Web Semantics: Science, Services and Agents on the World Wide Web 85 (2025) 100858

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Knowledge Graphs as a source of trust for LLM-powered enterprise question answering

Juan Sequeda *, Dean Allemang, Bryon Jacob

data.world AI Lab, Austin, TX, USA

ARTICLE INFO

Keywords:
Knowledge Graph
LLM
Large Language Model
Generative AI
Question answering
Knowledge engineering
SPARQL
SQL
OWL
R2RML

ABSTRACT

Generative AI provides an innovative and exciting way to manage knowledge and data at any scale; for small projects, at the enterprise level, and even at a world wide web scale. It is tempting to think that Generative AI has made other knowledge-based technologies obsolete; that anything we wanted to do with knowledge-based systems, Knowledge Graphs or even expert systems can instead be done with Generative AI. Our position is counter to that conclusion.

Our practical experience on implementing enterprise question answering systems using Generative AI has shown that Knowledge Graphs support this infrastructure in multiple ways: they provide a formal framework to evaluate the validity of a query generated by an LLM, serve as a foundation for explaining results, and offer access to governed and trusted data. In this position paper, we share our experience, present industry needs, and outline the opportunities for future research contributions.

1. Introduction

Question answering, the ability to interact with data using natural language questions and obtaining accurate results, has been a long-standing challenge in computer science for over half a century [1–4]. The field has advanced throughout the past decades [5–7], through Text-to-SQL approaches, as a means of facilitating chatting with the data that is stored in SQL databases [8–13]. With the rise of Generative AI and Large Language Models (LLMs) in early 2023, the interest increased dramatically [14]. These question answering systems that enable users to *chat with your structured data* hold tremendous potential for transforming the way self service and data-driven decision making is executed within enterprises.

Self service and data-driven decision making in organizations today is largely made through Business Intelligence (BI) and analytics reporting. Data teams gather the original data, integrate the data, build a SQL data warehouse (i.e. star schemas), and create BI dashboards and reports that are then used by business users and analysts to answer specific questions (i.e. metrics, KPIs) and make decisions. The bottleneck of this approach is that business users are only able to answer questions given the views of existing dashboards. If a new question is required and it is not answerable through a report, then a new report needs to be created or an existing report needs to be extended. Furthermore, the data may exist in the warehouse, but if a dashboard does not exist to access that data, then business users cannot use the data. Thus, self service is only possible within the current set of dashboards, even if the data is currently available. Ultimately, business users and executives want an expert that understands their business and is available at all times in order to ask questions and receive trustworthy answers.

Large Language Models (LLMs) have become prominent tools for driving chat-based experiences, offering conversational responses based on natural language inputs. However, this approach has notable limitations, particularly when the goal is to provide reliable and explainable answers.

In a first scenario, a user asks a question through a chat interface, which in turn consults an LLM which may be trained or fine tuned with enterprise data, and possibly access a vectorized store in a RAG approach, to generate a direct response. This setup has the following issues:

- Potential Hallucination: The LLM might fabricate information, providing a response that sounds plausible but is factually incorrect.
- Lack of Verification: There is no mechanism to verify whether the answer is correct or even if the model has the necessary knowledge to answer the question accurately.
- Opaque Origins: The user cannot determine where the answer came from or why it should be trusted, making it impossible to explain or validate the response.

* Corresponding author.
E-mail addresses: juan@data.world (J. Sequeda), dean.allemang@data.world (D. Allemang), bryon@data.world (B. Jacob).

https://doi.org/10.1016/j.websem.2024.100858
Received 25 November 2024; Received in revised form 20 December 2024; Accepted 23 December 2024
Available online 15 January 2025

In the second scenario, the LLM transforms the user's question into a structured query, which is then executed against a governed database or Knowledge Graph. This setup introduces several improvements:

- Readable and Verifiable Queries: Unlike an opaque response, a generated query can be reviewed, assessed for correctness, and matched against a schema/ontology.
- Potential for Explanation: The generated query allows for a transparent examination of the relationships it attempts to address, providing a way to explain how the LLM interpreted the question and why the given answer is correct.
- Governed Data Source: The response comes from curated data, which means the accuracy can be evaluated, and the data source can be referenced directly.

