability not only based on columns, but also column-to-column relationships. By capturing binary relationships between column values, SANTOS more accurately defines the table semantics and retrieves tables that align with the intention behind the query, thus reducing false positives significantly. Using a notion of intent columns to capture the topic(s) of the tables, SANTOS finds the semantics of columns and their relationships using both an existing knowledge graph and a self-curated knowledge graph, populated with values and relationships between values from the data lake.

More recently, Starmie [17] extends the notion of capturing binary relationships in tables to making use of the entire context within a table. Following the growing success of pre-trained language models in encoding column semantics, Starmie leverages self-supervised learning to encode contextualized column representations, such that the entire table context is taken into account. Starmie then uses indices like LSH and HNSW (Hierarchical Navigable Small World) [38] to find column embeddings with high similarity to those of the query table, and aggregates them into table-level similarities. By employing a table-context approach to the unionable table search problem, Starmie has been shown to be effective in efficiently returning tables with highly-aligned semantics to the query table.

2.6 Data Lake Navigation

Navigation is an alternative way to realize table discovery. Instead of returning a flat structure such as a list of tables or attributes, navigation makes use of hierarchies or Directed Acyclic Graphs (DAG) to organize information about tables. To reach this goal, some studies [18, 19] help users discover relevant data items by navigating over a linkage graph. The links or edges in the graph indicate the relevance between datasets, tables or attributes. Another approach focuses on constructing a hierarchical structure over data lake tables that allow users to discover a table relevant to a topic of interest from the process of navigation over an "organization" of data lakes [41, 42, 46] Auctus [6] enables dataset search over tables collected from multiple open-data portals by exploring different profiling and querying and augmentation techniques.

2.7 Support Data Science Applications

The outcome of table discovery can always benefit a variety of data science applications in many fields. A typical example is the machine learning based ones. ARDA [8] augments the relational table by relational operations, such as join, projection and aggregation, to automatically involve richer features to improve machine learning applications. Leva [58] proposes a graph representation learning framework to capture inter-table information which can benefit related regression and classification applications for tabular data. It is also illustrated in recent studies [15, 17] that finding joinable tables can help improve machine learning tasks. Another import use case that can benefit from table discovery is knowledge bases. This can be realized by table stitching [35], which aims at finding pairs of tables with semantically equivalent headers. Lehmberg and Bizer [29] show that stitching union tables can boost the application of knowledge base completion via matching facts with contents from web tables.

3 CHALLENGES AND OPPORTUNITIES

Another important goal of the tutorial will be to present and examine in detail open challenges and opportunities in data discovery. Our goal will be to provoke a discussion and interest in this important area. To this end, an important role of data discovery systems is the recovery of semantics. Tables are often shared with limited or incomplete meta-data or with meta-data that is represented in a way that is inconsistent with the meta-data of other tables in the data lake. Current approaches have successfully used knowledge

bases (KB) to recover semantics as well as a variety of machine learning approaches from statistical relational models, word embeddings, and the dense representations of language models. Common wisdom says that KBs provide higher precision with possibly low coverage or recall in contrast to the higher recall of language models at the cost of some precision [53]. But this trade-off (and how to best exploit this trade-off) has not been formally studied for data discovery systems. We will examine this trade-off, how it is currently handled, and suggest avenues for study.

Many of the approaches we will survey make use of indexes to provide scalability. The study of data structures in data management that index a single (or pair) of tables is well developed. Indeed a periodic table for single table indices has been proposed [25]. But for data lakes, we require indices over millions of tables or more. Researchers have employed indices that include variants of the inverted lists for textual data [16, 37, 59], sketching indices [48], LSH indices [2, 44, 60] and graph-based indices such as HNSW [38]. However, the community's study of the properties of such indices is in its infancy compared to traditional single-table data structures. Extending current work, the tutorial will present a vision for the design space of such data structures. Notably this will include issues like adapting to the data distributions within data lakes which may vary dramatically between data lakes and which may also be dynamic in some lakes. This points to the need to develop more effective cost-based and distribution-aware access methods that optimize access based on the data distribution and points to the question of whether learned indices can be effective beyond single table data structures [24, 49].

