



From questions to insights: how domain-specific semantics boost LLM access to touristic real-time occupancy data

Stefan Neubig^{1,2} · Linus Göhl² · Rahul Radhakrishnan² · Matthias Weirich¹ · Andreas Hein³ · Robert Keller⁴ · Helmut Krcmar¹

Received: 8 June 2025 / Revised: 31 December 2025 / Accepted: 2 February 2026
© The Author(s) 2026

Abstract

Timely access to occupancy data, combining real-time measurements and short-term predictions, is critical for tourism destinations to enhance visitor experiences and manage overcrowding. While this data is increasingly collected, it remains stored in complex databases that are challenging to access flexibly, particularly for large language models (LLMs), which cannot query structured sources directly. LLMs are increasingly valuable in smart tourism, offering conversational access to digital services that support both guests and staff. This paper evaluates the potential of semantic approaches to improve LLM-based access to complex occupancy databases, proposing a zero-shot prompt approach and a semantic agent that iteratively corrects queries until they execute successfully. Both rely on a multi-level tourism occupancy ontology, driving a knowledge graph that captures multiple perspectives on occupancy data, both measured and predicted, and enriched with metadata such as provenance. Evaluated on 19 realistic tourism questions, the agent achieves 42% accuracy and the simpler pipeline 37%, outperforming the evaluated leading text-to-SQL systems while reducing database load and cost. Our findings indicate that semantic guidance improves accuracy and stability in LLM-based access to structured tourism data. Theoretically, this clarifies the role of semantic abstraction, natural-language annotations, input sensitivity, and agentic behavior in query generation. Practically, these insights inform ontology engineering, constrained agent design, and efficient system architectures for LLM-driven tourism applications, such as visitor management and occupancy prediction systems.

Keywords Smart tourism · Visitor management · Occupancy prediction · Ontologies · Knowledge graphs · Large Language Models

1 Introduction

The rapid growth in the global tourism industry has led to concentrated tourist flows in many destinations, with overtourism and overcrowding becoming ubiquitous in sustainability debates (Benner 2020; Butler and Dodds 2022; Capocchi et al. 2019). While quantitative growth historically has driven tourism development, uncontrolled growth is increasingly considered unsustainable in ecological, social, and cultural terms (Benner 2020). As a result, visitor flows' spatial and temporal management (Schmücker et al. 2023) has gained attention in recent years.

Visitor management is increasingly driven by artificial intelligence (AI) leveraging real-time data and advanced forecasting (Bollenbach et al. 2024). Moreover, the rise of large language models (LLMs) and applications like ChatGPT has the potential to fundamentally transform how visitors plan their trips (Mich and Garigliano 2023). Major search engines like Google and Bing increasingly integrate LLMs into their search processes, and travel companies (e.g., Kayak) adopt conversational AI as part of their service offerings (Dwivedi et al. 2024; Kang et al. 2024). Controlled experiments indicate that around three-quarters of travelers prefer ChatGPT-style assistants over traditional search engines for exploratory trip planning, and this preference increases to almost 80% when the assistant is equipped with a domain-specific travel plugin supporting concrete booking decisions (Luo et al. 2025). Given the accelerating adoption of LLMs, AI-driven assistants will likely become the dominant interface for destination discovery and itinerary planning, offering a powerful new lever for visitor management.

However, LLM-based applications must incorporate occupancy data in real-time to effectively address this objective. While such integration seems feasible at first glance, current limitations pose substantial challenges. Occupancy data is typically stored in complex databases with mixed data types, non-standardized units, and ambiguous or highly domain-specific labels, requiring advanced SQL and background knowledge to extract meaningful insights. Although recent advances in text-to-SQL systems (i.e., systems translating natural language into database queries) promise intuitive access, their accuracy on real-world databases remains limited (XLANG Lab 2025). On the other hand, Sequeda et al. (2023) report significant accuracy gains after adding ontologies as semantic mediation layers, highlighting that structured semantic frameworks (e.g., ontologies and knowledge graphs) remain crucial even in the LLM era. Despite the widespread adoption of ontologies and knowledge graphs in tourism, an expressive ontology designed to represent dynamic occupancy data has long been missing (Neubig et al. 2023). Yet, to mitigate overtourism, LLM-based systems must accurately interpret such data in real-time, demanding not just syntactic access but also semantic understanding, thereby enabling automated visitor rerouting and proactive crowd management. In prior work (Neubig et al. 2025), we addressed this gap by introducing a comprehensive ontology for modeling and integrating multiple levels of occupancy data from distributed sources and applied this ontology to a real-world database sample. We demonstrated notable accuracy gains in ontology-guided text-to-SQL, almost tripling the overall performance. This article extends our work by (i) providing a more in-depth discussion of existing text-to-SQL methods, contextualizing our semantic approach within the broader landscape of LLM-database interac-