Position statement. Our position is that Knowledge Graphs continue to be useful and relevant even after the advent of generative AI because they are a source of trust. Our definition of trust is based on three main aspects: (1) LLMs hallucinate, therefore we need to ensure *accuracy* of responses, (2) LLMs have a black box nature, therefore we need to provide an *explanation* of where answers come from and (3) there is a risk of incorrect information being used, thus we need to ensure *governance*. Knowledge Graphs provide accuracy, explanation and governance. We will even go so far as to say that knowledge-based systems are even more important in the presence of generative AI than ever before.

In what follows, we first summarize our research which supports our position, focusing initially on accuracy. We then share our lessons learned and finish with future research considerations.

2. Knowledge graphs to increase accuracy of LLM-powered question answering on SQL databases

Knowledge Graphs (KGs) have been identified as a promising solution to fill the business context gaps in order to reduce hallucinations, thus enhancing the accuracy of LLMs [15]. From an industry perspective, Gartner stated in July 2023 that, *"Knowledge graphs provide the perfect complement to LLM-based solutions where high thresholds of accuracy and correctness need to be attained."*.[1]

Our hypothesis is that Knowledge Graphs play a critical role in LLM powered Question Answering systems on SQL databases. However, at the time that we started this research in July 2023, it was not clear to what extent. The starting point of our work is to understand the role of Knowledge Graphs for accuracy, given that hallucinations became one of the largest concerns in the industry.

Our first contribution is a benchmark, with data and knowledge in the insurance domain, and experimental results providing evidence that an GPT-4 powered question answering system using a zero-shot prompt that answers enterprise natural language questions over a Knowledge Graph representation of an enterprise SQL database returns 3x more accurate results compared to using only the SQL database schema [16]. Our follow up research demonstrates that leveraging an ontology to identify semantic errors in a SPARQL query generated by an LLM and subsequently use the LLM to repair queries, results in a total of 4x accuracy improvement compared to not using a Knowledge Graph at all [17,18]. Fig. 1 depicts an overview of our approach.

These research results have been productized and are part of the data.world AI Context Engine[2] (AICE) product, an agent-based and API-driven system that connects our customer's data and metadata with Generative AI to power trusted conversations with structured data and metadata.

[1] Adopt a Data Semantics Approach to Drive Business Value", Gartner Report by Guido De Simoni, Robert Thanaraj, Henry Cook, July 28, 2023.

[2] https://data.world/ai/

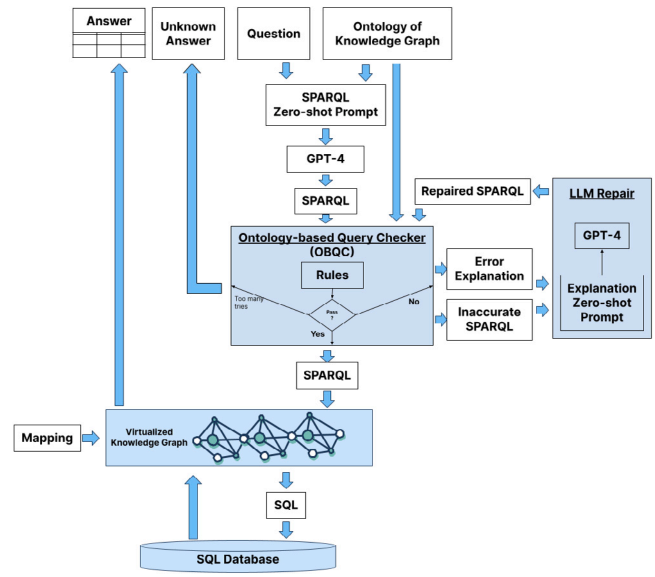

Fig. 1. Overview of our approach.