Current data discovery systems have offline components that process the whole data lake to do a variety of annotation tasks like annotating columns with semantic types or domains. Similarly, they may create column, row, or table representations that can then be indexed to use in online components like search or navigation. Moving between these two modes is an interesting research direction. For example, RONIN [46] creates (in an online fashion) navigations over the set of tables returned by a search query. In this tutorial, we will discuss some of the challenges with making semantic annotation solutions that are"query time" rather than batch offline tasks.

Many data lake solutions make use of knowledge graphs. Rather than viewing knowledge graphs as a static resource, an interesting open challenge is to view the data lake as a source of knowledge that can be utilized to verify and augment knowledge graphs. SANTOS [27] takes a step in this direction by creating a synthesize knowledge graph from parts of a data lakes that are not covered by existing knowledge graph. Integrating table discovery, both query and navigation, more comprehensively with knowledge graph augmentation is another theme we will highlight. Additionally, a data lake can itself be modeled as a graph or network. Such a view lets us use graph mining and network measures to better understand the data lake. One approach in this direction used graph centrality measures to find ambiguous values (homographs) within a data lake [32]. Another interesting direction is scaling up the graph embedding techniques developed for single tables [5] to massive table collections like data lakes. We will discuss other ways of using graph mining and network science to enhance table discovery.

4 TUTORIAL LOGISTICS

Target Audience Dataset discovery from data lakes has been a popular topic in the database community. As the presenters come from both academia and industrial institutes and have experience with data discovery practice, this tutorial targets researchers, engineers and practitioners with an interest in tabular data management. For database researchers, this tutorial can provide a comprehensive survey on recent studies and a vision of interesting opportunities in this field; Engineers and practitioners can expect the tutorial to provide useful insights in discovering datasets from large-scale data lake tables which is common in real commercial production pipelines. The expected background of this tutorial is that of an undergraduate level course of database systems. Some studies covered in the tutorial also requires related techniques about information retrieval and machine learning. Importantly, we will focus on data management innovations over the search/IR innovations covered elsewhere [7]. We will carefully provide necessary background knowledge and terminologies to help attendees unfamiliar with the topic to understand the basic concepts and problem settings.

Previous Tutorials This tutorial is spiritually connected to a previous one titled "Data Lake Management: Challenges and Opportunities" presented by Fatemeh Nargesian, Erkang Zhu, Renée J. Miller, Ken Q. Pu and Patricia C. Arocena in VLDB 2019 [43]. However, this tutorial substantially differs from the above tutorial in two aspects: (i) while the above tutorial provided a comprehensive overview of topics related to data lakes, this tutorial focused on the specific topic of dataset discovery from data lakes and make an in-depth discussion about its related works; (ii) this tutorial includes the significant new work since 2019 so as to cover the up-to-date advances in this important and growing field.

5 BIOGRAPHY

Grace Fan is a PhD student in the Khoury College of Computer Science at Northeastern University. Her research interests include data discovery and data integration in data lakes.

Jin Wang is a research scientist and research lead of Megagon Labs. He obtained his Ph.D. degree from the University of California, Los Angeles, and M.S. from Tsinghua University, both in computer science. His research interests include Data Integration, Database Query Language, and Natural Language Processing. He has published several papers in venues like SIGMOD, VLDB, ICDE, KDD, IJCAI, VLDB Journal, IEEE TKDE etc. He is a co-presenter of a CIKM' 19 tutorial on string similarity queries. He regularly served as program committee members of top conferences like SIGMOD, KDD, ICDM, AAAI and IJCAI. He is the member of ACM and IEEE.

Yuliang Li is an AI research scientist at Meta. His research interests include data integration and natural language processing. Li received multiple "bests of" paper awards from DEEM'22, VLDB'22, VLDB'19, and ICDT'19. He is a co-presenter of a VLDB'21 tutorial on data augmentation. He served as the program committee members of SIGMOD'20, SIGMOD'22 and VLDB'22 and reviewers for ACL and EMNLP. Before joining Meta, he was a senior research scientist at Megagon Labs.

