Burr: A Benchmark for Ontology Learning from Relational Databases

LUKAS LASKOWSKI, Hasso Plattner Institute, University of Potsdam, Germany
MICHAEL HLADIK, SAP SE, Germany
JAN PORTISCH, SAP SE, Germany
FABIAN PANSE, University of Augsburg, Germany
FELIX NAUMANN, Hasso Plattner Institute, University of Potsdam, Germany

Knowledge graphs and ontologies play an essential role in integrating, standardizing, and reasoning about complex data across domains. In recent studies, leveraging knowledge graphs in AI use cases, instead of traditional relational databases, led to quality improvements by up to 38 percentage points. However, learning ontologies from relational databases remains a challenging task due to the impedance mismatch between both modeling concepts. An understanding of which ontology learning system performs best, and why, is missing, as no established benchmark exists.

We present Burr[1], a benchmark for evaluating ontology learning systems from relational databases. To evaluate the ontology learning space, we introduce a novel mapping-based metric and provide a comprehensive benchmark data collection. This collection of 54 scenarios consists of real-world database-ontology mappings, including industry data, and of a micro-benchmark evaluating the behavior of systems in encapsulated scenarios. We demonstrate the applicability of Burr by evaluating widely used ontology learning systems, including traditional rule-based as well as LLM-based approaches, on the benchmark. The results emphasize the current strengths of simple rule-based approaches compared to LLM-based systems, while also highlighting the significant research potential of LLMs in ontology learning.

CCS Concepts: • **Computing methodologies** → **Ontology engineering**; • **Information systems** → **Information integration**; **Enterprise information systems**; **Ontologies**; *Language models*; **Information extraction**.

Additional Key Words and Phrases: Ontology learning; Ontology Mapping; Benchmark; D2RQ; LLM

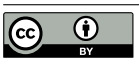
ACM Reference Format:
Lukas Laskowski, Michael Hladik, Jan Portisch, Fabian Panse, and Felix Naumann. 2025. Burr: A Benchmark for Ontology Learning from Relational Databases. *Proc. ACM Manag. Data* 3, 6 (SIGMOD), Article 305 (December 2025), 27 pages. https://doi.org/10.1145/3769770

1 Knowledge Graph Integration

During the last few years, artificial intelligence (AI) has disrupted society and businesses. AI systems help to optimize previously manual business and production processes and thus reduce costs for companies. A study by PWC [46] revealed that this trend continues, with artificial intelligence boosting the world's gross domestic product by 14% until 2030. However, artificial intelligence

[1]The name Burr is a reference to the character Aaron Burr (third Vice President of the United States) in Lin-Manuel Miranda's musical "Hamilton".

Authors' Contact Information: Lukas Laskowski, lukas.laskowski@hpi.de, Hasso Plattner Institute, University of Potsdam, Germany, Potsdam; Michael Hladik, michael.hladik@sap.com, SAP SE, Germany, Walldorf; Jan Portisch, jan.portisch@sap.com, SAP SE, Germany, Walldorf; Fabian Panse, fabian.panse@uni-a.de, University of Augsburg, Germany, Augsburg; Felix Naumann, felix.naumann@hpi.de, Hasso Plattner Institute, University of Potsdam, Germany, Potsdam.



ACM 2836-6573/2025/12-ART305
https://doi.org/10.1145/3769770

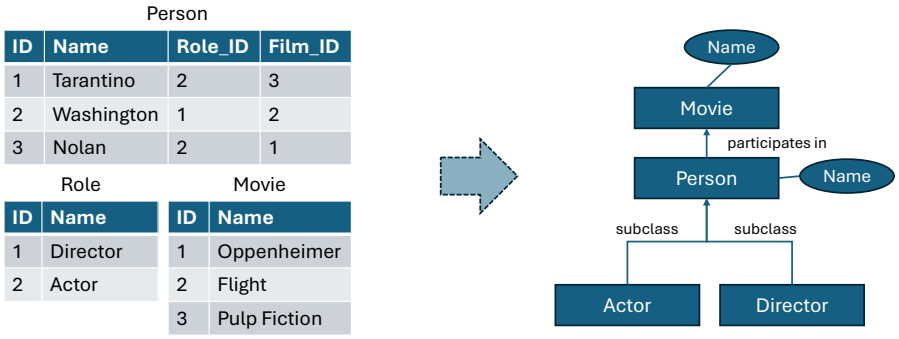


Fig. 1. **Ontology Learning.** Relational databases follow other design patterns than ontologies

methods are only successful if the underlying data is of high quality. Budach et al. [37] show that a poor data foundation has a significant negative impact on the quality of machine learning algorithms. It is crucial for companies to operate on high-quality data to effectively optimize processes using AI methods.

Interestingly, a study [2] by the open-source data quality tool "Great Expectations" found that 77% of all companies have data quality problems. According to the study, one of the main reasons for these quality problems is data incompatibility across different departments within the same company. To increase their data quality, companies must integrate data from their departments. Knowledge graphs are often used for this task, as they scale to many data sources, allow to easily link elements, and provide a holistic view of the data.

Hogan et al. [28] define a knowledge graph (KG) as a graph structure that represents knowledge of the real world, with nodes representing entities and edges representing relationships between those entities. Knowledge graphs are typically based on ontologies: ontologies define the semantic data model or vocabulary of a knowledge graph, i.e., the schema of the KG, but do not contain instance data. However, as companies usually store their data in relational databases and spreadsheet programs, these data sources need to be converted into knowledge graphs. Knowledge graphs are more easily extensible and, for example, a study by Sequeda et al. [51] has shown that AI use cases achieve higher qualities when working on knowledge graph representations than on relational databases. One challenging task for the conversion of databases to knowledge graphs is the *extraction of ontologies from relational data schemata*. The knowledge graph is then constructed based on the learned ontology. However, the database schema cannot be directly mapped to an ontology, since doing so leads to semantically weak models. For instance, databases are typically optimized to serve a specific use case. For example, they are often denormalized for read-heavy workloads to reduce query times for analytical queries. Thus, naively mapping each table to a concept and each attribute to an ontological attribute does not yield a correct ontological representation of the domain. For instance, some tables might hold multiple concepts and other tables, such as N-to-M tables, might not represent a concept at all. Consequently, the ontology would merely be a reflection of the relational database and not a representation of the real world.

Figure 1 illustrates the challenge of converting a database to an ontology: Although the tables `Person` and `Movie` could be represented using a concept in the ontology, the table `Role` cannot easily be converted. Here, each distinct record in the table `Role` represents an individual concept in the ontology. Furthermore, an ontology learning system needs to recognize that `Person` and `Movie` are in an "is_part_of" relationship, while subclass relationships need to be created for `Person` and the attribute values of `Role`.

This example motivates the need for ontology learning systems. Given a database schema, *ontology learning* aims to create an ontology [58], which consists of concepts, attributes, and relationships and which describes the domain of the input. Such systems need to extract the domain of the relational database and model the underlying knowledge. Therefore, it is not sufficient to copy the relational database; the systems need to understand what concepts look like and which relationships exist between them. In addition to the ontology, we expect an ontology learning system to also output a mapping that maps the schema elements of the input database to those in the created ontology. Thereby, each element in the ontology can be composed of complex queries on the input database. There are specific mapping languages for this purpose, such as R2RML [65] or D2RQ [9]. This mapping can transform the records of the database into instances of the knowledge graph. Thus, the ontology has only practical use when its corresponding database-ontology mapping is provided.

Learning ontologies from relational databases is a difficult task. To be able to holistically measure the quality of a learned ontology, several challenges need to be solved. First, *suitable benchmark data* that is used for evaluation needs to be selected. Thereby, the data should reflect real-world scenarios that the systems are expected to handle. The benchmark data needs to cover a sufficiently wide range of situations and edge cases, so that the systems' behavior can be studied as well as potential strengths and weaknesses identified. Second, a *precise evaluation metric* should be selected when analyzing the output of a system. The metric should accurately and comprehensively assess the quality of the resulting ontology. Thereby, the evaluation metric should be consistent and comparable across different systems and benchmark datasets.

We conducted a review of ten ontology learning systems following the review of Mosca et al. [38] and twelve prominent ontology mapping systems [68]. Our findings reveal that most papers do not evaluate their approach *at all* or provide only lexical comparisons without stating how the F_1-score is computed specifically. Some systems evaluate differently, e.g., by using query coverage, manual effort to correct the ontology, or task performance; however, they do not rely on a standardized evaluation approach.

To fill this gap, we present Burr – a benchmark for evaluating ontology learning systems from relational databases. It aims to unify the evaluation process of ontology learning systems. Thereby, we propose a novel evaluation approach, including a metric that evaluates quality using mappings from the database to the ontology, i.e., the actual references to tables and columns in the databases. Although the metric still uses the F_1-score, we redefine the meaning of a true positive based on the database-ontology mappings. By proposing a complete micro-benchmark and real-world scenarios, Burr is the first to provide practitioners with a complete picture of how an ontology learning system performs and what its strengths and weaknesses are compared to other systems. Such a standardization of evaluation allows a precise assessment to understand a system's quality in real-world conditions and comparisons to other systems. With Burr, we make the following contributions:

(1) **Mapping-based evaluation metric**: We propose a new mapping-based evaluation metric that enables a precise evaluation of ontology learning systems from databases.
(2) **Burr benchmark**: We propose a conceptual benchmark for ontology learning from relational databases, which includes a framework for evaluation and a taxonomy of challenges derived from real-world database scenarios.
(3) **Mapping scenarios**: We provide a practical realization of Burr including 54 manually created database-ontology-mapping scenarios for evaluating ontology learning systems. Firstly, we offer a set of micro benchmarks that allow to analyze the behavior of a system in encapsulated scenarios. Furthermore, Burr contains a suite of real-world databases, corresponding

ontologies, and their mappings, enabling evaluation under realistic conditions. This part of the benchmark allows for testing the practicability of a system.

(4) **Comprehensive evaluation**: We show the applicability of Burr by executing it on two popular rule-based ontology learning systems [6, 14, 40], one LLM-based ontology learning system for CSV-data [59] that we extended to databases, and three LLMs with four different prompting strategies.

Burr is published as open-source. Its code and further artifacts, in particular our collected benchmark scenarios and detailed results, are accessible at https://github.com/HPI-Information-Systems/Burr.

The remainder of the paper is structured as follows: In Section 2, we introduce related work. We discuss the general requirements for a benchmark and present existing evaluation approaches in Section 3. Next, Section 4 introduces the newly proposed mapping-based evaluation metric. In Section 5, we present the Burr-benchmark and describe its structure. We discuss the results of executing several ontology learning systems on Burr (see Section 6). Finally, in Section 7, we conclude and give an outlook on future work.

2 Related Work

The research area of ontology learning [35] has been explored for many years. It is not limited to the extraction of ontologies from relational databases, but is based on different input modes. Apart from relational databases, ontologies are often learned from natural language text [3, 21, 34] or semi-structured data [62], such as HTML or XML. Especially recent advancements in the field of deep learning and large language models have accelerated new research directions as examined by Babaei Giglou et al. [4].

Ontology Learning from Databases. The process of learning ontologies [38] usually involves extracting schema information, e.g., tables and attributes to ontology constructs, e.g., concepts and relationships. Most approaches leverage rule-based techniques to extract ontologies. For example, Shen et al. [56] define mapping rules for concepts, relationships, and restrictions that define how database schema elements are translated to OWL-constructs. OWL [63] is a formal description language specified by the "World Wide Web Consortium" (W3C) to define ontologies. Ben Mahria et al. [6] present an approach that describes the whole life cycle from learning the ontology to creating a knowledge graph based on the actual database records. We use this approach in our experimental evaluation in Section 6. Similar to Shen et al., they propose rules to extract the ontology out of a relational database. These rules include simple mappings, such as mapping each table to a concept. They also include an advanced handling of specific database concepts, such as N-to-M tables that are then mapped to relationships. In this paper, we use Burr to evaluate how well ontology learning systems can handle concepts, such as N-to-M tables in Section 6.