Creating a well-defined, structured understanding of data, in the form of a Knowledge Graph, not only enhances data management overall and enables consistent interpretation across different applications and reports, but is a foundation to improve the accuracy of LLMs in the enterprise. Our work has influenced the wider data industry to acknowledge the need to invest in semantics and Knowledge Graphs in this new AI era. The benchmark and the results were first independently reproduced and validated by dbt Labs.[3] Several semantic layer vendors have further validated our results.[4,5,6,7,8,9] The GraphRAG Manifesto by Neo4j argues that one of the benefits of GraphRAG relative to vector-only RAG is due to higher accurate responses, citing our benchmark and results.[10]

3. Lessons learned

3.1. Knowledge engineering is key for accuracy

We have come to understand that knowledge engineering, though not commonly emphasized in practice, forms a latent part of various roles across an organization. Anecdotally, we observe that this work is often sporadic and unsystematic, hindering consistent data understanding. The knowledge engineering work occurs among various roles (i.e. data engineers, data stewards, analytics engineers, data analysts), imposing their own assumptions which can lead to inconsistency.

For example, the definition of a policy may vary across different departments. Is a policy with an expired status still considered a policy? Or is a policy only one that has a status of active? Furthermore, there may exist a table named policy, but that does not mean that every row

[3] https://roundup.getdbt.com/p/semantic-layer-as-the-data-interface

[4] https://www.atscale.com/blog/semantic-layers-make-genai-more-accurate/

[5] https://www.wisecube.ai/blog/optimizing-llm-precision-with-knowledge-graph-based-natural-language-qa-systems/

[6] https://blog.kuzudb.com/post/llms-graphs-part-1/

[7] https://delphihq.substack.com/p/delphi-at-100-dbt-semantic-layer

[8] https://cube.dev/blog/semantic-layers-the-missing-piece-for-ai-enabled-analytics

[9] https://www.stratio.com/blog/stratio-business-semantic-data-layer-delivers-99-answer-accuracy-for-llms/

[10] https://neo4j.com/blog/graphrag-manifesto/

in that table represents a policy. The policy table has a status column with various codes. Which are the codes that represent an active status? A developer may not have all the answers to these questions and ends up codifying the meaning on the basis of their own assumptions.

This brings us to an important realization: this work of knowledge engineering – figuring out what the data truly means – is critical and should ideally be a standard practice in organizations.

Another lesson learned, and arguably a controversial one: the ontology should be defined in a way that is conducive to the success of the LLM. Often, the "correct" ontology might confuse the LLM resulting in incorrect SPARQL queries being generated. For example, consider modeling *a customer has an address that has a city*. A first approach is to model the following:

```
:Customer a owl:Class.
:Address a owl:Class.
:hasAddress a owl:ObjectProperty;
  rdfs:domain :Customer;
  rdfs:range :Address.

:city a owl:DatatypeProperty;
  rdfs:domain :Address.
```

A second approach is the following:

```
:Customer a owl:Class.
:city a owl:DatatypeProperty;
  rdfs:domain :Customer.
```

Given the question *Return all Customers in the city Austin*, for the first approach, the LLM would need to generate the following SPARQL query:

```
SELECT *
WHERE {
 ?c a :Customer;
   :hasAddress ?a.
 ?a a :Address;
   :city "Austin".
}
```

while for the second approach it would have to generate:

```
SELECT *
WHERE {
 ?c a :Customer;
   :city "Austin".
}
```

It is important to understand under what type of modeling approach is the LLM generating queries that are conducive to higher accuracy. The first approach follows best practices, but what if the LLMs consistently generates the correct query given the ontology in the second approach? Thus, one approach is to adapt the ontology to align with how the LLM responds, all for the sake of increasing accuracy. We learned this through extensive testing (see below) and refining of the ontology based on the LLM's outputs.