tion; (ii) embedding the ontology into a full semantic agent; (iii) benchmarking this agent against state-of-the-art text-to-SQL systems on a realistic tourism question set; and (iv) expanding the evaluation to include both accuracy and LLM-related costs. Building on this comparative evaluation, we clarify the theoretical role of semantic knowledge in agent-based architectures and provide practical guidance for ontology engineering, agent configuration, and cost-aware deployment of LLM-based smart tourism systems. Together, these insights advance both our conceptual understanding of semantic mediation in LLM-database interaction and the practical design of conversational AI systems for visitor management.

The remainder of this paper is structured as follows. Section 2 reviews the relevant background, while Sect. 3 presents the methodology behind our semantic approach. Section 4 presents our experimental results, which are discussed in Sect. 5. Section 6 concludes the paper, outlining limitations and directions for future research.

2 Background and related work

In the following subsections, we introduce key concepts such as overtourism and overcrowding, emphasizing the critical role of occupancy data in managing tourism-related challenges. We then explore semantic technologies, particularly knowledge graphs and ontologies, which provide structured representations of domain knowledge and support the integration of heterogeneous data sources. These frameworks are increasingly leveraged to supply LLMs with explicit context information, enhancing their ability to respond to complex user questions. However, while effective in qualitative scenarios, such semantic approaches struggle with large-scale quantitative data, such as real-time occupancy data. Consequently, research has focused on text-to-SQL systems, which translate natural language into executable database queries. This line of work has shown substantial progress in recent years and is central to the following discussion, ultimately motivating the need for domain-specific, semantically grounded alternatives.

2.1 Overtourism, overcrowding, and occupancy prediction

Overtourism is a consequence of years of uncontrolled growth of visitor numbers that historically have driven tourism development (Benner 2020). It poses significant sustainability challenges, leading to numerous adverse effects on destinations, including environmental degradation, socio-cultural disruptions, and economic challenges (Benner 2020; Butler and Dodds 2022; Capocchi et al. 2019). While overtourism is characterized by a general overload of a destination's available facilities and services (R. Butler 2018), overcrowding refers to the temporary accumulation of visitors at a specific time and place (Bollenbach et al. 2024; R. Butler 2018).

We refer to the time- and location-dependent degree of crowdedness as occupancy, which can be measured or predicted using various data sources and methodological approaches. Typical data sources range from primary sources such as tourist arrivals (Abu et al. 2021), parking (Chawathe 2019), and booking data (Phumchusri and Suwatanapongched 2021) to supplementary data like pricing (Tsang and Benoit

2020), weather, or public holidays (Bi et al. 2021; Bollenbach et al. 2024). Occupancy prediction models are often classified into time-series models, econometric models, and machine learning (ML) (Jiao and Chen 2019). In the context of visitor management and overcrowding avoidance, Keller et al. (2024) categorize occupancy prediction strategies besides time series models and ML into rule-based anticipation (i.e., based on expert experiences) and causal models (i.e., models to uncover causal relationships such as weather impact). Although more advanced ML algorithms often yield higher predictive power, the authors conclude that simpler approaches can be sufficient for many practical applications, including visitor management, especially when complemented by human judgment.

These observations underline a central insight: occupancy prediction is highly use-case-specific. While some scenarios require high accuracy and tolerate occasional data gaps, others value continuous and stable data even if the overall precision is lower. Despite this variety of requirements, current approaches rarely provide integrated solutions that flexibly combine different data sources, measurement methods, and prediction techniques. To address this shortcoming, Neubig et al. (2025) introduced the concept of multi-level occupancy, a semantic framework modeled through an ontology for representing multiple occupancy estimates for the same time and location. Each estimate is accompanied by metadata such as provenance and confidence, allowing for context-aware selection and the integration of occupancy data into knowledge graphs.

2.2 Knowledge graphs and ontologies

Knowledge graphs have been defined in various ways, ranging from broad descriptions of graphs containing representations of real-world entities to more formalized, semantically enriched structures grounded in ontologies (Ehrlinger and Wöß 2016; Hogan et al. 2022). Ontologies describe the meaning of data more precisely. They formally define key concepts (e.g., trails, points of interest, events), their relationships, and logical constraints (e.g., valid dates or regional assignment) (Guarino et al. 2009). From a data modeling perspective, they can be viewed as schemas that provide a conceptual foundation for structuring knowledge graphs – for instance, linking a hiking route with nearby accommodation and scheduled events.