Ontology learning is different from *ontology matching*: While ontology learning aims to create a novel ontology from a database schema, ontology matching aims to find correspondences between two different, already existing ontologies [24]. Hence, ontology learning is relevant in the field of knowledge graph construction, while ontology matching is rather relevant for schema integration use cases. Despite the name similarity, *matching learning* is also different from *ontology learning*; it encompasses leveraging machine learning within ontology matching systems [25].

Semantic Table Interpretation. A related research area to ontology learning from relational databases is semantic table interpretation. Liu et al. [32] present in their survey a set of tasks that interpret data stored in tables and spreadsheets as semantic data. The goal is to present differently formatted data in a unified way to make it processable by machines. To do so, they define subtasks

that extract entities, attributes, and relationships out of a table. These subtasks could be used to extract concepts, attributes, and relationships, and thus an ontology from relational data. For example, the task *Columns-Property Annotation* (CPA) [18, 47] aims to detect pairs of columns within a table that relate to each other. Solutions to this task might contribute to extracting relationships of an ontology within a table. *Column-Type Annotation* (CTA) [18, 60, 61] identifies knowledge graph types for a column. This task might be useful for detecting concepts of an ontology.

Ontology Mapping. While ontology learning from databases creates a new ontology based on a database, ontology mapping [54, 68] defines the task of automatically creating a mapping between a database and an already existing ontology. Ontology mapping focuses on the task of integrating relational data into an *existing* ontology – primarily for the task of ontology-based data integration, called OBDA [30, 31, 66]. Ontology mapping is different from ontology learning, since relational-to-ontology mapping requires a target ontology that does not exist yet in the case of ontology learning. Many systems, such as BootOX [29] or MIRROR [16], build upon simple rules. For example, BootOX translates each table, except N-to-M tables, into an OWL-Class, attributes to attributes in ontologies, and foreign key constraints to relationships.

Ontology Evaluation. Learned ontologies and extracted mappings between databases and ontologies have been evaluated in different ways. The following section surveys these evaluation approaches in more detail and places them in context with Burr.

3 A Survey on Ontology Evaluation

A reliable and efficient evaluation benchmark is relevant to iteratively assess and improve ontology learning approaches. Immediate feedback helps to quickly analyze whether changes to the ontology learning approach are beneficial. In addition, a benchmark helps in comparing systems with each other and in finding the best system for a specific use case. Ideally, the benchmark enables the user to test the system's behavior in different scenarios and thus understand its advantages and disadvantages. In the past, ontology learning approaches have been evaluated in different ways. Generally, according to Dai and Berleant [15], benchmarks should fulfill several criteria to enable a fair comparison of the different systems that should be evaluated:

(1) **Relevance**: The chosen metrics should accurately reflect the essential qualities of the system that are critical for its intended use.
(2) **Representativeness**: Metrics that are used for evaluation should be recognized in research and industry.
(3) **Equity**: The benchmark should ensure fair comparisons across all systems.
(4) **Repeatability**: The results should be consistently reproducible.
(5) **Cost-effectiveness**: The tests can be conducted without excessive expense.
(6) **Scalability**: The benchmark adapts to varying sizes.
(7) **Transparency**: Metrics and their calculation should be easy to understand.

There are already various evaluation and benchmark approaches that can be used to evaluate the quality of learned ontologies [42, 45, 48, 68]. Table 1 shows which benchmarks in the domain of ontology learning from databases fulfill which benchmark characteristics. In this section, we present the existing evaluation approaches, which we classify as "task-based", "pattern-based", "manual", and "lexical". We describe how they are used to evaluating ontologies and discuss their advantages and disadvantages, considering the benchmark requirements of Dai and Berleant [15].

Table 1. **Database-ontology benchmarks.** Rated based on the benchmark characteristics proposed by Dai and Berleant [15]

	Task [68]	Pattern [45]	Manual [42]	Lexical [48]	BURR
Relevance	✗	✗	✓	✓	✓
Representativeness	✓	✗	✓	✓	✓
Equity	✗	✓	✗	✗	✓
Repeatability	✓	✓	✗	✓	✓
Cost-effectiveness	✓	✓	✗	✓	✓
Scalability	✓	✓	✗	✓	✓
Transparency	✓	✓	✗	✓	✓

3.1 Task-based Evaluation

Task-based evaluation assesses how well the ontologies work for a specific use case. To do so, these methods pose representative tasks that must be solved with the help of the ontology.

Zaveri et al. [68] present the RODI-Benchmark, which comprises three different scenarios. Each scenario has several tasks, each of which consists of a database, an ontology, and several SQL queries. Each task poses different challenges to ontology mapping systems, such as naming conflicts, structural heterogeneities, and semantic heterogeneities. The systems are expected to learn a mapping between the database and the ontology. Then, SQL queries against the database are translated into SPARQL queries using the created mapping. Subsequently, the SQL query is executed on the database and the SPARQL query on the knowledge graph (ontology); the result sets are compared afterward using F_1-score.

This technique can also be used to evaluate learned ontologies. It evaluates precisely how the created ontology helps to solve a specific task. Only the quality of the results when executing the task is assessed and no confounding factors, such as different modeling decisions, influence the evaluation. However, the evaluation metric is biased towards the specifically designed task. No general statement can be made about how well the ontology represents the database domain. An ontology could perform particularly well in the execution of a certain task, while it does not achieve good quality in another task.

Following the benchmark criteria by Dai and Berleant [15], illustrated in Table 1, this approach is not suitable for assessing a system because it does not evaluate how accurately the ontology represents the real-world but only the capabilities to solve a specific task. Furthermore, it does not treat all systems equally, as systems specifically developed for this task would be favored. However, we aim to find ontology learning systems that represent the real-world domain as precisely as possible. Consequently, this method should only be used to evaluate the quality of ontologies for a clearly specified task, not for the holistic evaluation of ontologies.

3.2 Pattern-based Evaluation

Pattern-based evaluation methods aim to detect poor design decisions. To do so, they scan the ontology for typical bad practices in ontology design that were defined beforehand and often lead to errors and inconsistencies.

Poveda-Villalón et al. propose a system called “Oops!” [45] detecting 40 bad practices in ontology design, ranking severity, and proposing potential improvements. It detects bad practices in different dimensions, such as the structural or the functional dimension. Such approaches have the advantage

that they do not require reference ontologies for evaluation. Consequently, the quality assessment is not influenced by divergent modeling decisions in the learned and reference ontology. However, the pattern-based evaluation method does not evaluate the semantic correctness or completeness of the learned ontology, but only assesses its structural properties. Furthermore, such systems are incomplete and can only detect known and then modeled bad practices.

Pattern-based evaluation approaches should be used to analyze the structural quality of ontologies. However, they form only part of the quality assessment: otherwise important factors are ignored. Thus, the evaluation approach is not suitable for evaluating an ontology learning system holistically (see "Relevance" in Table 1). Additionally, pattern-based evaluations only count the amount of detected pitfalls and do not use widely used metrics, such as F_1-score ("Representativeness"). Ideally, systems that detect bad practices can be used directly during the creation of ontologies to prevent potential misdesigns already during the creation process.

3.3 Manual Evaluation

Experts can have a profound understanding of the respective domain as well as of ontology design. This allows them to manually evaluate the quality of the designed ontology by analyzing its structure and content. For example, Pak and Zhou [42] propose an evaluation framework with error categorizations. Such frameworks can be used to structure the manual evaluation process.

Manual evaluation benefits from the respective experts' in-depth understanding of the application domain. They can reliably assess whether the respective design decisions correctly reflect reality. The expert can also access reference ontologies and compare them manually. This method is not affected by naming differences between the engineered and the ground truth ontology, as an expert can recognize which names represent the same concepts in the ontologies. However, manual evaluation is also subjective, and its outcome depends on the respective expert. To minimize subjectivity, the evaluation should follow a clearly defined process.

Furthermore, the manual evaluation process is usually very time-consuming. Consequently, such approaches are not cost-effective and do not scale well, as shown in Table 1, and users do not receive immediate feedback on the quality of their ontology, forestalling rapid iterative improvements. In addition, there are also the challenges of reproducibility and transparency, as different persons evaluate the ontology differently without providing detailed insights into their decision-making.

3.4 Lexical Evaluation

The lexical evaluation method relies on the comparison of a learned ontology to a (single) source of truth based on the lexical names of the concepts, attributes, and relationships. It assumes that an ontology exists which defines the domain to be covered precisely and correctly. Figure 2 illustrates the setup: Based on a database D, an ontology learning system learns the ontology O_L. The ontology consists of a set of concepts C_L, a set of relationships R_L, a set of attributes A_L. This evaluation approach then compares for each ontology element whether an ontology element in the ground truth exists that has the same name.

The *lexical precision* and *lexical recall* as defined by Sabou et al. [48] provide a means of assessing the quality of the learned ontology O_L in relation to the ground truth ontology O_{GT}. The lexical precision LP and lexical recall LR are defined as:

$$LP(C_{GT}, C_L) = \frac{|C_{GT} \cap C_L|}{|C_L|} \quad (1) \qquad LR(C_{GT}, C_L) = \frac{|C_{GT} \cap C_L|}{|C_{GT}|} \quad (2)$$

Lexical recall can be used to analyze how many concepts of the ground truth ontology are correctly detected by the learned ontology. However, to assess the overlap correctly, these metrics require the same concepts in O_{GT} and O_L to also have the same name. Otherwise, the equality of

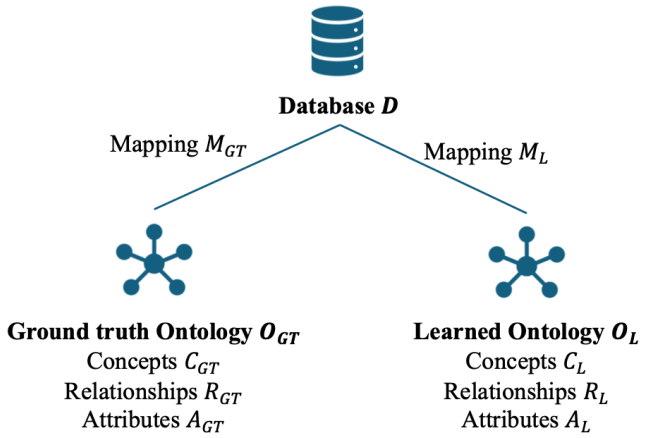


Fig. 2. **Structure of a mapping-based evaluation.**

these concepts is missed by this metric. Nevertheless, many concepts in ontologies could have multiple names for the same meaning, such as "car" and "automobile". Thus, systems that follow a different (but valid) naming convention than the ground truth ontology would not be treated fairly compared to systems that do. For this reason, the "Equity" condition in Table 1 is not fulfilled. Additionally, Sabou et al. [48] evaluate only the extraction of concepts and do not consider the extraction of relationships or attributes.

Additionally, this evaluation approach depends on a source of truth. Without having a ground truth available, the lexical evaluation is not applicable. Lastly, *LP* and *LR* do not consider structural differences in the ontology. For example, they do not test whether the ground truth ontology and the engineered ontology connect the same concepts in the database. Nevertheless, they can be used as part of an evaluation to assess how well the names of ontology elements are curated according to a given terminology.