3.2. Explainability

Our system, being agent-based, performs multiple steps to generate an answer. We observe anecdotally that showing each step of the process, including any auto-corrections it performs, increases the trust for the user. This transparency allows users to understand how the answer is derived. Fig. 2 provides an example of the thought process from our system

We are providing further explainability by presenting back to users the generated SPARQL query, the parts of the ontology and mappings referenced, the specific business terms applied, and the resulting generated SQL query. All these elements contribute to building trust. This level of explainability allows technical users to verify each part of the process and increases confidence in the final answer.

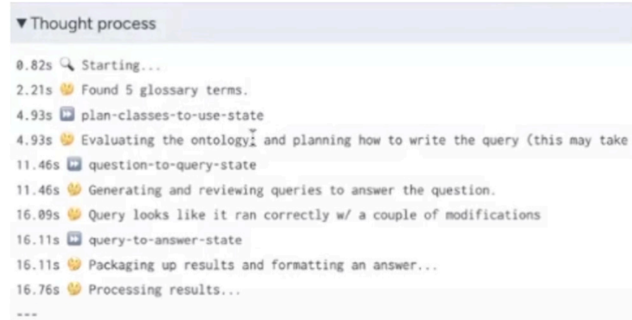

Fig. 2. Example of thought process.

3.3. Governance

Governance involves ensuring that the terms used in the system align with the organization's business glossary, which is maintained by data stewards. These terms and definitions are passed as additional context during processing. Governance is also crucial in defining specific metrics, providing consistency, and ensuring that the terminology used in querying aligns with approved business definitions. We anecdotally observe that users follow the business glossary terms searching for more details about specific concepts and metrics definitions.

For example, the metric *Loss Ratio* is defined as the sum of expense payment and loss payments, divided by the premium. This definition may live in a Business Glossary term and governed by a data steward. Therefore when a question regarding a Loss Ratio is asked, the content of the governed business glossary can be passed to the LLM as additional context.

3.4. Avoiding "boiling the ocean"

A key lesson we learned was to avoid trying to tackle too many questions at once, a situation often referred to as "boiling the ocean". Instead, we follow the "pay-as-you-go" methodology, starting small and building up gradually [19]. This approach begins with identifying a small set of business questions, and building the ontology and mapping to model those questions. The lesson learned is that this is new technology and stakeholders are eager to see results, thus we must show results quickly (on a weekly basis) and iterate.

These questions must be business-oriented, not technical, and require clear context: Who is asking the question? Why is it being asked? To aid in selecting the right questions, we employ the quadrant framework from [16] which categorizes questions based on their schema and question complexity.

Questions are classified as low or high complexity:

- Low question complexity: Pertains to business reporting use cases, aimed at facilitating daily business operations.
- High question complexity: Arises in the context of Metrics and Key Performance Indicators (KPIs) within an organization. These questions are posed to make informed strategic decisions crucial for organizational success.

Questions also depend on the number of tables required to provide an answer:

- Low schema complexity: Small number of tables (i.e. 0–4), denormalized schema
- High schema complexity: Larger number of tables (5+), normalized schema, many-to many join tables, etc.

Combining these two approaches generates four quadrants to classify questions: (1) Low Question/Low Schema Complexity, (2) High Question/Low Schema Complexity, (3) Low Question/High Schema Complexity, and (4) High Question/High Schema Complexity. This quadrant helps us to choose strategically which questions to address first.

3.5. The importance of test cases

Lastly, we emphasize the importance of creating a comprehensive set of test cases. This involves defining a framework for testing question–answer pairs to validate the system's responses. The pair consists of (1) a question in natural language and (2) corresponding SPARQL/SQL query that provides the accurate answer. During this process, we often encounter ambiguities in questions, resulting in different possible answers.

For example, consider the question *Which agent had the best performance last month?*. How is *best performance* interpreted? Is it the agent that sold the most policies? Or the agent that sold the highest total premium? Furthermore, what is the interpretation of last month? Is it the last calendar month, or is it the last 30 days from today? Identifying and managing these ambiguities is a crucial part of refining the system.