Renée J. Miller is a University Distinguished Professor of Computer Science at Northeastern University. She is a Fellow of the

Royal Society of Canada, Canada's National Academy of Science, Engineering and the Humanities. She received the US Presidential Early Career Award for Scientists and Engineers (PECASE), the highest honor bestowed by the United States government on outstanding scientists and engineers beginning their careers. She received an NSF CAREER Award, the Ontario Premier's Research Excellence Award, and an IBM Faculty Award. She formerly held the Bell Canada Chair of Information Systems at the University of Toronto and is a fellow of the ACM. She and her colleagues received the 2013 ICDT Test-of-Time Award and the 2020 Alonzo Church Award for Outstanding Contributions to Logic and Computation for their influential work establishing the foundations of data exchange. Professor Miller is a former president of the Very Large Data Base (VLDB) Foundation and currently serves as Editor-in-Chief of the VLDB Journal. She received her PhD in Computer Science from the University of Wisconsin, Madison and bachelor's degrees in Mathematics and in Cognitive Science from MIT.

ACKNOWLEDGMENTS

This work was supported in part by NSF under award numbers IIS-1956096 and IIS-2107248.

REFERENCES

[1] Parag Agrawal, Arvind Arasu, and Raghav Kaushik. 2010. On indexing error-tolerant set containment. In *SIGMOD*. 927–938.
[2] Alex Bogatu, Alvaro A. A. Fernandes, Norman W. Paton, and Nikolaos Konstantinou. 2020. Dataset Discovery in Data Lakes. In *ICDE*. 709–720.
[3] Dan Brickley, Matthew Burgess, and Natasha F. Noy. 2019. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In *WWW*. 1365–1375.
[4] Michael J. Cafarella, Alon Y. Halevy, and Nodira Khoussainova. 2009. Data Integration for the Relational Web. *Proc. VLDB Endow.* 2, 1 (2009), 1090–1101.
[5] Riccardo Cappuzzo, Paolo Papotti, and Saravanan Thirumuruganathan. 2020. Creating Embeddings of Heterogeneous Relational Datasets for Data Integration Tasks. In *SIGMOD*. 1335–1349.
[6] Sonia Castelo, Rémi Rampin, Aécio S. R. Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. 2021. Auctus: A Dataset Search Engine for Data Discovery and Augmentation. *Proc. VLDB Endow.* 14, 12 (2021), 2791–2794.
[7] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. 2020. Dataset search: a survey. *VLDB J.* 29, 1 (2020), 251–272.
[8] Nadiia Chepurko, Ryan Marcus, Emanuel Zgraggen, Raul Castro Fernandez, Tim Kraska, and David R. Karger. 2020. ARDA: Automatic Relational Data Augmentation for Machine Learning. *Proc. VLDB Endow.* 13, 9 (2020), 1373–1387.
[9] Michael Chui, Diana Farrell, and Kate Jackson. 2014. How government can promote open data. *McKinsey Company* (2014).
[10] Joel Coffman and Alfred C. Weaver. 2014. An Empirical Performance Evaluation of Relational Keyword Search Techniques. *IEEE Trans. Knowl. Data Eng.* 26, 1 (2014), 30–42.
[11] Tianji Cong, James Gale, Jason Frantz, H. V. Jagadish, and Çagatay Demiralp. 2022. WarpGate: A Semantic Join Discovery System for Cloud Data Warehouses. *CoRR* abs/2212.14155 (2022).
[12] Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Y. Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. 2012. Finding related tables. In *SIGMOD*. 817–828.
[13] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *SCG*. ACM, 253–262.
[14] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: Table Understanding through Representation Learning. *Proc. VLDB Endow.* 14, 3 (2020), 307–319.
[15] Yuyang Dong, Kunihiro Takeoka, Chuan Xiao, and Masafumi Oyamada. 2021. Efficient Joinable Table Discovery in Data Lakes: A High-Dimensional Similarity-Based Approach. In *ICDE*. 456–467.
[16] Mahdi Esmailoghli, Jorge-Arnulfo Quiané-Ruiz, and Ziawasch Abedjan. 2022. MATE: Multi-Attribute Table Extraction. *Proc. VLDB Endow.* 15, 8 (2022), 1684–1696.
[17] Grace Fan, Jin Wang, Yuliang Li, Dan Zhang, and Renée J. Miller. 2022. Semantics-aware Dataset Discovery from Data Lakes with Contextualized Column-based Representation Learning. *CoRR* abs/2210.01922 (2022).
[18] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A Data Discovery System. In *ICDE*. 1001–1012.
[19] Raul Castro Fernandez, Essam Mansour, Abdulhakim Ali Qahtan, Ahmed K. Elmagarmid, Ihab F. Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. 2018. Seeping Semantics: Linking Datasets Using Word Embeddings for Data Discovery. In *ICDE*. 989–1000.