Ontologies have also been combined with relational databases, allowing end users to interact with them on a conceptual and intuitive level. As part of the Ontology-Based Data Access (OBDA) paradigm (Xiao et al. 2019), ontologies act as a bridge enabling users to express queries in SPARQL (i.e., a formal query language for retrieving structured, concept-driven data) instead of more technical SQL. Through automatic query rewriting, these SPARQL queries are adapted to the schema of the underlying database, effectively presenting the data as a virtual knowledge graph.

The use of ontologies and knowledge graphs spans many fields and industries. A widely adopted example is Schema.org (Schema.org Community Group 2022), introduced by Google in 2012 and forming the basis of the Google Knowledge Graph. Over time, more specific ontologies and knowledge graphs have been developed for tourism. Examples include the FIWARE Smart Data Models (FIWARE Foundation 2022), defining tourism concepts such as destinations, points of interest, and

weather, and Schema.org extensions (Kärle et al. 2017) refining hotel and attraction descriptions.

Despite numerous initiatives, current tourism knowledge graphs often lack support for dynamic data such as real-time occupancy, either due to missing concepts in existing ontologies or lifecycle models designed for slower, batch-based rather than real-time updates (Neubig et al. 2023). Our previous work (Neubig et al. 2025) proposes an extensible ontology for occupancy-related data and adopts an OBDA approach to enable live data access without manual ingestion.

2.3 Adding context to LLMs

While ontologies and knowledge graphs provide structured, domain-specific knowledge, LLMs are inherently static systems constrained by the data available at training time. Consequently, LLMs are prone to generating outdated or hallucinated responses after deployment (Zhang et al. 2023). A widely adopted solution to this limitation is retrieval-augmented generation (RAG), which enhances an LLM's output by integrating external, up-to-date knowledge. In its most fundamental version (Lewis et al. 2020), a retriever selects relevant context (e.g., documents or websites) using dense vector embeddings (i.e., numerical representations that place semantically similar content nearby) and a generator produces the LLM's output based on this context. More recent variants (Gao et al. 2024) complement the retriever with (user) input refinement, re-ranking, or summarization. In tourism, besides enhancing accuracy, such strategies can even incorporate new perspectives, for instance, a preference for sustainable city trips (Banerjee et al. 2025). Graph-based RAG approaches further improve information retrieval by utilizing knowledge graphs to select relevant content (Edge et al. 2025; Guo et al. 2025; Wang et al. 2024).

However, while these retrieval strategies are effective for extracting qualitative information (e.g., accommodations near hiking routes), they are unsuitable for large volumes of structured, quantitative data such as sensor readings or predictive models. Graph traversal or clustering methods are not designed for querying time series data or analytical operations such as filtering or aggregating values. Hence, more efficient and purpose-built approaches are required, such as text-to-query methods that convert natural language into executable database queries.

2.4 Text-to-SQL

The text-to-query paradigm transforms unstructured natural language into executable database statements. More specifically, text-to-SQL translates inputs into SQL, the dominant query language for relational databases. Traditionally, benchmarks like Spider (Yu et al. 2018) and BIRD (Li et al. 2023) were crafted to make text-to-SQL approaches comparable and maintain leaderboards. While Spider uses curated databases and excludes queries requiring general knowledge, BIRD adds real-world noise and a broader range of SQL constructs. However, both are insufficient to challenge modern LLMs. Consequently, Lei et al. (2025) introduced Spider 2.0, a more demanding benchmark reflecting the capabilities of current LLMs and causing models that performed well on Spider to drop sharply in performance. For instance,

DIN-SQL (Pourreza and Rafiei 2023) declined from 85.3/100 (Yale-LILY 2025) to 1.5/100 in Spider 2.0-lite (XLANG Lab 2025). More recent approaches (Deng et al. 2025) have since improved results to 37.11/100 (as of 28 May 2025), though performance remains well below earlier, simpler benchmarks.