Furthermore, test cases serve as a foundation to make sure that accuracy does not decrease while extending the ontology and mappings. As we modify or add to the ontology, we consistently run these test cases to ensure that the system's functionality remains intact and that changes do not break previous implementations (i.e. regression testing)

4. Industry needs and future research contributions

We highlight several key areas of industry needs where research can contribute:

4.1. Simplifying knowledge engineering

Creating Knowledge Graphs by defining target ontologies and mappings from source relational databases to a target ontology is still a complex social process. Knowledge engineering needs to be simplified which underscores the requirement for more effective tools and methodologies.

From a tooling perspective, we believe there is an opportunity to draw lessons from established practices in areas such as Extract, Transform, Load (ETL) tools, which have effectively navigated similar challenges with tabular data. For example, the open source Data build tool (dbt)[11] enables sql analyst to quickly transform data (i.e. mappings) and applies software engineering practices to the transformation code: git workflow and version control, modularity, testing, and continuous integration/continuous deployment (CI/CD).

From a methodology perspective, we observe that traditional ontology engineering methodologies do not include data mapping tasks [20–22]. Our position is that ontology engineering methods needs to extend their approaches to consider mapping to existing data sources [19,23–25].

Finally, LLMs introduce a promising new mechanism that could streamline knowledge engineering processes through co-pilot style approaches that are approachable for both technical and non-technical users [26–28], which is quickly becoming a target focus area of research.[12] For example, (1) reduce the effort of knowledge acquisition while interviewing domain experts by summarizing a conversation and generating an initial draft of an ontology, (2) reverse engineer SQL or application code that could be implemented as mappings.

We believe that the interplay between ontology engineering, data mapping, and LLMs will be critical in shaping our success.

4.2. User-centric explainability

Explainable AI is an important research topic in academia [29,30]. Knowledge Graphs are being studied as a means to provide explainability [31–34].

[11] https://docs.getdbt.com/

[12] https://kastle-lab.github.io/llms-and-kg-engineering/index

Our position is that we must also focus on: "explainable to whom?" A research opportunity is to better understand the specific needs of users regarding explainability. While there are numerous ways to provide explainability, it is crucial to tailor these methods to the requirements and expectations of different user groups. A user-centric approach to explainability means understanding what level of detail or type of explanation various users need, ranging from business analysts to technical experts. For example, a non technical user asking a question about Loss Ratio may want to see the business glossary to understand how the metric is defined. A technical user may want to see the mapping code.

These explanations can help build trust and make LLM-powered question answering systems more accessible and understandable to a broader audience.

4.3. New approaches to testing non deterministic systems

We are entering an era of working with non-deterministic systems (i.e LLMs). Traditional testing approaches assume systems are deterministic, providing the same output for a given input every time. We need to explore new frameworks and approaches for testing these non-deterministic systems. For example, given a natural language question as input, an LLM-powered system may generate a different query every time it executes. The resulting query may be correct (i.e. returns the expected result), may be incorrect (i.e. the results are wrong) or partially correct (i.e. the result is a subset of the expected result). Thus, these tests should keep track of the variability of the results.

Developing robust testing strategies will be essential to ensure consistency and reliability, and reduce brittleness.

4.4. Small semantics vs. Larger semantics

The concept of semantics is being increasingly referenced in industry, particularly in the context of "semantic layers" within business intelligence and analytics tools. The idea of semantic layers is not new; it dates back to earlier tools such as Business Objects[13] and Looker's LookML,[14] which aimed to provide the definition of what each column is to facilitate analytics.

Currently, many companies and tools employ what can be described as "low-key semantics", primarily focusing on analytics use cases that require modeling metrics and dimensions coming from fact and dimension tables in a data warehouse. Example semantic layers are those from Snowflake,[15] dbt,[16] Cube,[17] AtScale.[18]

In these scenarios, lightweight semantics may be sufficient, and there may not be a need to adopt more expressive ontologies and Knowledge Graph approaches. However, as the complexity of use cases increases, the limitations of these lightweight semantic approaches become apparent. They often lack support for more expressive semantics and advanced knowledge management needs.

We believe the future lies in understanding this gap between the simpler semantics currently used for fact/dimension and the point at which more advanced semantic approaches become necessary.