4 Mapping-based Evaluation

To avoid inaccuracies in the ground truth-based evaluation process, we propose novel mapping-based evaluation metrics. These metrics exploit the mappings from the relational database to the respective ontologies. As described in Section 1, a mapping M from a relational database to the ontology is a necessary output of the ontology learning systems. Without a mapping, it is not defined how relational data could be transformed into a knowledge graph. Thus, the ontology could not be practically used for queries or other workflows. Mapping-based quality metrics enable the evaluation of how well ontology learning systems can extract database elements relevant to the ontology. Later, in Section 6, we experimentally demonstrate these advantages.

We evaluate the quality of an ontology using the mapping of the underlying database to the respective ontology and compare it to the ground truth mapping. As the learned mapping and the ground truth mapping refer to the same database, they can be used to trace which elements in the ontologies refer to the same schema element in the database. This helps to understand which concepts in O_L and O_{GT} represent the same information. In contrast to the lexical metrics presented in Section 3.4, metrics relying on this mechanism are not susceptible to naming conflicts.

We collect for each ontology element o the related schema elements in the database, i.e., those elements that are included in the mapping to construct o out of the database. Similarly to the lexical ground truth metrics (see Section 3.4), we calculate precision, recall, and F_1-score. In contrast to

Section 3.4, we define the intersection of the schema elements of two ontologies O_1 and O_2 as the intersection of their mappings:

$$O_1 \cap_M O_2 = \{e_1 \in O_1 | \exists e_2 \in O_2 : M(e_1) = M(e_2)\} \tag{3}$$

The mapping-based intersection $\cap_M$ treats two elements as equal when their mappings are equal, i.e., they refer to the same database elements. By doing so, the evaluation metric is not dependent on the names of the concepts, but on the information the ontology elements represent. Thus, the system under evaluation is not penalized by selecting different names compared to the ground truth ontology. Using this mapping-based intersection definition, we can calculate mapping-based precision and recall:

$$MP(O_L, O_{GT}) = \frac{|O_L \cap_M O_{GT}|}{|O_L|} \tag{4}$$

$$MR(O_L, O_{GT}) = \frac{|O_L \cap_M O_{GT}|}{|O_{GT}|} \tag{5}$$

The mapping-based F_1-score is the harmonic mean of MP and MR:

$$MF_1(MR, MP) = 2 \cdot \frac{MP \cdot MR}{MP + MR} \tag{6}$$

While the mapping-based evaluation still relies on widely used metrics, such as the F_1-Score, we redefine what constitutes a true positive, i.e., how two elements match. In contrast to existing F_1-Score-based methods, we do not rely on lexical measures for comparison, but compare based on the database reference of two ontology elements. This evaluation technique can be used to measure how well the generated ontology-based representation preserves the semantics and structure of the original database. This analysis is essential for using the mapping/ontology to build a virtual knowledge graph representation of the database. It fulfills the category "Relevance" in Table 1 because Burr evaluates only those parts of the database that are actually covered by the ground-truth ontology. This is a crucial requirement when the engineered ontology is used, e.g., for querying data. Using the widely recognized F_1-Score as a metric ensures that the mapping-based evaluation approach is representative. Furthermore, it treats all systems equally ("Equity" in Table 1), irrespective of naming conventions. In a task-based evaluation, for example, ontologies are not evaluated in a general way, but only for a specific downstream-task.

5 The Burr Benchmark

Using our novel evaluation metric (see Section 4) requires implemented mappings between a database and a ground-truth ontology. To obtain a holistic overview of the quality of an ontology learning system, a diverse set of mapping scenarios should be part of that evaluation, each testing different characteristics of such systems. Burr provides a micro-benchmark of encapsulated scenarios that serves the use case of evaluating the systems' behavior in individual encapsulated scenarios. In addition, it contains several real-world scenarios that are based on real-world databases. These scenarios test the systems' behavior under in-practice conditions.

The ground truth ontology serves as a reference framework by providing an objective baseline against which the structure and semantics of other ontologies can be evaluated. When modeling the corresponding mappings and ontologies for a database, different designs are suitable. Because of different perspectives and design preferences, different ontology experts might model the same domain differently. To maintain simplicity, Burr is designed in a flexible and easily extensible way, allowing quick *adaptation* to other design decisions. Thus, we ensure fairness across diverse systems and accommodate this challenge.

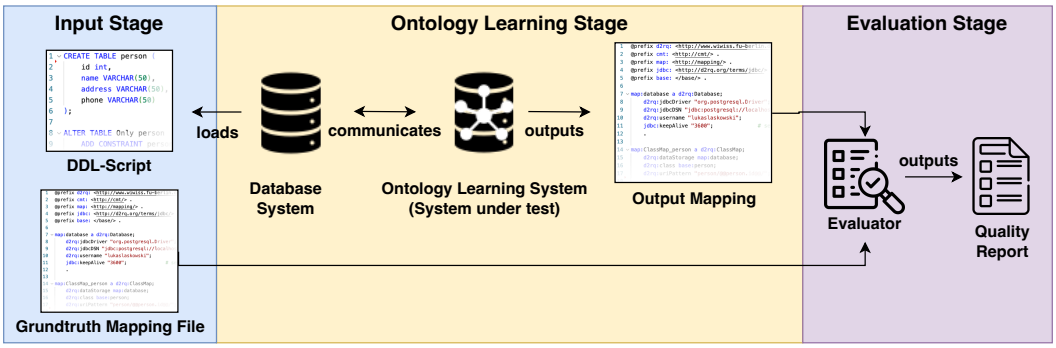


Fig. 3. **Workflow of Burr.** After loading the mapping scenarios, the system is executed and the produced mapping evaluated.

5.1 Benchmark Setup

Figure 3 illustrates the workflow of Burr. Each scenario consists of a DDL-script that describes the schema of the underlying database ("Input Stage" in Figure 3). To do so, it contains the descriptions of the tables to be created, any constraints, such as primary keys or foreign keys constraints, and exemplary data. Next to the schema file of the relational database, a ground truth mapping exists that describes in which way the tables and attributes of the database are mapped to the domain ontology. The domain ontology can be derived based on the mapping file as all necessary information is included, such as its concepts, relationships, and attributes. In our experiments, we used the mapping language *D2RQ* developed by Bizer and Seaborne [9]. D2RQ defines a language that can be used to define how to translate data in relational databases to knowledge graphs. We opted for D2RQ because it is specifically developed for relational databases. Additionally, it is the most flexible, for example by allowing conditional mappings, which, others mapping languages, such as R2RML [65] or RML [19], do not allow out of the box. We provide all mappings in a JSON-format that can be easily used to convert to R2RML or other mapping languages.

To assess the quality of the ontology learning systems, we applied the following procedure ("Ontology Learning Stage" of Figure 3): In the first step, the database is created based on the DDL-script. Then, the ontology learning system receives the database as input and extracts the ontology as well as the associated mapping. This generated mapping is subsequently evaluated based on the ground truth mapping file using the evaluation metrics presented in Sections 3 and 4 ("Evaluation Stage" in Figure 3). Finally, the resulting quality scores, such as the mapping-based F_1-score, are summarized in a quality report.

5.2 Micro-Benchmark

The micro-benchmark of Burr enables a precise evaluation of the behavior of ontology learning systems in encapsulated scenarios. Each scenario represents individual small databases that pose a specific challenge to the ontology learning system. This setup allows for understanding whether an ontology learning system is capable of handling a specific model situation. Each of the benchmark databases includes only the schema elements that are required to represent the associated database construct. The databases are of different domains, such as product databases or academic databases.

To obtain a holistic view of the quality of the ontology learning system under evaluation, we collected 54 scenarios and grouped them by their database structure. These scenarios test basic functionalities of ontology learning, such as representing tables or attributes of databases appropriately in an ontology. Apart from testing basic functionalities, the scenarios also test more

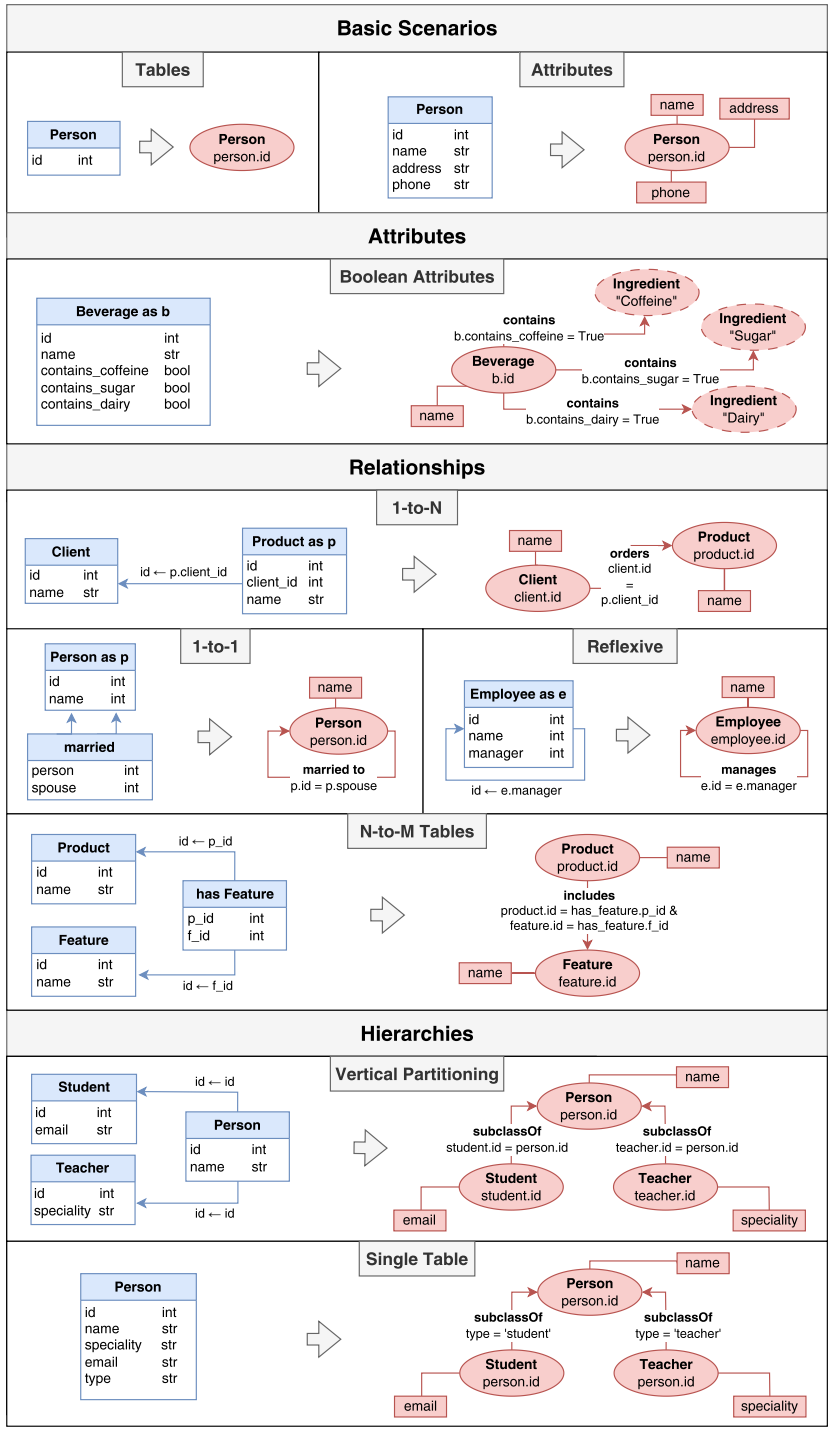


Fig. 4. **Micro-Benchmark.** A selection of challenging database-ontology pairs. Rectangles (blue) represent tables of a database. Ellipses (red) represent concepts of the ontology. Text at the arrows represent conditions and joins included in the mappings.

advanced capabilities that exist in real-world scenarios, such as the mapping of N-to-M tables or the representation of table hierarchies in ontologies. We compiled a diverse set of scenarios by exploring database concepts described in books and papers [5, 23, 27] and investigating existing work on mapping patterns [52, 53, 66]. For each of these database-mapping pairs, we provide approximately 500 meaningful records for the input database, which were generated using GPT-o4-mini-high. Additionally, we analyzed industrial real-world database-ontology pairs. These real-world scenarios contain complex mappings between the database and the ontology, which both already existed. For example, we used a process database and the corresponding OWL-ontology from a large enterprise company. Then, we manually added a mapping between them.