Modern text-to-SQL methods predominantly leverage LLMs and typically follow the structure shown in Fig. 1. They comprise several steps that address specific (sub-) tasks, often interacting with one or more LLMs via crafted prompts. These tasks can broadly be categorized into schema linking (i.e., making the database schema available to the LLM), query generation (i.e., generating one or more SQL query candidates), and error correction (i.e., correcting potential errors and refining the query). A higher-level control flow orchestrates these tasks and specifies how they interact, ranging from purely sequential execution to complex iterative refinements. Importantly, not all the mentioned tasks are mandatory and may vary with use case or database complexity. The individual components of text-to-SQL methods are discussed below.

2.4.1 Models and prompting

Models can be broadly categorized as general-purpose and finetuned models. General-purpose models (e.g., OpenAI GPT, Anthropic Claude, Google Gemini) have been trained on a broad range of data with no specific task. They generalize well to new, unseen tasks by providing them with appropriate context (as part of the prompt). Conversely, finetuning continues the training process of general-purpose models to extend them with domain-specific knowledge and skills, such as the alignment between user questions and database schemas (Deng et al. 2021). Advanced finetun-

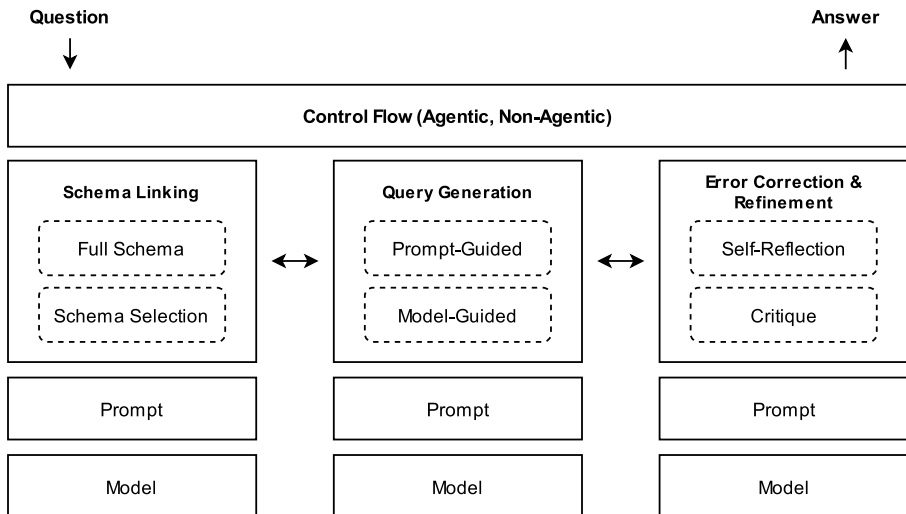


Fig. 1 Synthesized text-to-SQL framework. Three main tasks (i.e., schema linking, query generation, error correction & refinement) performing calls to LLM models via tailored prompts are orchestrated by an overall control flow to answer (natural language) user questions based on a structured database (source: authors' own)

ing strategies use data augmentation or architectural changes to further increase a model's capabilities, including cross-dialect portability (Pourreza et al. 2024) and template-based query generation (Fu et al. 2023). In many cases, finetuned models enable the use of smaller and more cost-efficient models while maintaining performance (Li et al. 2024).

Finetuned and general-purpose models rely on prompting techniques to generate SQL queries, usually combining the user question with supplementary instructions. Most text-to-SQL approaches apply few-shot prompting by providing examples of expected outputs (Gao et al. 2023; Mai et al. 2024), while some use zero-shot prompting and omit such examples (Dong et al. 2023). More advanced prompting approaches, such as chain-of-thought reasoning, encourage multi-step problem solving (Zhang et al. 2023).

2.4.2 Tasks

Text-to-SQL typically involves three core tasks: schema linking, query generation, and error correction. Schema linking identifies relevant parts of a database schema for a given question. While several approaches (Gao et al. 2023; Maamari et al. 2024) simply incorporate the full schema, selective schema extraction remains essential for large databases or limited context windows (i.e., the text a model can process at once). Existing selection approaches vary and include prompt-based solutions (Lee et al. 2024), finetuned expert models (Pourreza and Rafiei 2024), specialized agents (Talaie et al. 2024), and the compression of an extensive schema into a question-focused context (Deng et al. 2025). Besides, prior research discussed different schema representation methods, ranging from using the original SQL queries (Gao et al. 2023) to alternative representations, possibly enriched with examples (Gao et al. 2025).