[20] Sainyam Galhotra and Udayan Khurana. 2020. Semantic Search over Structured Data. In *CIKM*.
[21] Alon Y. Halevy, Flip Korn, Natalya Fridman Noy, Christopher Olston, Neoklis Polyzotis, Sudip Roy, and Steven Euijong Whang. 2016. Goods: Organizing Google's Datasets. In *SIGMOD*. 795–806.
[22] Vagelis Hristidis and Yannis Papakonstantinou. 2002. DISCOVER: Keyword Search in Relational Databases. In *VLDB*. 670–681.
[23] Madelon Hulsebos, Kevin Zeng Hu, Michiel A. Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César A. Hidalgo. 2019. Sherlock: A Deep Learning Approach to Semantic Data Type Detection. In *SIGKDD*. 1500–1508.
[24] Stratos Idreos and Tim Kraska. 2019. From Auto-tuning One Size Fits All to Self-designed and Learned Data-intensive Systems. In *SIGMOD*. 2054–2059.
[25] Stratos Idreos, Kostas Zoumpatianos, Manos Athanassoulis, Niv Dayan, Brian Hentschel, Michael S. Kester, Demi Guo, Lukas M. Maas, Wilson Qin, Abdul Wasay, and Yiyou Sun. 2018. The Periodic Table of Data Structures. *IEEE Data Eng. Bull.* 41, 3 (2018), 64–75.
[26] Mehdi Kargar, Aijun An, Nick Cercone, Parke Godfrey, Jaroslaw Szlichta, and Xiaohui Yu. 2014. MeanKS: meaningful keyword search in relational databases with complex schema. In *SIGMOD*. 905–908.
[27] Aamod Khatiwada, Grace Fan, Roee Shraga, Zixuan Chen, Wolfgang Gatterbauer, Renée J. Miller, and Mirek Riedewald. 2023. SANTOS: Relationship-based Semantic Table Union Search. In *SIGMOD*.
[28] Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, and Asterios Katsifodimos. 2021. Valentine: Evaluating Matching Techniques for Dataset Discovery. In *ICDE*. 468–479.
[29] Oliver Lehmberg and Christian Bizer. 2017. Stitching Web Tables for Improving Matching Quality. *Proc. VLDB Endow.* 10, 11 (2017), 1502–1513.
[30] Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A Large Public Corpus of Web Tables containing Time and Context Metadata. In *WWW*. 75–76.
[31] Oliver Lehmberg, Dominique Ritze, Petar Ristoski, Robert Meusel, Heiko Paulheim, and Christian Bizer. 2015. The Mannheim Search Join Engine. *J. Web Semant.* 35 (2015), 159–166.
[32] Aristotelis Leventidis, Laura Di Rocco, Wolfgang Gatterbauer, Renée J. Miller, and Mirek Riedewald. 2021. DomainNet: Homograph Detection for Data Lake Disambiguation. In *EDBT*. 13–24.
[33] Keqian Li, Yeye He, and Kris Ganjam. 2017. Discovering Enterprise Concepts Using Spreadsheet Tables. In *SIGKDD*. 1873–1882.
[34] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. 2010. Annotating and Searching Web Tables Using Entities, Types and Relationships. *Proc. VLDB Endow.* 3, 1 (2010), 1338–1347.
[35] Xiao Ling, Alon Y. Halevy, Fei Wu, and Cong Yu. 2013. Synthesizing Union Tables from the Web. In *IJCAI*. 2677–2683.
[36] Colin Lockard, Xin Luna Dong, Prashant Shiralkar, and Arash Einolghozati. 2018. CERES: Distantly Supervised Relation Extraction from the Semi-Structured Web. *Proc. VLDB Endow.* 11, 10 (2018), 1084–1096.
[37] Jiaheng Lu, Chunbin Lin, Jin Wang, and Chen Li. 2019. Synergy of Database Techniques and Machine Learning Models for String Similarity Search and Join. In *CIKM*. 2975–2976.
[38] Yury A. Malkov and Dmitry A. Yashunin. 2020. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 4 (2020), 824–836.
[39] Renée J. Miller. 2018. Open Data Integration. *Proc. VLDB Endow.* 11, 12 (2018), 2130–2139.
[40] Renée J. Miller, Fatemeh Nargesian, Erkang Zhu, Christina Christodoulakis, Ken Q. Pu, and Periklis Andritsos. 2018. Making Open Data Transparent: Data Discovery on Open Data. *IEEE Data Eng. Bull.* 41, 2 (2018), 59–70.
[41] Fatemeh Nargesian, Ken Q. Pu, Bahar Ghadiri Bashardoost, Erkang Zhu, and Renée J. Miller. 2023. Data Lake Organization. *IEEE Trans. Knowl. Data Eng.* 35, 1 (2023), 237–250.
[42] Fatemeh Nargesian, Ken Q. Pu, Erkang Zhu, Bahar Ghadiri Bashardoost, and Renée J. Miller. 2020. Organizing Data Lakes for Navigation. In *SIGMOD*. 1939–1950.
[43] Fatemeh Nargesian, Erkang Zhu, Renée J. Miller, Ken Q. Pu, and Patricia C. Arocena. 2019. Data Lake Management: Challenges and Opportunities. *Proc. VLDB Endow.* 12, 12 (2019), 1986–1989.
[44] Fatemeh Nargesian, Erkang Zhu, Ken Q. Pu, and Renée J. Miller. 2018. Table Union Search on Open Data. *Proc. VLDB Endow.* 11, 7 (2018), 813–825.