Figure 4 illustrates a selection of challenging database-ontology-mappings. All database-mapping pairs are accessible at https://github.com/HPI-Information-Systems/Burr. In the following, we introduce the scenarios categorized by their overarching challenge and describe them in more detail.

5.2.1 Basic constructs. These scenarios test the most fundamental capabilities of an ontology learning system. We group as follows:

Table. Mapping a single table and its attributes to a concept in the ontology ("Tables" in Figure 4).

Attributes. Mapping an attribute of a table to an attribute in the ontology ("Attributes" in Figure 4).

Relationship. The system is tested to detect a foreign key relationship between two tables and represent it as a relationship while extracting the correct join partners.

No PK Constraint available. Many ontology learning systems use primary keys to identify concepts. Without having such information available, these systems need to find other ways, such as detection of unique column combinations [8]. In the scenarios of this group, we evaluate those capabilities.

No FK Constraint available. In these scenarios, we analyze how well ontology learning systems identify relationships without having foreign keys available.

5.2.2 Attributes. In this category, the qualities of ontology learning systems are tested based on seven different groups:

Opaque Names. The database attribute names are cryptic and do not hint to the content of the respective columns. Thus, an ontology learning system needs to correctly extract appropriate concepts and relationships based on the actual records and metadata, such as foreign key constraints.

Boolean attributes. Boolean attributes represent certain properties of an entity, such as to which industries a specific product belongs. In the specific example depicted in Figure 4 (see category "Attributes"), a database is defined that includes ingredient information for beverages. Each ingredient is represented as a Boolean attribute. This mapping is solved by creating the two concepts `Beverage` and `Ingredient`. Each record of the `Beverage` table is then mapped to a specific instance of the concept `Ingredient` if its value in the corresponding Boolean attribute is set to "True". We observed a similar mapping pattern in an industrial database-ontology scenario. An ontology learning system must recognize concepts hidden in Boolean attributes and understand that only instances of these concepts are connected whose corresponding database records have the value "True" in the respective attributes.

Weak entity. A weak entity cannot be uniquely identified by its own attributes and depends on one or more related strong entities for its identification [23]. For example, only by combining the attributes `room-number` and `hotel-id` allows a unique identification of a hotel room. An ontology learning system should detect the dependency between the tables and correctly represent the unique composite key to identify the weak entity.

Composite attributes. In this scenario, an attribute contains several pieces of information that must be separated for the ontology. For example, the attribute `address` is split into `street`, `city`, `state` and `country`.

Multi-valued attributes. In the real world, entities can have multiple values for the same attribute (e.g., a person has multiple first names or hobbies). However, the relational data model requires strict atomicity for attribute values, meaning that multi-value attributes must be modeled using an additional table [23, 27]. In ontologies, however, it is possible to model multi-value attributes. The challenge for an ontology learning system is therefore to recognize that the additional table does not correspond to a separate concept, but to a multi-value attribute of the referenced concept.

5.2.3 Relationships. In relational databases, relationships between concepts can be modeled in different ways [23]. This depends on whether they are 1-to-1, 1-to-N, or N-to-M relationships, but also on whether they have additional attributes, and how many entities are involved in such a relationship. To validate all these subtleties, we evaluate on seven different groups:

Binary 1-to-N. These database constructs contain a relation that links a single record from one table (1-side) to multiple records in another table (N-side). Figure 4 illustrates one possible design of this relationship: a reference to the client is stored in the `product` table. Another design option is to have an additional table that stores the information of which person bought which product. We have added a scenario for both designs. In both scenarios, the ontology learning systems are expected to build the same ontology.

Binary Reflexive. In relational databases, records can have relationships with other records from the same table. In Figure 4, the scenario of a worker-manager relationship is illustrated. Both, the worker and manager are stored in an `employee`-table. For each record, a reference to the same table, i.e. the manager of the person, is stored. In such cases, the ontology learning system needs to detect this reflexive relationship and correctly represent it.

Binary 1-to-1. This relationship associates each record in one table with exactly one record in another table, and vice versa. These relationships can be modeled in different ways. In Figure 4 an additional table is used. Here, a marriage scenario is illustrated. In table `married`, a reference to two records of the table `person` is stored. Each record can have exactly one partner. By doing so, every marriage is represented by two records in the table `married`. These aspects need to be considered when learning an ontology and its corresponding database mapping. Another modeling approach is to use a simple foreign key within the `person` table that links every person to another. We have added a scenario for both approaches. In both scenarios, the ontology learning systems are expected to build the same ontology.

Binary N-to-M. In this scenario, the foundational case of N-to-M relationship tables is considered: Two tables, connected via an N-to-M table. Such tables are used to represent complex relationships in databases, i.e., when multiple records in a table are associated with multiple records in another table. The N-to-M table should not be represented as a concept in an ontology, but as a relationship of the two participating tables.

Additional Attributes (N-to-M). Apart from the foreign key attributes, an N-to-M table might have additional attributes to describe the relationship in more detail. Since relationships in ontologies cannot have attributes, in such a case, an additional concept must be created that represents the N-to-M table and contains these additional attributes. Furthermore, an explicit connection between the two participating concepts needs to be modeled by the ontology learning system to represent the actual relationship.

Composite Keys (N-to-M). Primary Keys can comprise one or more attributes. All key attributes need to be included in the foreign keys used by N-to-M tables. However, many related ontology

learning systems detect N-to-M tables by selecting tables that consist of exactly two attributes that form the primary key and each attribute having a foreign key reference to some other table. In such cases, they miss N-to-M tables that build upon composite keys.

Ternary N-to-M. Ternary relations represent relationships in which three entities are involved. In relational databases, such a relationship is represented by an N-to-M table, but with three participating foreign keys that reference three other tables [23]. The ontology learning system needs to detect such relationships and model them without modeling the connecting tables as separate concepts.

5.2.4 Normalization. Databases are often designed to serve a specific use case or query workload. They might be denormalized to decrease the number of necessary joins, which are costly. However, a denormalized database design does not reflect reality as aimed in ontologies. Furthermore, enterprise databases are often extended over time. These changes lead to inconsistencies and irregularities in the database design. Ontology learning systems must be able to deal with such scenarios and continue to produce high-quality ontologies. Apart from denormalized databases, also strictly normalized databases [22] exist. However, the normalization degree of a database does not always align with the requirements of an optimal ontology. To construct an accurate ontology, it is often necessary to join tables from normalized databases or decompose tables from denormalized ones. Normal forms (NFs) are guidelines for structuring tables to minimize redundancy by organizing data into logical groups [11, 12]. Each normal form defines a specific set of requirements that the database must meet. We include the scenario of an office database that stores employee information. The database exists in multiple versions, each being in a different normal form (NF 1 to 3). Since the modeling differs for all three versions, but not the actual information they provide, they are all mapped to the same ontology. Using these mappings allows comparing the capabilities of the systems to extract the correct ontology based on different normalization levels of the same database. In addition, we provide four denormalized scenarios that we identified occurring in real-world settings. For example, one scenario includes redundantly stored information that results from joining one referenced table with two referencing tables during the denormalization process.

5.2.5 Hierarchies. The relational data model does not have an explicit concept for modeling concept hierarchies, so that one of several workarounds must be used instead [23, 27]. In database ontology mapping systems, In systems for creating the mapping between a given database and a given ontology, a subset of such hierarchy scenarios are already discussed and evaluated [44]. However, ontology learning systems have not been tested on hierarchical relationships between concepts. Burr provides test scenarios for each of these workarounds, which all map to the same ontology. In all cases, the ontology learning system needs to detect the existing hierarchy with its superconcepts and subconcepts, and allocate the attributes from the different tables to the correct concepts. We included the five following scenario groups:

Vertical partitioning. Attributes of the superconcept are only stored in the parent table and not propagated to its subtables. Figure 4 illustrates a corresponding student-teacher scenario where the two tables `student` and `teacher` have specialized attributes that the other table does not have. Additionally, there is the parenttable `person` that stores all attributes shared by the two corresponding subconcepts, such as the `name`.

Concept combinations. Entities can belong to several subconcepts within the same hierarchy. For example, a person might be reviewer and author at the same time. One way to represent this within a relational database is to create one subtable for every potential combination of subconcepts [27]. Ontology learning systems must recognize such combinations and be able to extract the correct set of subconcepts from them.

Table 2. **Overview of Real-World Scenarios.**

	Database			Ontology		
	#Tables	∅Attr	#FK	#Concepts	∅Attr	#Rel
Mondial	46	3.96	33	27	3.30	63
Enterprise	8	22.75	0	20	4.30	68
ISWC	9	5.11	11	11	2.09	9
NPD	71	13.60	96	84	5.32	99

Complete redundancy. All attributes of the parent table are replicated to the subtables. In this scenario, the ontology learning systems must recognize and remove this redundancy in order to represent the hierarchy correctly.

Single table. In this scenario, the entire hierarchy is modeled within a single table: Attributes that are included in only some of the subconcepts are left empty for all records of the other subconcepts. Categorical and/or Boolean attributes are used to indicate to which sub-concepts the individual entities belong. Ontology learning systems need to normalize the table correctly to extract the correct hierarchy (see Figure 4).

No parent table. In this scenario, every entity belongs to at least one of the subconcepts. Thus, there is no parent table required and all attributes of the superconcept are replicated in all subtables. The absence of the parent table makes it difficult for the ontology learning system to recognize the hierarchical relationships between the individual tables.

5.2.6 Special. Databases are usually composed of many of the above-described scenarios. Often, these scenarios, such as many-to-many tables or weak entities, are not easily separable but interconnected. Additionally, they might require a mapping that mapping languages are not able to cover due to their language constraints. The micro-benchmark group "Special" includes scenarios that are built by combining two or three of the previous scenarios and are expected to be especially difficult for an ontology learning system. For example, the database schema of a special scenario consists of a many-to-many table, but also includes a table with multiple boolean attributes (see "Boolean attributes" in Section 5.2.2) or requires an SQL-expression.

5.3 Real-World Benchmark

Real-world benchmarks enable an assessment of the behavior of a system in complex and realistic situations. In contrast to synthetic benchmarks, they cover properties and challenges that systems must also be able to master in real use. With Burr, we present four real-world database-ontology pairs (see Table 2). In contrast to the micro-benchmark, these databases have a realistic schema complexity across various complexity dimensions, e.g., varying number of tables (up to 71), columns (up to 46 in one table), foreign keys (up to 96), and normalization degree. To ensure correctness, we relied on existing and widely accepted databases and ontologies for the Mondial, ISWC, and NPD scenarios, which we describe below. The ontology for the enterprise database was constructed in collaboration with a domain expert and a modeling expert.

5.3.1 Mondial Database. The Mondial database is a geographical database, compiled from multiple sources, such as the "CIA World Factbook" [1]. The database was initially presented by May [36] and stores information about countries, political organizations, and societal details. Burr provides a manually created mapping between the relational database and the ground truth ontology of Mondial.