Building upon schema linking, the query generation task transforms user questions into executable SQL queries. Most approaches are primarily guided by prompting (few-/zero-shot). While longer prompts lead to a higher token consumption, supplying examples often improves a method's accuracy (Dong et al. 2023), particularly when examples are selected based on the user question (Mai et al. 2024). In addition, finetuning further strengthens a model's capability for SQL-related tasks such as classifying user questions or generating SQL (Qin et al. 2025) or helps ensure syntactic correctness through template-based generation (Fu et al. 2023).

After query generation, text-to-SQL methods may incorporate an error correction and refinement task, based on self-reflection or critique. Self-reflection approaches inspect and revise their own outputs, for instance, by iteratively re-executing and refining a query draft (Xia et al. 2024) or generating explanations in natural language to identify and fix mistakes (Chen et al. 2023). Conversely, critique approaches employ external evaluator models to judge query candidates rather than leaving the judgment to the query generation model itself (Shen and Kejriwal 2024).

2.4.3 Control flow

Text-to-SQL systems rely on a control flow that defines how individual tasks are orchestrated. Approaches range from deterministic sequences to dynamic, agentic workflows. Non-agentic systems follow predefined steps where the overall execution logic remains static, for instance, with fixed loop counts guiding query generation (Dong et al. 2023) or LLM involvement limited to final candidate selection without affecting the overall control flow (Lee et al. 2024). In contrast, agentic approaches allow models to make orchestration decisions based on reasoning outcomes. Their autonomy varies, from limited guidance within iterative correction loops (Xia et al. 2024) or user clarification requests (Guo et al. 2025) to more flexible systems that operate autonomously across multiple tasks, coordinating feedback and correction through a central agent (Askari et al. 2024) or employing multiple agents to detect and fix SQL errors (Cen et al. 2025). However, too much autonomy can lead to a potentially unbounded number of interactions and unpredictable token costs.

2.5 Need for a domain-specific semantic approach

Despite extensive research and advancements in text-to-SQL, current approaches remain far from practical application. Their accuracy on real-world databases continues to fall short, reaching no more than 37.11/100 on the simplest Spider 2.0 benchmark as of 28 May 2025 (XLANG Lab 2025). While knowledge graphs have gained significant popularity in other LLM domains and support embedding primarily qualitative information into applications like chatbots (e.g., points of interest, route descriptions), studies integrating domain-specific semantics into the text-to-SQL context remain scarce – despite their potential for representing quantitative data such as occupancy. Although such approaches may reduce generalizability, recent studies (Sequeda et al. 2023) show that incorporating domain knowledge via ontologies can significantly improve query accuracy, potentially bridging the gap between academic benchmarks and real-world demands. This is particularly important in tourism, where users often pose vague or highly contextual questions (Cossatin et al. 2025) involving current weather, proximity, or personal preferences – such as looking for a “good place nearby” or “something interesting” with scenic views. Moreover, ontologies and knowledge graphs are well established, providing a solid foundation for domain-specific semantic integration.

In the following, we extend the occupancy framework mentioned earlier with a semantic agent, combining widely adopted ontologies with the multi-level occupancy ontology to improve the querying accuracy of geospatial occupancy data from a real-world database of Outdooractive, one of Europe’s largest outdoor platforms. Extending (Neubig et al. 2025), we further enhance our fundamental approach by utilizing a semantic agent with error correction and compare it against state-of-the-art text-to-SQL methods that have performed strongly on recent benchmarks.

3 Method

While qualitative data, such as points of interest and route descriptions, are available to LLMs using existing RAG methods, embedding large volumes of quantitative data, such as occupancy data, remains challenging – particularly if this data must be pre-processed at query time, for example, when requiring sophisticated aggregation or grouping. To address this, we adopt a text-to-SPARQL approach instead of a conventional text-to-SQL pipeline.

The conceptual difference between these text-to-query approaches is illustrated in Fig. 2. While text-to-SQL directly translates user questions into executable SQL queries over a technical schema, text-to-SPARQL operates on a domain-specific virtual knowledge graph, following the OBDA paradigm introduced earlier, making it particularly well-suited for capturing the intent behind natural language questions. OBDA is usually described as a triple $\langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$, where a database schema $\mathcal{S}$ is connected with an ontology $\mathcal{O}$ using a set of conceptual mappings $\mathcal{M}$ (Xiao et al. 2019). These mappings specify how elements from the database correspond to concepts and relationships in the ontology. Therefore, instead of requiring detailed knowledge about the database's relational table structures, SPARQL queries can be formulated using high-level vocabularies that describe meaningful domain concepts (e.g., routes, points of interest, or their relationships), explicitly specified in an ontology. Moreover, with the query formulation decoupled from the database structure, we can flexibly integrate evolving domain knowledge without affecting the underlying data.