This requires a careful assessment of when to transition from basic semantic layers to more comprehensive semantic solutions like Knowledge Graphs and ontologies. From a research and tooling perspective, this means investigating the capabilities and limitations of existing "lightweight" semantic tools to better understand when they suffice and when they fall short.

[13] https://patents.google.com/patent/US5555403

[14] https://cloud.google.com/looker/docs/what-is-lookml

[15] https://docs.snowflake.com/en/user-guide/snowflake-cortex/cortex-analyst/semantic-model-spec

[16] https://docs.getdbt.com/docs/use-dbt-semantic-layer/dbt-sl

[17] https://github.com/cube-js/cube

[18] https://github.com/semanticdatalayer/SML

4.5. Multi agent-based question answering systems and problem decomposition

As we build question–answering agent systems, a major consideration is how to break down the problem into smaller, more manageable components. Should an agent deal with the core task of answering questions and managing ambiguity, or should these be split into separate agents.

For example, one agent could handle ambiguity, taking as input questions that may be unclear or open to multiple interpretations. Following our previous example of the question *Which agent had the best performance last month?*, an agent can have a back and forth conversation with the user and determine that a specific version of the question is *Which agent sold the most policies in the last 30 days from today?* Another agent could focus solely on answering well-defined questions that are not expected to have ambiguity.

These agents could work interactively with the user to refine the question into a more specific and manageable form.

The trend indicates a shift toward designing more modular agent systems, where different agents specialize in specific tasks, potentially increasing the overall system's accuracy and user interaction quality. However, this may increase the complexity of the system and the need to have more robust testing. For example, how are test cases defined for an agent that transforms an ambiguous input into a well defined question?

5. Final remarks

Trust in a question answering system comes from a combination of sources. The agent that provides the final answer must be an accountable one; an LLM alone does not qualify. The database or a knowledge base can, as they are curated and accountable resources. When a Knowledge Graph is used as the data source, the ontology can be formally used to verify queries, help vet incorrect queries and identify necessary fixes, leading to increased accuracy, explainability and governance.

When your Knowledge Graph is also a data catalog, the classes and properties in the ontology are related to cataloged data resources. These, in turn, are part of a data governance workflow; the catalog tracks data quality (e.g., whether the dataset was updated on time), provenance (where the information comes from, whether it is the result of a view on another data source), and data stewardship (who vouches for this data). By integrating these aspects — curated data sources, formal validation, standardized documentation, and data governance — Knowledge Graphs provide a robust foundation of trust for LLM-powered enterprise question answering.

CRediT authorship contribution statement

Juan Sequeda: Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Dean Allemang:** Writing – original draft, Methodology, Investigation, Conceptualization. **Bryon Jacob:** Resources, Investigation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Juan Sequeda reports administrative support was provided by data.world. Juan Sequeda reports a relationship with data.world that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] B.F. Green, A.K. Wolf, C. Chomsky, K. Laughery, Baseball: An automatic question-answerer, in: Papers Presented at the May 9–11, 1961, Western Joint IRE-AIEE-ACM Computer Conference, in: IRE-AIEE-ACM '61 (Western), Association for Computing Machinery, New York, NY, USA, 1961, pp. 219–224, http://dx.doi.org/10.1145/1460690.1460714.

[2] C.C. Green, B. Raphael, The use of theorem-proving techniques in question-answering systems, in: Proceedings of the 1968 23rd ACM National Conference, ACM '68, Association for Computing Machinery, New York, NY, USA, 1968, pp. 169–181, http://dx.doi.org/10.1145/800186.810578.

[3] W.A. Woods, Transition network grammars for natural language analysis, Commun. ACM 13 (10) (1970) 591–606, http://dx.doi.org/10.1145/355598.362773.

[4] G.G. Hendrix, E.D. Sacerdoti, D. Sagalowicz, J. Slocum, Developing a natural language interface to complex data, ACM Trans. Database Syst. 3 (2) (1978) 105–147, http://dx.doi.org/10.1145/320251.320253.