As shown in Table 2, the Mondial database consists of 46 tables with approx. four columns each. However, not every table can be translated directly to a concept, as the ground truth ontology has only 33 concepts. Thus, many tables in the database do not represent a concept, but enhance other concepts or build relationships. In general, many relationships exist across tables with 33 foreign key constraints in total.

Thereby, this mapping includes challenging scenarios that an ontology learning system has to solve. First, the original database does not contain any foreign key constraints. To mitigate this unusual situation, we provide two Mondial database versions – the original version without foreign key constraints and one with all manually added foreign constraints. The comparison of the quality of systems on these two versions can be used to measure the dependency of ontology learning systems on foreign key constraints under real-world conditions. Furthermore, the Mondial scenario includes many challenges that we have already described in the micro-benchmark in Section 5.2. For example, primary keys are often composite; concepts are often extended by dimensional tables, which add additional information as political context; N-to-M tables have additional columns; and some tables and columns include multiple concepts – for example, the column name in the table mountain represents either the concept mountain or volcano, depending on the value in column type. Mondial was chosen for its wide schema breadth with 46 tables. Mondial is also used in the ontology mapping benchmark RODI [36], proposing a task-based evaluation.

5.3.2 Enterprise. Burr provides a real-world enterprise database ontology mapping from SAP SE that was built for integrating specialized data into a unified view. To do so, a modeling expert created a domain model of the department's data based on interviews with domain experts and then created a mapping between the database and the ontology. This database was derived from a real-world enterprise system, which is in use by many companies. It enables a realistic evaluation of the quality of ontology learning systems in the industry. Originally, the database was represented in an Excel file consisting of eight spreadsheets, each with one table. As shown in Table 2, the enterprise database consists of eight large tables with an average of 22.75 columns (up to 46 columns in one table). As this data originates from multiple spreadsheets, no foreign keys exist. Despite consisting of only eight tables, the resulting ontology has 20 concepts. Each concept is described by, on average, 4.3 attributes. Furthermore, 68 relationships connect concepts in the ground truth ontology. This indicates the large number of relationships that are implicitly modeled within and between the eight tables of the database.

While this database does not hold many tables, its complexity lies within each of the tables. Every table is quite denormalized with many columns and numerous hierarchies, relationships, and concepts. Applying a normalization algorithm [43] with an internal algorithm for profiling functional dependencies resulted in 36 tables. In fact, the underlying, well-structured data schema itself, which is used internally by the enterprise system (and is not publicly available), consists of 32 tables and 50 foreign key relationships. Due to its complexity, the denormalized data schema is a good asset for our benchmark, as it poses great challenges for ontology learning systems. Apart from extracting those concepts, hierarchies, and relationships correctly, the enterprise scenario holds the additional challenge of being able to benefit from additional domain knowledge. Systems might be able to retrieve valuable information for the ontology engineering process from external sources and websites, such as the "SAP Enterprise Architecture Framework"-glossary [49].

5.3.3 ISWC Database. The "International Semantic Web Conference" (ISWC) domain is a real-world ontology about conferences and often used in semantic web benchmarks. Originally, the database and its associated mapping to the ISWC-ontology was provided on the D2RQ website as an example by Bizer and Seaborne [9]. The database holds example data about the ISWC conferences from 2002 and 2006. For each conference, it stores published papers, attendees, and conference topics.

As shown in Table 2, the ISWC database has nine tables, with each having around five columns and, on average, one connection to another table. The associated ontology has similar properties that fit the characteristics of the connected database. The domain holds multiple challenges. First, it includes many N-to-M tables. Additionally, the domain holds many conditional mappings, such as different types of organizations, stored in the column type in the organization table that need to be correctly converted to the correct concepts in the ontology.

5.3.4 NPD FactPages. The Norwegian Petroleum Directorate's FactPages [57] holds up-to-date data about petroleum-related activities on the Norwegian Continental Shelf, such as operating companies or geographical locations of wells. This database is operated and used by the Norwegian Offshore Directorate. The authors provide an ontology and a D2RQ-mapping, which we cleaned (e.g., of unavailable files, see GitHub for details) and integrated into our benchmark. This database consists of 71 tables and 84 concepts.

In summary, each benchmark database tests different aspects: For example, the Mondial database tests the capabilities of systems to learn ontologies based on a schema with many tables, while the Enterprise database tests the ability to learn ontologies based on a few but complex tables.

6 Benchmarking Results

To showcase the applicability and value of Burr, we conducted experiments using four diverse ontology learning systems. We analyzed the systems' strengths and weaknesses while also examining the usefulness of Burr. In the following, we first introduce the experimental setup and selected ontology learning systems (Section 6.1). Then, we present the results of executing the systems on Burr and discuss the most interesting results (Section 6.2).

6.1 Experimental Setup

We executed all experiments on a host system with an AMD EPYC 7702P 64-core processor with 504 GB of memory. For our experiments, we used the following ontology learning systems.

W3C-Mapping / Ontop. The "World Wide Web Consortium" (W3C) defines a recommendation [64] for mapping relational data to RDF. It defines rules to construct an ontology, including a mapping between the relational database and the ontology. This rule-based system, which is also used by the well-known virtual knowledge graph system Ontop [14], relies on simple rules. Thus, it maps a table to a concept, database columns to ontology attributes, and foreign key constraints to relationships. For tables without any primary key constraints, W3C recommends representing each row with a blank node to ensure identifiability. We rely on the implementation included in the repository of D2RQ [9]. As stated by Calvanese et al. [10], many modern systems that are still in use, such as Ultrawrap [55], rely on the W3C Direct Mapping standard. Thus, we selected this system as a representative of this family of methods.

RDB2Onto. Ben Mahria et al. [6] presents a recent enhanced rule-based system that includes algorithms across the entire lifecycle, from the ontology construction up to building the actual knowledge graph. We focus our evaluation on the ontology learning algorithm of the system, which we refer to as RDB2Onto. In a first step, RDB2Onto extracts metadata, such as constraints, and analyzes them. For example, it tests whether symmetric relationships exist or whether primary keys are composite keys. Then, based on the analysis's results, the system creates the conceptual ontology using appropriate transformation rules. We extended an existing implementation [7] to output an ontology-database-mapping next to the OWL-ontology based on the mapping rules defined in their paper.

LLM-based models. Deep learning and large language models (LLMs) have disrupted the state-of-the-art in many fields of research. To show how Burr helps understand the capabilities of LLMs for the task of ontology learning, we evaluate various models and prompts on this task. We designed four prompts of varying complexity. During the engineering of the prompts, we were inspired by guidelines introduced by OpenAI on high-quality prompts, such as providing instructions and checklists [33]. Additionally, as prompts can be improved by LLMs themselves [69], we revised our prompts using GPT-o4-mini-high. In particular, we experiment with GPT-4.1 [41], Gemini-2.5 Pro [13], and the open-source model Qwen3-14B [67] using the following four prompts: (1) **Simple**: This prompt asks the model to create a D2RQ-mapping based on a given relational schema and instance data without any additional information. (2) **Enhanced**: The enhanced prompt adds a predefined generic example mapping to the prompt, which should decrease syntax errors and guide the model in creating the output mapping. (3) **Fewshot**: Here, we provide meaningful scenarios from three different (sub)categories as described in the micro-benchmark (one-to-many relationships, many-to-many relationships, and hierarchies). (4) **Checklist**: We add to the fewshot-prompt an additional checklist that the model needs to consider. This list includes checks, such as whether all tables were considered or all prefixes were added to the mapping file.

After the response of the LLM, we apply minor post-processing steps on the response, such as extracting the actual mapping out of the response or cleaning concept and attribute names. Details can be found in our GitHub repository.

OntoGenix. Despite much research on LLMs in the field of ontology learning, no current research system uses LLMs specifically for ontology learning from relational databases. Thus, we extended the LLM-based ontology learning system for CSV-data OntoGenix [59] to work on relational databases. OntoGenix uses the LLM-based chatbot GPT-4.1 to build an ontology in multiple steps based on a CSV-file in a chain-of-thought-inspired setup. To enable the system to work with relational databases, we asked the model to create data descriptions not only from one CSV-file, but from multiple tables. Instead of one data description, OntoGenix generates a description for each table. We automated the current interactive process and added the data description to the final mapping-creation prompt to decrease hallucinations that were observed in preliminary experiments. Furthermore, we adapted the model to output an R2RML-mapping [65], which is optimized for database-ontology mappings, instead of an RML-mapping [20]. We provide mapping templates so that the chatbot can use them to create parseable files.

Evaluation Metrics. To understand the quality of ontology learning systems, we evaluate them using different quality metrics. First, we apply the lexical precision and lexical recall score as defined by Sabou et al. [48] (see Section 3.4). We rely on that score to understand how well those systems can extract names and the general ontology. Preliminary experiments indicated that systems often output different names for the same attributes in the resulting ontology. In some cases, these differences were only marginal: for example, they differed in the selection of characters that connect two words ("_" vs. "-"). Therefore, we apply basic transformations to improve the compatibility of the naming schemes of different systems. Second, we evaluate ontologies using our newly proposed mapping-based metric, as defined in Section 4. This evaluation approach helps understand how well the necessary components of the database are selected by the ontology learning system.

For each metric, we calculate a "concept"-score, an "attribute"-score, and a "relationship"-score. The individual scores represent how well the systems are capable of extracting the respective ontology elements compared to the ground truth mapping.

In summary, the ontology learning evaluation profile consists of *lexical* quality metrics to assess the naming capabilities of the learning systems and *mapping-based* quality metrics to assess the database coverage as well as accuracy of the resulting mappings.

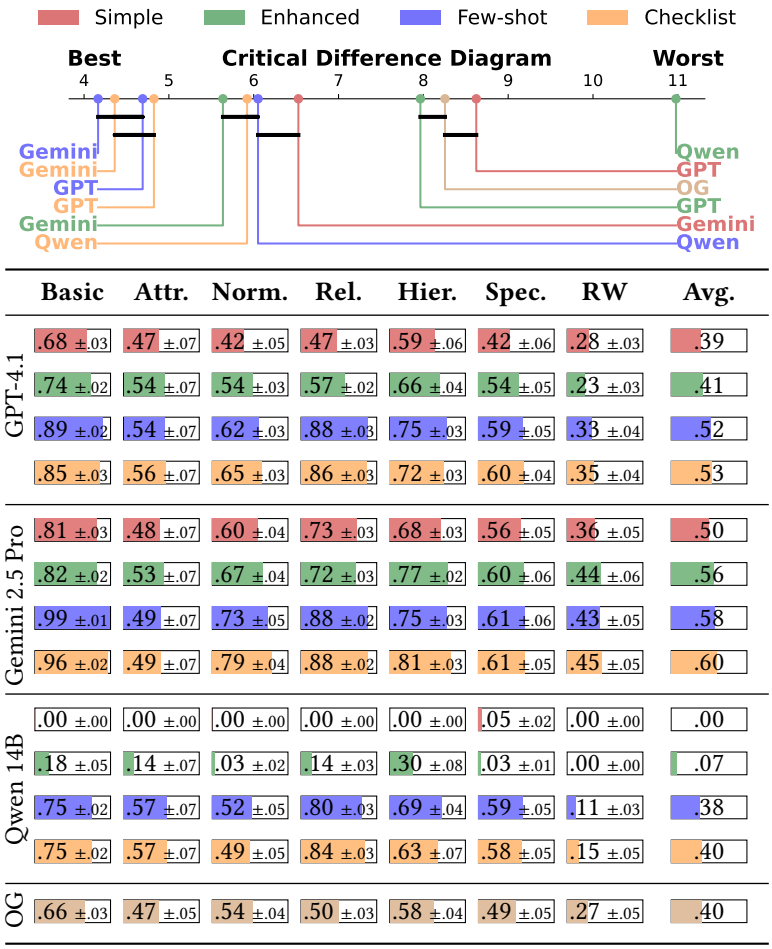


Fig. 5. **Quality overview of LLM-based systems.** The colors shown in the legend on top represent the different prompting strategies. For improved readability, we removed the worst-performing system Qwen-14B with the "Simple"-prompt from the critical difference diagram. Subscripted numbers are the half-width of the 95% confidence interval. OntoGenix is abbreviated as "OG".