To investigate to what extent LLMs benefit from such a domain-specific abstraction in the context of dynamic occupancy data retrieval, we combine a touristic schema.org ontology extension (i.e., ODTA) with an ontology specifically designed for occupancy data. This combined ontology mediates queries between end users and a relational database, using LLMs to first translate natural language inputs into formal SPARQL queries. Second, we leverage the Ontop framework (Calvanese et al. 2015) to establish a conceptual mapping between the combined ontology and the target database. Two semantic text-to-SPARQL approaches are compared to traditional text-to-SQL methods concerning correctness, the number of necessary native database queries, LLM token counts, and associated costs.

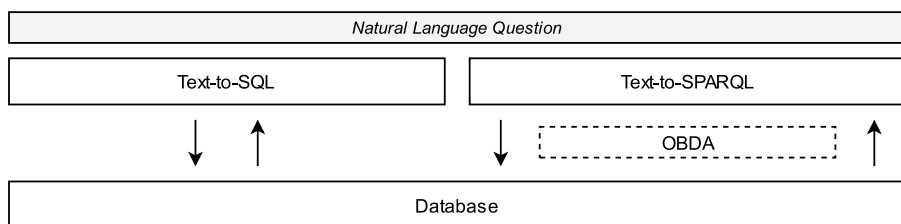


Fig. 2 Text-to-SQL vs. Text-to-SPARQL (source: authors' own)

3.1 Tourism occupancy ontology

The tourism occupancy ontology (TOO) implements the concept of multi-level occupancy. It incorporates the theoretical framework of Keller et al. (2024), representing multiple occupancy measures and estimates. It offers metadata such as provenance and confidence to allow a context-aware selection of the most suitable occupancy values. The latest TOO version is first described in (Neubig et al. 2025). It was built following the NeOn ontology engineering framework (Suárez de Figueroa 2010) and allows the continuous evolution of live and reusable ontologies, one of the critical requirements for an evolving field like tourism. To allow LLMs to leverage the TOO more effectively, concepts and properties are not only formally specified but have been enriched with rich natural-language descriptions. This additional semantic context facilitates alignment between the ontology's symbolic structure and the way LLMs process language, improving their ability to interpret and reason over domain knowledge. The TOO is illustrated in Fig. 3 and categorizes possible prediction approaches and data sources alongside dimensions for detailed measurement and determination details. Its modules are briefly described below.

- The *Geography* module defines abstract representations of fundamental spatial concepts, including points, line strings, and polygons. While it could be used as a standalone geographic vocabulary, its primary purpose lies in serving as a structural adapter for connecting to more specific third-party ontologies. This enables seamless alignment with existing domain vocabularies, such as the ODTA ontology (Kärle et al. 2017).
- The *Occupancy Core* module establishes a connection between occupancy information and spatial entities. The occupancy concept is central to this module and combines concrete geo-objects, defined in the Geography module, with basic occupancy information. Using its more specific definitions, such an occupancy can refer to areas (e.g., parking places) or passages, capturing directional flow (e.g., passing through an entrance). Relationships between different occupancy

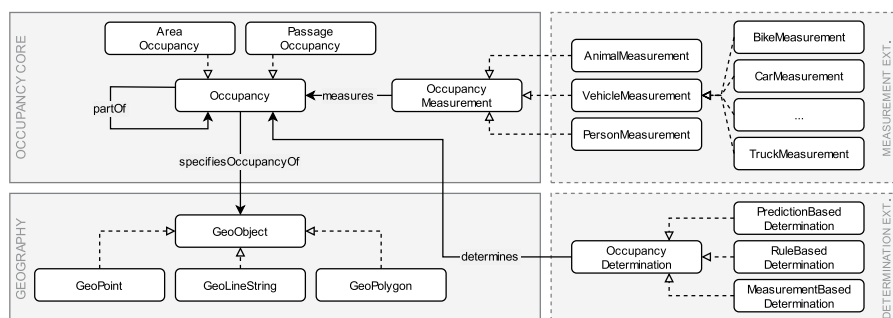


Fig. 3 Multi-level occupancy ontology comprising four modules: geography (i.e., adapter to 3rd-party ontologies), occupancy core (i.e., its main concepts), measurement extension (i.e., specification of occupying objects), and determination extension (i.e., specification of the occupancy determination method). Source: Neubig et al. (2025), reproduced with permission from SNCSC; not part of this article's Open Access licence

instances can also be modeled hierarchically – for instance, multiple entrances (represented as passage occupancies) may form parts of a larger area occupancy (representing the complete parking place). The actual values are captured through occupancy measurements, including capacity information and recorded or estimated occupancy levels (either absolute or relative) over a given period.