[5] D.A. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, E. Shriber, Expanding the scope of the ATIS task: The ATIS-3 corpus, Proc. Work. Hum. Lang. Technol. (1994) 43–48, URL: http://dl.acm.org/citation.cfm?id=1075823.

[6] J.M. Zelle, R.J. Mooney, Learning to parse database queries using inductive logic programming, in: Proceedings of the Thirteenth National Conference on Artificial Intelligence - Vol. 2, 1996, pp. 1050–1055, URL: http://dl.acm.org/citation.cfm?id=1864519.1864543.

[7] L.R. Tang, R.J. Mooney, Automated construction of database interfaces: Intergrating statistical and relational learning for semantic parsing, in: 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 2000, pp. 133–141, URL: http://www.aclweb.org/anthology/W00-1317.

[8] N. Yaghmazadeh, Y. Wang, I. Dillig, T. Dillig, SQLizer: Query synthesis from natural language, in: International Conference on Object-Oriented Programming, Systems, Languages, and Applications, ACM, 2017, pp. 63:1–63:26, http://dx.doi.org/10.1145/3133887.

[9] F. Li, H.V. Jagadish, Constructing an interactive natural language interface for relational databases, Proc. VLDB Endow. 8 (1) (2014) 73–84, http://dx.doi.org/10.14778/2735461.2735468.

[10] S. Iyer, I. Konstas, A. Cheung, J. Krishnamurthy, L. Zettlemoyer, Learning a neural semantic parser from user feedback, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 963–973, URL: http://www.aclweb.org/anthology/P17-1089.

[11] A.-M. Popescu, O. Etzioni, H. Kautz, Towards a theory of natural language interfaces to databases, in: Proceedings of the 8th International Conference on Intelligent User Interfaces, 2003, pp. 149–157, URL: http://doi.acm.org/10.1145/604045.604070.

[12] A. Giordani, A. Moschitti, Automatic generation and reranking of SQL-derived answers to NL questions, in: Proceedings of the Second International Conference on Trustworthy Eternal Systems Via Evolving Software, Data and Knowledge, 2012, pp. 59–76, http://dx.doi.org/10.1007/978-3-642-45260-4_5.

[13] V. Zhong, C. Xiong, R. Socher, Seq2SQL: Generating structured queries from natural language using reinforcement learning, 2017, CoRR abs/1709.00103.

[14] J. Li, B. Hui, G. Qu, J. Yang, B. Li, B. Li, B. Wang, B. Qin, R. Geng, N. Huo, et al., Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls, Adv. Neural Inf. Process. Syst. 36 (2024).

[15] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, IEEE Trans. Knowl. Data Eng. (2024).

[16] J. Sequeda, D. Allemang, B. Jacob, A benchmark to understand the role of knowledge graphs on large language model's accuracy for question answering on enterprise SQL databases, in: O. Hartig, Z. Kaoudi (Eds.), Proceedings of the 7th Joint Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics, NDA, Santiago, Chile, 14 June 2024, ACM, 2024, pp. 5:1–5:12, http://dx.doi.org/10.1145/3661304.3661901, URL: https://arxiv.org/abs/2311.07509.

[17] D. Allemang, J. Sequeda, Increasing the LLM accuracy for question answering: Ontologies to the rescue! 2024, http://dx.doi.org/10.48550/ARXIV.2405.11706, CoRR abs/2405.11706.

[18] D. Allemang, J. Sequeda, Increasing the accuracy of LLM question-answering systems with ontologies, in: G. Demartini, K. Hose, M. Acosta, M. Palmonari, G. Cheng, H. Skaf-Molli, N. Ferranti, D. Hernández, A. Hogan (Eds.), The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11–15, 2024, Proceedings, Part III, in: Lecture Notes in Computer Science, vol. 15233, Springer, 2024, pp. 324–339, http://dx.doi.org/10.1007/978-3-031-77847-6_18.