For each non-deterministic system, we repeated the same experiment 15 times, resulting in 885 experimental runs per individual system. Then, to test for significance, we applied the Friedman test [26] at a significance level $\alpha = 0.05$. To keep the family-wise error rate at 0.05, we apply the Nemenyi [39] post-hoc test for all pairs of methods.

6.2 Experimental Results

To understand the quality of the presented ontology learning systems, we executed them using Burr. Table 3 represents the results for the two rule-based systems Ontop and RDB2Onto, as well as for the LLM-based systems OntoGenix and Gemini 2.5 Pro with the "Checklist"-prompt. The table provides a detailed quality overview for each system. The first set of rows represents the quality of each real-world scenario, while the latter set represents the quality of each micro-benchmark category. Of the LLM-based family, we selected OntoGenix as it is a widely used system, and

Table 3. **Quality Overview per Scenario and System executed on Burr.** All results are F_1-score. Results are divided into how well concepts (C), relationships (R), and attributes (A) can be extracted. Subscripted numbers are the half-width of the 95% confidence interval.

System	W3C-Mapper / Ontop						RDB2Onto						OntoGenix						Gemini 2.5 Pro					
Metric	Mapping			Lexical			Mapping			Lexical			Mapping			Lexical			Mapping			Lexical		
Element	C	R	A	C	R	A	C	R	A	C	R	A	C	R	A	C	R	A	C	R	A	C	R	A
Real-World Scenarios																								
Mondial	.53	.00	.51	.38	.00	.38	.53	.00	.38	.38	.00	.28	$.29_{\pm.14}$	$.11_{\pm.06}$	$.40_{\pm.20}$	$.27_{\pm.13}$	$.04_{\pm.03}$	$.32_{\pm.16}$	$.61_{\pm.04}$	$.46_{\pm.04}$	$.81_{\pm.03}$	$.59_{\pm.01}$	$.19_{\pm.04}$	$.67_{\pm.03}$
Mondial-FK	.53	.35	.51	.38	.00	.38	.49	.24	.48	.39	.02	.35	$.25_{\pm.12}$	$.06_{\pm.04}$	$.39_{\pm.20}$	$.21_{\pm.11}$	$.01_{\pm.01}$	$.25_{\pm.16}$	$.68_{\pm.04}$	$.50_{\pm.04}$	$.84_{\pm.05}$	$.58_{\pm.02}$	$.21_{\pm.04}$	$.65_{\pm.03}$
Enterprise	.00	.00	.51	.00	.00	.00	.00	.00	.50	.00	.00	.00	$.12_{\pm.03}$	$.07_{\pm.02}$	$.38_{\pm.09}$	$.31_{\pm.08}$	$.00_{\pm.00}$	$.02_{\pm.01}$	$.02_{\pm.02}$	$.10_{\pm.02}$	$.34_{\pm.02}$	$.44_{\pm.02}$	$.00_{\pm.00}$	$.29_{\pm.02}$
ISWC	.53	.11	.39	.00	.00	.00	.67	.62	.44	.00	.00	.00	$.58_{\pm.07}$	$.35_{\pm.11}$	$.44_{\pm.01}$	$.60_{\pm.06}$	$.02_{\pm.02}$	$.14_{\pm.05}$	$.65_{\pm.11}$	$.68_{\pm.05}$	$.56_{\pm.01}$	$.54_{\pm.01}$	$.00_{\pm.00}$	$.30_{\pm.01}$
NPD	.68	.84	.48	.06	.00	.00	.60	.81	.48	.06	.00	.00	$.22_{\pm.08}$	$.13_{\pm.05}$	$.23_{\pm.09}$	$.05_{\pm.03}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.30_{\pm.09}$	$.16_{\pm.04}$	$.08_{\pm.02}$	$.12_{\pm.02}$	$.00_{\pm.00}$	$.00_{\pm.00}$
Micro-Benchmark																								
Basic	.83	.83	.28	1	.67	.23	.83	.83	.28	1	.67	.23	$.85_{\pm.04}$	$.83_{\pm.06}$	$.30_{\pm.06}$	$.89_{\pm.03}$	$.56_{\pm.07}$	$.11_{\pm.03}$	$.99_{\pm.01}$	$.91_{\pm.04}$	$.99_{\pm.02}$	$.99_{\pm.01}$	$.66_{\pm.07}$	$.93_{\pm.02}$
Attributes	.62	.57	.42	.79	.29	.25	.62	.57	.45	.79	.29	.27	$.42_{\pm.07}$	$.45_{\pm.08}$	$.53_{\pm.06}$	$.77_{\pm.05}$	$.27_{\pm.09}$	$.11_{\pm.04}$	$.61_{\pm.07}$	$.43_{\pm.10}$	$.43_{\pm.07}$	$.80_{\pm.03}$	$.29_{\pm.09}$	$.44_{\pm.06}$
Normalization	.47	.17	.81	.44	.12	.36	.48	.25	.78	.45	.12	.32	$.72_{\pm.05}$	$.16_{\pm.06}$	$.75_{\pm.04}$	$.54_{\pm.05}$	$.12_{\pm.05}$	$.32_{\pm.05}$	$.90_{\pm.04}$	$.59_{\pm.08}$	$.87_{\pm.03}$	$.85_{\pm.03}$	$.33_{\pm.07}$	$.67_{\pm.05}$
Relationships	.77	.18	.55	.77	.00	.49	.85	.60	.62	.85	.00	.54	$.72_{\pm.04}$	$.25_{\pm.05}$	$.53_{\pm.03}$	$.55_{\pm.05}$	$.08_{\pm.03}$	$.08_{\pm.02}$	$.95_{\pm.02}$	$.71_{\pm.06}$	$.99_{\pm.00}$	$.98_{\pm.01}$	$.50_{\pm.06}$	$.91_{\pm.03}$
Hierarchies	.41	.60	.71	.83	.60	.51	.41	1	.71	.83	1	.51	$.39_{\pm.03}$	$.65_{\pm.11}$	$.70_{\pm.02}$	$.67_{\pm.05}$	$.65_{\pm.11}$	$.30_{\pm.04}$	$.48_{\pm.08}$	$1_{\pm.00}$	$.97_{\pm.01}$	$.96_{\pm.02}$	$1_{\pm.00}$	$.78_{\pm.06}$
Special	.61	.56	.46	.68	.14	.33	.62	.59	.49	.70	.14	.35	$.56_{\pm.06}$	$.45_{\pm.08}$	$.46_{\pm.06}$	$.45_{\pm.07}$	$.14_{\pm.06}$	$.13_{\pm.03}$	$.68_{\pm.05}$	$.57_{\pm.08}$	$.57_{\pm.06}$	$.82_{\pm.03}$	$.18_{\pm.06}$	$.44_{\pm.05}$
Average:	.54	.37	.51	.46	.15	.26	.55	.49	.51	.47	.19	.25	.45	.30	.46	.47	.16	.16	.61	.54	.66	.68	.29	.54

Gemini 2.5 Pro as it is the best-performing LLM-based model for the task of ontology learning. To obtain a statistically robust estimate of the average quality, we repeatedly resampled with replacement from the experimental result set (10,000 times).

Next to this detailed quality profile for these four systems, Figure 5 illustrates the results for each evaluated LLM with the corresponding prompts. The top shows a critical-difference diagram [17], which represents a ranking of the methods with the left side being best. A line connects systems if their quality difference is not statistically significant. The average quality is aligned with the averages in Table 3, where each real-world scenario is handled as its own category. However, we combined these scenarios for visual clarity in Figure 5.

In the following, we first discuss the most interesting and surprising results of the ontology learning systems executed on the micro-benchmark, including an overview of the engineered prompt-LLM variations. Then, we analyze the experiments conducted within the real-world scenarios. Finally, we summarize the takeaways and discuss the applicability of Burr.

6.2.1 Micro-Benchmark. On average, the general-purpose language model Gemini 2.5 Pro achieves the highest mapping quality and lexical quality for the micro-benchmark scenarios. The rule-based ontology learning systems RDB2Onto and Ontop have a similar performance. However, they cannot keep up with the LLM-based model in any dimension. The traditional systems and Gemini 2.5 Pro benefit from a lexical evaluation when extracting concepts (see "C"), as the lexical quality is higher for most experiments compared to the mapping-based score. This result contrasts with the quality of extracting relationships (see "R") and attributes (see "A"). The systems achieve a higher quality if their mappings are evaluated and not the sole naming of the ontology elements. One reason for this effect could be that the names of relationships and attributes still differ in small details despite our basic transformations. However, the ground truth mapping and learned mapping both continue to refer to the correct attribute in the database, indicating that the database elements were correctly identified.

As shown in Figure 5, the reasoning model Gemini 2.5 Pro works best across all LLMs. The best performing prompt is the "Checklist"-prompt, although its results are not statistically significant compared to the "Fewshot"-prompt. We can conclude that few-shot examples improve modeling quality. For Gemini 2.5 Pro, the quality rises from 0.56 up to 0.60. GPT-4.1 is the second-best performing model with an average of 0.53 in the best configuration. For the categories "Attributes", "Relationships", and "Special", it achieves competitive results compared to Gemini 2.5 Pro. Qwen-14B, which is an open-source LLM that can be executed on consumer-grade hardware, cannot keep up with the quality of the other two LLMs. Especially without any modeling examples, it almost always produces non-parseable output (0.00 with the "Simple"-prompt). However, when including a few use-case specific examples of database-mapping pairs in the prompt, Qwen-14B improves from 0.07 to 0.38. Without these examples, Qwen-14B misses the necessary knowledge to correctly model the respective database concepts. Interestingly, in the "Special"-scenarios, Qwen-14B achieves a quality similar to GPT-4.1 and Gemini 2.5 Pro. However, please note the differences in the number of parameters. While the size of GPT-4.1 and Gemini 2.5 Pro is estimated to be 1.8 trillion [50], Qwen-14B has approximately 128 times fewer parameters and can be executed on a single GPU.

With the different micro-benchmark scenarios, the strengths and weaknesses of the systems become clear. For example, while all systems mostly perform well in encapsulated scenarios, they already struggle when two of those simple scenarios are combined (category "Special"). In these cases, even the best performing model Gemini 2.5 Pro achieves an F_1-score of only 0.61. Extracting the correct relationships and related joins is especially challenging in those scenarios. For example, one scenario requires a different handling of two many-to-many tables within one database. Here,

the system decides on one modeling strategy for such tables and fails to detect the difference between the two many-to-many tables.

A more detailed analysis of the capabilities of RDB2Onto and Ontop suggests that both benefit from normalization. The higher the normalization degree of a database, the better the extraction of concepts. The concept F_1-score increases for RDB2Onto from a denormalized database to a normalized database from 0.50 to 0.75. However, attributes can be better extracted at a lower level of normalization, as the quality of the systems for extracting attributes is highest at first normal form.