- The *Measurement Extension* module adds granularity to occupancy measurements by allowing distinctions between different object categories. This is particularly relevant in tourism-related settings, where infrastructure may serve a variety of user groups and transportation modes. For instance, a parking facility in a hiking area might accommodate both private cars and tour buses, each with different spatial requirements. The module supports defining object-specific capacities and enables different types of space utilization.
- The *Occupancy Determination Extension* captures the provenance of occupancy data, specifying how individual measurements were obtained. It distinguishes between sources such as sensors, predictions, or manual counts. Additionally, it allows for the inclusion of confidence values, expressing the reliability or uncertainty associated with a given measurement or prediction, and is helpful for a (semi-)automatic selection of use-case-specific values.

To evaluate text-to-query methods in a more realistic setting, we integrate the TOO with selected elements of the ODTA domain specifications (Kärle et al. 2017), incorporating basic concepts such as points of interest or routes. This integration builds on the geography module’s concept abstractions, which are aligned with their ODTA counterparts through semantic mappings (i.e., explicit correspondences between ontology classes and properties). The resulting ontology ($\mathcal{O}$) is then mapped to the evaluation’s data described below.

3.2 Data overview

To evaluate semantic approaches against text-to-SQL methods, we assembled a benchmark dataset based on real-world data from the Outdooractive platform. The dataset captures a representative subset of tourism infrastructure, including hiking tours, points of interest, and regional entities. These elements are interconnected through spatial and semantic relations, enabling complex query scenarios. To simulate realistic occupancy conditions across different contexts, we enriched the dataset with synthetically generated values reflecting diverse temporal and spatial usage patterns.

Figure 4 illustrates the relevant parts of the underlying database schema ($\mathcal{S}$). The **_entities* tables persist geographic objects and their associated metadata, such as geometry, difficulty level, duration, and additional attributes specific to each object type. Descriptive and multilingual content is handled separately in associated text tables (**_texts*). Object relationships, such as containment or spatial hierarchy, are stored as subject-predicate-object triples (e.g., a point of interest intersects with a hiking tour) in the central *relations* table. Occupancy information is modeled in two layers. The *occupancy_object* table connects occupying agents (e.g., people, vehicles) to relevant spatial objects, while *occupancy* captures time-resolved measure-

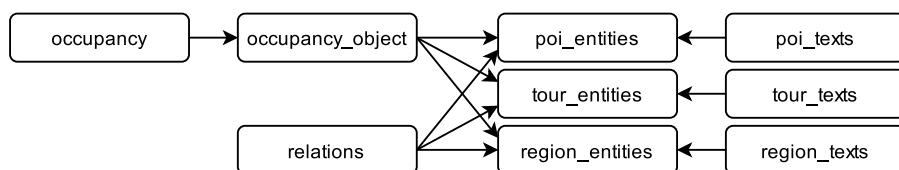


Fig. 4 Simplified database schema of the Outdooractive excerpt. Solid arrows indicate foreign key relations. Source: Neubig et al. (2025), reproduced with permission from SNCSC; not part of this article's Open Access licence

Table 1 Database tables of the Outdooractive excerpt, including row and column counts (source: Author's own)

Table	# Columns	# Rows
poi_entities	46	100,000
poi_texts	5	160,270
tour_entities	59	100,000
tour_texts	5	223,884
region_entities	31	3,500
region_texts	5	4,542
relations	5	594,249
occupancy_object	5	2,902
occupancy	8	7,187
Sum	169	1,196,534

ments alongside confidence indicators estimating data reliability. All original data was sampled from a productive system environment but narrowed to selected tourism regions to maintain experimental tractability. Table 1 provides a breakdown of included entity types and their numbers.

To expose the data as a virtual knowledge graph, we defined a formal mapping ($\mathcal{M}$) that connects each database table to one or more ontological concepts. For instance, specific segments of the *occupancy* table are mapped to *OccupancyMeasurement*, while others correspond to *OccupancyDetermination*.