[45] Masayo Ota, Heiko Mueller, Juliana Freire, and Divesh Srivastava. 2020. Data-Driven Domain Discovery for Structured Datasets. *Proc. VLDB Endow.* 13, 7 (2020), 953–965.
[46] Paul Ouellette, Aidan Sciortino, Fatemeh Nargesian, Bahar Ghadiri Bashardoost, Erkang Zhu, Ken Pu, and Renée J. Miller. 2021. RONIN: Data Lake Exploration. *Proc. VLDB Endow.* 14, 12 (2021), 2863–2866.
[47] Rakesh Pimplikar and Sunita Sarawagi. 2012. Answering Table Queries on the Web using Column Keywords. *Proc. VLDB Endow.* 5, 10 (2012), 908–919.
[48] Aécio S. R. Santos, Aline Bessa, Christopher Musco, and Juliana Freire. 2022. A Sketch-based Index for Correlated Dataset Search. In *ICDE*. 2928–2941.
[49] Mihail Stoian, Andreas Kipf, Ryan Marcus, and Tim Kraska. 2021. PLEX: Towards Practical Learned Indexing. *CoRR* abs/2108.05117 (2021).
[50] Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çagatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating Columns with Pre-trained Language Models. In *SIGMOD*. 1493–1503.
[51] Petros Venetis, Alon Y. Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu. 2011. Recovering Semantics of Tables on the Web. *Proc. VLDB Endow.* 4, 9 (2011), 528–538.
[52] Daheng Wang, Prashant Shiralkar, Colin Lockard, Binxuan Huang, Xin Luna Dong, and Meng Jiang. 2021. TCN: Table Convolutional Network for Web Table Interpretation. In *WWW*. 4020–4032.
[53] Gerhard Weikum. 2021. Knowledge Graphs 2021: A Data Odyssey. *Proc. VLDB Endow.* 14, 12 (2021), 3233–3238.
[54] Mohamed Yakout, Kris Ganjam, Kaushik Chakrabarti, and Surajit Chaudhuri. 2012. InfoGather: entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD*. 97–108.
[55] Ce Zhang, Jaeho Shin, Christopher Ré, Michael J. Cafarella, and Feng Niu. 2016. Extracting Databases from Dark Data with DeepDive. In *SIGMOD*. 847–859.
[56] Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çagatay Demiralp, and Wang-Chiew Tan. 2020. Sato: Contextual Semantic Type Detection in Tables. *Proc. VLDB Endow.* 13, 11 (2020), 1835–1848.
[57] Yi Zhang and Zachary G. Ives. 2020. Finding Related Tables in Data Lakes for Interactive Data Science. In *SIGMOD*. 1951–1966.
[58] Zixuan Zhao and Raul Castro Fernandez. 2022. Leva: Boosting Machine Learning Performance with Relational Embedding Data Augmentation. In *SIGMOD*. 1504–1517.
[59] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J. Miller. 2019. JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes. In *SIGMOD*. 847–864.
[60] Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, and Renée J. Miller. 2016. LSH Ensemble: Internet-Scale Domain Search. *Proc. VLDB Endow.* 9, 12 (2016), 1185–1196.