OntoGenix and Gemini 2.5 Pro achieve higher lexical quality scores in the "normalization"-scenarios than the two rule-based systems, which was to be expected due to their advanced natural language capabilities. In addition, the LLM-based approaches performed better in extracting concepts and relationships than the rule-based approaches from scenarios where the databases are highly normalized or denormalized.

Burr also helps understand nuances in the behavior of the different systems. For instance, RDB2Onto uses a more sophisticated process of handling N-to-M tables compared to Ontop. The results show that this additional effort is rewarded. While RDB2Onto achieves an averaged mapping-based F_1-score of 0.66, Ontop can only achieve an F_1-score of 0.45.

Takeaways. Burr's micro-benchmark reveals the strengths and weaknesses of the individual ontology learning systems. Gemini 2.5 Pro achieves the best results, but still struggles with the special scenarios. RDB2Onto and Ontop have a similar quality, while OntoGenix achieves worse results in the mapping-score and comparable results in the lexical evaluation compared to the rule-based systems. The difference between the lexical score and the mapping-based score becomes clear: While the lexical score only evaluates how well a system names concepts, relationships, and attributes, the mapping-based score evaluates precisely which database elements are correctly covered. Burr shows that LLMs deliver very good results when modeling ontologies, but only if used carefully, for example, with an appropriate prompt.

In ontology learning, researchers must always consider not only the lexical quality of an ontology learning system but also the system's ability to create mappings. Our evaluation shows that Burr provides the necessary means to analyze ontology learning systems in detail and to compare their advantages and disadvantages with other systems. It supports the evaluation by providing suitable metrics and challenging scenarios.

6.2.2 Real-World Scenarios. In general, all systems perform poorly under real-world conditions. As already shown in the experimental analysis of the micro-benchmark, the systems are not ready to handle more than one modeling scenario in the same database. Adding additional complexity and combining even more of these scenarios into one complex database leads to further quality losses.

Generally, Gemini 2.5 Pro with the "checklist"-prompt is the best-performing model. Similarly to the micro-benchmark, GPT-4.1 achieves the second-best quality. Qwen-14B achieves an averaged quality of 0.15. OntoGenix achieves a worse quality than GPT-4.1 (0.27), although both rely on GPT-4.1 as a base model. Thus, careful prompt design is equally important as LLM selection. Gemini 2.5 Pro again outperforms the rule-based systems as well. However, on the NPD-database, the traditional systems achieve a higher quality than any LLM-based approach. This result highlights the advantage of simple rule-based systems: While LLMs excel on denormalized schemata that require a semantic understanding, rule-based systems are still able to outperform such systems if, for example, large databases generally follow similar patterns as defined in their rules.

Experiments with adding foreign key constraints to the original Mondial database (scenario Mondial-FK) illustrate the effect of foreign keys on rule-based ontology learning systems. The quality in the relationship category of the system Ontop improves the most (from 0.00 to 0.35),

followed by RDB2Onto (from 0.00 to 0.24). Both systems rely on foreign keys for relation extraction. Interestingly, Gemini 2.5 Pro does not benefit a lot from the foreign keys. A reason might be that the prompt length rises with the additional constraints, which, then, hinders further quality improvements.

In the enterprise scenario, all systems are incapable of building a suitable ontology and an associated mapping. This underlines the contribution of this highly denormalized database to the benchmark, as it pushes the systems to their limits. W3CMapper and RDB2Onto do not model any concepts or relationships correctly, but can only extract a few correct attributes. However, they name these attributes wrongly. As shown in the analysis of the experiments executed on the micro-benchmark, the systems do not work reliably in denormalized scenarios. Thus, as the enterprise database mainly consists of such scenarios, they do not perform well on it.

OntoGenix requires a much higher runtime compared to the two rule-based systems and the LLM-based systems. While the latter finish within 11 s, OntoGenix requires up to 45 minutes due to multiple LLM calls.

Takeaways. Burr's real-world scenarios help to understand the performance of ontology learning systems under in-practice conditions. So far, such systems do not work well on the task of creating high-quality ontologies and database-ontology mappings. Differences between the lexical and mapping-based metrics emphasize the necessity of both quality measurements. Especially due to the hallucinations of LLMs, LLM-based approaches require critical examination, as sometimes the lexical F_1-score pretends a higher quality than their actual output mapping has (see concept-score in the enterprise scenario). However, the opposite case can also occur, whereby the lexical score suggests a lower quality than the mapping-based score. Nevertheless, these are often only minor differences in terminology. Burr provides real-world scenarios that allow the exploration of the capabilities of ontology learning systems to create correct results.

7 Conclusion and Outlook

Ontologies and knowledge graphs are important resources in the age of AI: they provide structured, interconnected representations of knowledge, stored in multiple data sources. They enable machines to understand, reason, and make informed decisions about complex information. Many systems exist to automatically build ontologies, bringing the metadata and instance data of individual sources, such as relational databases, into a standardized format. However, these systems have all been evaluated in different ways and no standardized benchmark exists so far.

We introduced the benchmark Burr for ontology learning systems using relational databases as their input, aiming to unify the evaluation process of such systems. Burr includes improved evaluation metrics exploiting the mappings between the database and the ontology. These metrics enable a more precise assessment of the ontology learning systems. Furthermore, Burr provides a *micro-benchmark* including 54 challenging encapsulated scenarios and *four complex real-world scenarios*, which include a large industrial enterprise database. In our experimental evaluation, we assessed the applicability and usefulness of Burr while simultaneously analyzing popular ontology learning systems. Extending Burr to the research area of ontology mapping, i.e., learning the translation between an existing database and an existing ontology, is a promising next step. As ontology mapping is a necessary part of learning ontologies, i.e. when no ontology already exists, such a step could be benchmarked in detail individually.

Acknowledgments

This research was partially funded by SAP SE.

References

[1] Central Intelligence Agency. 2024. The World Factbook. https://www.cia.gov/the-world-factbook/. Accessed: 2024-12-30.

[2] AI Data Analytics Network and Elliot Leavy. 2022. Data Quality Crisis: New Survey Reveals 77% of Organizations Have Quality Issues. https://www.aidataanalytics.network/data-governance/articles/data-quality-crisis-new-survey-reveals-77-of-organizations-have-quality-issues Accessed: 2024-11-27.

[3] Knarig Arabshian and Peter J. Danielsen. 2011. Semi-automated Ontology Creation for High-Level Service Classification. In *Proceedings of the International Conference on Semantics Knowledge and Grid (SKG)*. IEEE, 17–20. https://doi.org/10.1109/SKG.2011.39

[4] Hamed Babaei Giglou, Jennifer D'Souza, and Sören Auer. 2023. LLMs4OL: Large Language Models for Ontology Learning. In *Proceedings of the International Semantic Web Conference (ISWC)*. Springer, 408–427.

[5] Carlo Batini, Maurizio Lenzerini, and Shamkant B. Navathe. 1986. A Comparative Analysis of Methodologies for Database Schema Integration. *Comput. Surveys* 18, 4 (1986), 323–364. https://doi.org/10.1145/27633.27634

[6] Bilal Ben Mahria, Ilham Chaker, and Azeddine Zahi. 2021. A novel approach for learning ontology from relational database: from the construction to the evaluation. *Journal of Big Data* 8, 1 (2021), 1–25. https://doi.org/10.1186/s40537-021-00412-2

[7] Bilal Benmammar. 2024. *FromRDBtoOnto: A framework for transforming relational database schemas into ontologies*. https://github.com/bilalbenma/FromRDBtoOnto Accessed: 2024-12-29.

[8] Johann Birnick, Thomas Bläsius, Tobias Friedrich, Felix Naumann, Thorsten Papenbrock, and Martin Schirneck. 2020. Hitting Set Enumeration with Partial Information for Unique Column Combination Discovery. *Proceedings of the VLDB Endowment (PVLDB)* 13, 11 (2020), 2270–2283. http://www.vldb.org/pvldb/vol13/p2270-birnick.pdf

[9] Christian Bizer and Andy Seaborne. 2005. D2RQ—Treating non-RDF databases as virtual RDF graphs. *World Wide Web Internet and Web Information Systems* (01 2005).

[10] Diego Calvanese, Avigdor Gal, Davide Lanti, Marco Montali, Alessandro Mosca, and Roee Shraga. 2023. Conceptually-grounded mapping patterns for Virtual Knowledge Graphs. *Data and Knowledge Engineering (DKE)* 145 (2023), 1–26. https://doi.org/10.1016/j.datak.2023.102157

[11] E. F. Codd. 1970. A Relational Model of Data for Large Shared Data Banks. *Commun. ACM* 13, 6 (1970), 377–387. https://doi.org/10.1145/362384.362685

[12] E. F. Codd. 1971. Further Normalization of the Data Base Relational Model. *Research Report / RJ / IBM / San Jose, California* RJ909 (1971).

[13] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit S. Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and further authors. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *CoRR* abs/2507.06261 (2025). https://doi.org/10.48550/ARXIV.2507.06261 arXiv:2507.06261

[14] Óscar Corcho, Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. 2017. Ontop: Answering SPARQL queries over relational databases. *Semantic Web* 8, 3 (2017), 471−−487. https://doi.org/10.3233/SW-160217

[15] Wei Dai and Daniel Berleant. 2019. Benchmarking Contemporary Deep Learning Hardware and Frameworks: A Survey of Qualitative Metrics. In *Proceedings of the IEEE International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE, 148–155. https://doi.org/10.1109/COGMI48466.2019.00029

[16] Luciano Frontino de Medeiros, Freddy Priyatna, and Óscar Corcho. 2015. MIRROR: Automatic R2RML Mapping Generation from Relational Databases. In *Proceedings of the International Conference on Web Engineering (ICWE)*, Vol. 9114. Springer, 326–343. https://doi.org/10.1007/978-3-319-19890-3_21

[17] Janez Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research (JMLR)* 7 (Dec. 2006), 1—-30.

[18] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: Table Understanding through Representation Learning. *Proceedings of the VLDB Endowment (PVLDB)* 14, 3 (2020), 307–319. https://doi.org/10.5555/3430915.3442430

[19] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. 2014. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In *Proceedings of the Workshop on Linked Data on the Web co-located with the International World Wide Web Conference (LDOW@WWW) (CEUR Workshop Proceedings)*, Vol. 1184. https://ceur-ws.org/Vol-1184/ldow2014_paper_01.pdf

[20] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. 2014. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In *Proceedings of the Workshop on Linked Data on the Web co-located with the International World Wide Web Conference (LDOW@WWW) (CEUR Workshop Proceedings)*, Vol. 1184. CEUR-WS.org. https://ceur-ws.org/Vol-1184/ldow2014_paper_01.pdf