[19] J.F. Sequeda, W.J. Briggs, D.P. Miranker, W.P. Heideman, A pay-as-you-go methodology to design and build enterprise knowledge graphs from relational databases, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I.F. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II, in: Lecture Notes in Computer Science, vol. 11779, Springer, 2019, pp. 526–545, http://dx.doi.org/10.1007/978-3-030-30796-7_32.

[20] E.F. Kendall, D.L. McGuinness, Ontology engineering, in: Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool Publishers, 2019, http://dx.doi.org/10.2200/S00834ED1V01Y201802WBE018.

[21] C.M. Keet, An Introduction to Ontology Engineering, 2020, URL: https://people.cs.uct.ac.za/%7Emkeet/files/OEbook.pdf.

[22] A. Gómez-Pérez, M. Fernández-López, Ó. Corcho, Ontological engineering: With examples from the areas of knowledge management, e-commerce and the semantic web, Advanced Information and Knowledge Processing, Springer, 2004, http://dx.doi.org/10.1007/B97353.

[23] D. Chaves-Fraga, A. Dimou, Declarative description of knowledge graphs construction automation: Status & challenges, in: D. Chaves-Fraga, A. Dimou, P. Heyvaert, F. Priyatna, J. Sequeda (Eds.), Proceedings of the 3rd International Workshop on Knowledge Graph Construction (KGCW 2022) Co-Located with 19th Extended Semantic Web Conference, (ESWC 2022), Hersonissos, Greek, May 30, 2022, in: CEUR Workshop Proceedings, vol. 3141, CEUR-WS.org, 2022, URL: https://ceur-ws.org/Vol-3141/paper5.pdf.

[24] D. Fraga, Knowledge Graph Construction from Heterogeneous Data Sources exploiting Declarative Mapping Rules (Ph.D. thesis), Technical University of Madrid, Spain, 2021, URL: https://oa.upm.es/67890/.

[25] J. Sequeda, O. Lassila, Designing and building enterprise knowledge graphs, in: Synthesis Lectures on Data, Semantics, and Knowledge, Morgan & Claypool Publishers, 2021, http://dx.doi.org/10.2200/S01105ED1V01Y202105DSK020.

[26] B.P. Allen, L. Stork, P. Groth, Knowledge Engineering Using Large Language Models, Trans. Graph Data Knowl. 1 (1) (2023) 3:1–3:19, http://dx.doi.org/10.4230/TGDK.1.1.3, URL: https://drops.dagstuhl.de/entities/document/10.4230/TGDK.1.1.3.

[27] L. Meyer, C. Stadler, J. Frey, N. Radtke, K. Junghanns, R. Meissner, G. Dziwis, K. Bulert, M. Martin, LLM-assisted knowledge graph engineering: Experiments with ChatGPT, in: C. Zinke-Wehlmann, J. Friedrich (Eds.), First Working Conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow, AI Tomorrow 2023, Leipzig, Germany, 29–20 June, 2023, in: Informatik Aktuell, Springer, 2023, pp. 103–115, http://dx.doi.org/10.1007/978-3-658-43705-3_8.

[28] P. Mateiu, A. Groza, Ontology engineering with large language models, in: 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2023, Nancy, France, September 11–14, 2023, IEEE, 2023, pp. 226–229, http://dx.doi.org/10.1109/SYNASC61333.2023.00038.

[29] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.

[30] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on explainable artificial intelligence (XAI), IEEE Access 6 (2018) 52138–52160.

[31] I. Tiddi, F. Lécué, P. Hitzler, Knowledge Graphs for Explainable Artificial Intelligence: Foundations, Applications and Challenges, IOS Press, 2020.

[32] F. Lecue, On the role of knowledge graphs in explainable AI, Semant. Web 11 (1) (2020) 41–51.

[33] I. Tiddi, S. Schlobach, Knowledge graphs as tools for explainable machine learning: A survey, Artificial Intelligence 302 (2022) 103627.

[34] E. Rajabi, K. Etminani, Knowledge-graph-based explainable AI: A systematic review, J. Inf. Sci. 50 (4) (2024) 1019–1029.