[21] Euthymios Drymonas, Kalliopi Zervanou, and Euripides G. M. Petrakis. 2010. Unsupervised Ontology Acquisition from Plain Texts: The *OntoGain* System. In *Proceedings of the International Conference on Applications of Natural Language*

to Information Systems (NLDB), Vol. 6177. Springer, 277–287. https://doi.org/10.1007/978-3-642-13881-2_29
[22] Erki Eessaar. 2016. The Database Normalization Theory and the Theory of Normalized Systems: Finding a Common Ground. *Baltic Journal of Modern Computing* 4, 1 (2016).
[23] Ramez Elmasri and Shamkant B. Navathe. 2010. *Fundamentals of Database Systems, 6th Edition.* Addison-Wesley-Longman.
[24] Jérôme Euzenat and Pavel Shvaiko. 2013. *Ontology Matching* (2nd ed.). Springer, New York, Chapter 2, 25–54.
[25] Jérôme Euzenat and Pavel Shvaiko. 2013. *Ontology Matching* (2nd ed.). Springer, New York, Chapter 7, 149–197.
[26] Milton Friedman. 1937. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American statistical association* 32, 200 (1937), 675–701. http://www.jstor.org/stable/2279372
[27] Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. 2008. *Database Systems: The Complete Book* (2 ed.). Prentice Hall Press, USA.
[28] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2022. Knowledge Graphs. *Comput. Surveys* 54, 4 (2022), 71:1–71:37. https://doi.org/10.1145/3447772
[29] Ernesto Jiménez-Ruiz, Evgeny Kharlamov, Dmitriy Zheleznyakov, Ian Horrocks, Christoph Pinkel, Martin G. Skjæveland, Evgenij Thorstensen, and Jose Mora. 2015. BootOX: Practical Mapping of RDBs to OWL 2. In *Proceedings of the International Semantic Web Conference (ISWC)*, Vol. 9367. Springer, 113–132. https://doi.org/10.1007/978-3-319-25010-6_7
[30] Elem Güzel Kalaycı, Irlan Grangel González, Felix Lösch, Guohui Xiao, Anees ul Mehdi, Evgeny Kharlamov, and Diego Calvanese. 2020. Semantic Integration of Bosch Manufacturing Data Using Virtual Knowledge Graphs. In *Proceedings of the International Semantic Web Conference (ISWC)*. Springer, 464—-481. https://doi.org/10.1007/978-3-030-62466-8_29
[31] Evgeny Kharlamov, Ernesto Jiménez-Ruiz, Dmitriy Zheleznyakov, Dimitris Bilidas, Martin Giese, Peter Haase, Ian Horrocks, Herald Kllapi, Manolis Koubarakis, Özgür Özçep, Mariano Rodríguez-Muro, Riccardo Rosati, Michael Schmidt, Rudolf Schlatte, Ahmet Soylu, and Arild Waaler. 2013. Optique: Towards OBDA Systems for Industry. In *The Semantic Web: ESWC Satellite Events*. Springer, 125–140.
[32] Jixiong Liu, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin. 2023. From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. *Journal of Web Semantics* 76 (2023), 1–23. https://doi.org/10.1016/j.websem.2022.100761
[33] Noah MacCallum and Julian Lee. 2025. GPT-4.1 Prompting Guide. OpenAI Cookbook. https://cookbook.openai.com/examples/gpt4-1_prompting_guide.
[34] Alexander Maedche. 2002. *The TEXT-TO-ONTO Environment.* Springer US, 151–170. https://doi.org/10.1007/978-1-4615-0925-7_7
[35] Alexander Maedche and Steffen Staab. 2001. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* 16, 2 (2001), 72–79. https://doi.org/10.1109/5254.920602
[36] Wolfgang May. 1999. *Information extraction and integration with Florid: The Mondial case study.* Technical Report.
[37] Sedir Mohammed, Lukas Budach, Moritz Feuerpfeil, Nina Ihde, Andrea Nathansen, Nele Sina Noack, Hendrik Patzlaff, Felix Naumann, and Hazar Harmouch. 2025. The effects of data quality on machine learning performance on tabular data. *Information Systems (IS)* 132 (2025), 1–18. https://doi.org/10.1016/J.IS.2025.102549
[38] Rosalba Mosca, Massimo De Santo, and Rosario Gaeta. 2023. Ontology learning from relational database: a review. *Journal of Ambient Intelligence and Humanized Computing (AIHC)* 14, 12 (01 Dec 2023), 16841–16851. https://doi.org/10.1007/s12652-023-04693-8
[39] Peter Björn Nemenyi. 1963. *Distribution-free Multiple Comparisons.* Ph.D. Dissertation. Princeton University.
[40] OpenAI. 2024. GPT-4o Technical Report. https://openai.com/research/gpt-4o Accessed: 2025-03-31.
[41] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and further authors. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023). https://doi.org/10.48550/ARXIV.2303.08774 arXiv:2303.08774
[42] Jinie Pak and Lina Zhou. 2009. A Framework for Ontology Evaluation. In *Proceedings of the Workshop on e-Business (WEB) (Lecture Notes in Business Information Processing)*, Vol. 52. 10–18. https://doi.org/10.1007/978-3-642-17449-0_2
[43] Thorsten Papenbrock and Felix Naumann. 2017. Data-driven Schema Normalization. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*. OpenProceedings.org, 342–353. https://doi.org/10.5441/002/EDBT.2017.31
[44] Christoph Pinkel, Carsten Binnig, Ernesto Jiménez-Ruiz, Wolfgang May, Dominique Ritze, Martin G. Skjæveland, Alessandro Solimando, and Evgeny Kharlamov. 2015. RODI: A Benchmark for Automatic Mapping Generation in Relational-to-Ontology Data Integration. In *Proceedings of the European Semantic Web Conference (ESWC)*. Springer, 21––37. https://doi.org/10.1007/978-3-319-18818-8_2
[45] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. 2014. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. *International Journal on Semantic Web and Information Systems*

(IJSWIS) 10, 2 (2014), 7––34. https://doi.org/10.4018/ijswis.2014040102

[46] PricewaterhouseCoopers. 2017. Sizing the Prize: What's the Real Value of AI for Your Business and How Can You Capitalise? https://preview.thenewsmarket.com/Previews/PWC/DocumentAssets/476830.pdf

[47] Dominique Ritze, Oliver Lehmberg, and Christian Bizer. 2015. Matching HTML Tables to DBpedia. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS)*. Association for Computing Machinery. https://doi.org/10.1145/2797115.2797118

[48] Marta Sabou, Chris Wroe, Carole A. Goble, and Gilad Mishne. 2005. Learning domain ontologies for Web service descriptions: an experiment in bioinformatics. In *Proceedings of the International World Wide Web Conference (WWW)*. Association for Computing Machinery, 190–198. https://doi.org/10.1145/1060745.1060776

[49] SAP SE. 2024. *SAP Enterprise Architecture Framework*. https://help.sap.com/docs/SAP_ENTERPRISE_ARCHITECTURE_FRAMEWORK Accessed: 2024-11-25.

[50] Maximilian Schreiner. 2023. GPT-4 architecture, datasets, costs and more leaked. The Decoder (online). https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked/

[51] Juan Sequeda, Dean Allemang, and Bryon Jacob. 2024. A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model's Accuracy for Question Answering on Enterprise SQL Databases. In *Proceedings of the Joint Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*. Association for Computing Machinery. https://doi.org/10.1145/3661304.3661901

[52] Juan Sequeda and Ora Lassila. 2021. *Designing and Building Enterprise Knowledge Graphs*. Morgan & Claypool Publishers. https://doi.org/10.2200/S01105ED1V01Y202105DSK020

[53] Juan Sequeda, Freddy Priyatna, and Boris Villazón-Terrazas. 2012. Relational database to RDF mapping patterns. In *Proceedings of the Workshop on Ontology Patterns co-located with the International Semantic Web Conference (WOP@ISWC)*. CEUR-WS.org, 97—-108.

[54] Juan Sequeda, Syed Hamid Tirmizi, Óscar Corcho, and Daniel P. Miranker. 2011. Survey of directly mapping SQL databases to the Semantic Web. *The Knowledge Engineering Review* 26 (2011), 445–486. Issue 4. https://www.cambridge.org/core/journals/knowledge-engineering-review/article/survey-of-directly-mapping-sql-databases-to-the-semantic-web/0688CCA9A831376C4EE5DCE09382563B

[55] Juan F. Sequeda and Daniel P. Miranker. 2015. Ultrawrap Mapper: A Semi-Automatic Relational Database to RDF (RDB2RDF) Mapping Tool. In *Proceedings of the ISWC Posters & Demonstrations Track co-located with the International Semantic Web Conference (ISWC) (CEUR Workshop Proceedings)*, Vol. 1486. http://ceur-ws.org/Vol-1486/paper_105.pdf

[56] Guohua Shen, Zhiqiu Huang, Xiaodong Zhu, and Xiaofei Zhao. 2006. Research on the Rules of Mapping from Relational Model to OWL. In *Proceedings of the Workshop on OWL: Experiences and Directions (OWLED)*. https://ceur-ws.org/Vol-216/submission_5.pdf

[57] Martin G. Skjæveland, Espen H. Lian, and Ian Horrocks. 2013. Publishing the Norwegian Petroleum Directorate's FactPages as Semantic Web Data. In *Proceedings of the International Semantic Web Conference (ISWC)*. Springer, 162––177. https://doi.org/10.1007/978-3-642-41338-4_11

[58] Samir Tartir, I. Budak Arpinar, Michael Moore, Amit P. Sheth, and Boanerges Aleman-Meza. 2005. OntoQA: Metric-Based Ontology Quality Analysis. In *Proceedings of IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*.

[59] Mikel Val Calvo, Mikel Egaña Aranguren, and Jesualdo Tomas Fernandez Breis. 2024. *OntoGenix*. https://github.com/tecnomod-um/OntoGenix

[60] Gilles Vandewiele, Bram Steenwinckel, Filip De Turck, and Femke Ongenae. 2019. CVS2KG: Transforming Tabular Data into Semantic Knowledge. In *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching co-located with the International Semantic Web Conference (SemTab@ISWC) (CEUR Workshop Proceedings)*, Vol. 2553. 33–40. https://ceur-ws.org/Vol-2553/paper5.pdf

[61] Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Qili Zhu. 2012. Understanding Tables on the Web. In *Proceedings of the International Conference on Conceptual Modeling (ER)*, Vol. 7532. Springer, 141–155. https://doi.org/10.1007/978-3-642-34002-4_11

[62] Shaobo Wang, Yi Zeng, and Ning Zhong. 2011. Ontology Extraction and Integration from Semi-structured Data. In *Proceedings of the International Conference on Active Media Technology (AMT)*. Springer, 39–48.

[63] World Wide Web Consortium. 2012. OWL 2 Web Ontology Language Document Overview (Second Edition). https://www.w3.org/TR/owl2-syntax/.

[64] World Wide Web Consortium (W3C). 2012. A Direct Mapping of Relational Data to RDF . World Wide Web Consortium Recommendation. https://www.w3.org/TR/rdb-direct-mapping/ Accessed: 2025-07-26.

[65] World Wide Web Consortium (W3C). 2012. R2RML: RDB to RDF Mapping Language. World Wide Web Consortium Recommendation. https://www.w3.org/TR/r2rml/ Accessed: 2024-11-26.

[66] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. 2018. Ontology-based data access: a survey. In *Proceedings of the International Joint Conference on*

Artificial Intelligence (IJCAI). AAAI Press, 5511—-5519.

[67] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and further authors. 2025. Qwen3 Technical Report. *CoRR* abs/2505.09388 (2025). https://doi.org/10.48550/ARXIV.2505.09388 arXiv:2505.09388

[68] Amrapali Zaveri, Dimitris Kontokostas, Sebastian Hellmann, Jürgen Umbrich, Christoph Pinkel, Carsten Binnig, Ernesto Jiménez-Ruiz, Evgeny Kharlamov, Wolfgang May, Andriy Nikolov, Ana Sasa Bastinos, Martin G. Skjæveland, Alessandro Solimando, Mohsen Taheriyan, Christian Heupel, Ian Horrocks, Amrapali Zaveri, Dimitris Kontokostas, Sebastian Hellmann, and Jürgen Umbrich. 2018. RODI: Benchmarking relational-to-ontology mapping generation quality. *Semantic Web* 9, 1 (2018), 25—-52. https://doi.org/10.3233/SW-170268

[69] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large Language Models are Human-Level Prompt Engineers. In *Proceedings of the International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=92gvk82DE-

Received April 2025; revised July 2025; accepted August 2025