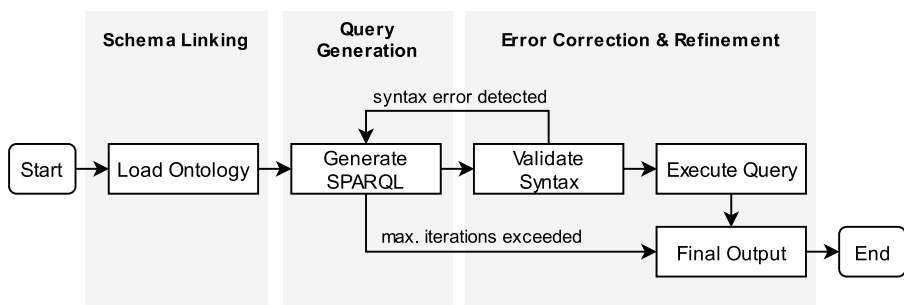
3.3 Text-to-SPARQL

We follow two distinct approaches to investigate the potential of text-to-SPARQL methods for occupancy data retrieval. As in Neubig et al. (2025), we employ a zero-shot LLM call to a general-purpose model, containing only basic instructions for query generation, the user question and the schema in a structured prompt (see Table 2).

Additionally, we introduce a semantic agent following our synthesized framework for text-to-SQL approaches (see Fig. 5). Using an up-to-date model, we provide the agent with the complete ontological schema following the observations in Maamari et al. (2024) that more recent LLMs tend to be capable of processing full schemas effectively. For query generation, we recycle the prompt specified in Table 2. However, after generating a query candidate, we perform an iterative syntax error detection to address our previous finding (Neubig et al. 2025) that SPARQL query generations often suffer from (relatively trivial) syntax errors (e.g., missing prefix definitions or

Table 2 LLM configuration including prompts and variables. Upper-case input is rendered in *italic* to improve readability (source: Author's own)

Zero-Shot Text-to-SPARQL	
Prompt	Consider the OWL 2 ontology below (TTL format): {schema} You are a helpful assistant. You specialize in creating SPARQL queries that answer a given question. Only return the query, assuming any literal as tagged string (@en). Strictly adhere to the domain of properties and follow all paths to the deepest level necessary. Always fetch data properties using optional. Today is {date} . Here's the question: {question}
Schema	Ontology in Turtle (TTL) format
Model	GPT-4o (General-purpose LLM)

**Fig. 5** Semantic agent architecture following the same synthesized framework as text-to-SQL approaches: load ontology corresponds to the schema linking task, generate SPARQL implements query generation, and Validate syntax and execute query correspond to error correction and refinement. The overall control flow is agentic with a low degree of freedom (source: Author's own)

unsupported function calls). To avoid unnecessary SPARQL executions, syntax validation is performed locally to check for the most common errors. If syntax errors are identified, these are appended to the generator prompt, providing additional context for the following query generation iteration. To prevent the agent from going into endless loops when syntax errors remain repeatedly unfixed, we introduced a maximum of 10 query generation attempts.

After an executable query candidate (i.e., a SPARQL query with no syntax errors) has been found, the semantic agent executes this query on a running Ontop instance (Calvanese et al. 2015), a software implementing the OBDA paradigm. Ontop, initialized with the ontology $\mathcal{O}$, the target database schema $\mathcal{S}$, and the specified mapping $\mathcal{M}$, then translates the SPARQL query into SQL and hands it over to the database. The results of this query are then composed and shown to the user as the final output.

3.4 Model comparison

To evaluate our text-to-SPARQL approaches in the context of the tourism occupancy ontology, we compare their performance to state-of-the-art text-to-SQL approaches. As an illustrative example, we crafted a set of 19 tourism-specific competency questions (CQs) representing common questions that could help make informed decisions

and mitigate overcrowding in tourism destinations (Neubig et al. 2025). Moreover, we designed the CQs to cover a broad range of the ontology's capabilities: concepts and relationships modeled to represent occupancy. Our CQs are a result of an expert workshop. They are summarized in Table 3 and may be categorized as follows:

Table 3 Curated list of competency questions. Quotes are not part of the question but have been added to improve readability. Source: Neubig et al. (2025), reproduced with permission from SNCSC; not part of this article's Open Access licence

Competency Question	
Group 1: Occupancy	
1	How full is the parking place "Ried Allgäuer Straße Birkenweg" around 6 pm today?
2	Which vehicles are occupying the parking space "Heitlern School Center P1" this morning around 10 AM?
3	How crowded will be the hiking tour "Grüntalspitze Crossing" today from 8 AM to 10 AM?
4	What percentage of the total capacity of tour "Wannenjoch – Iseler" is occupied at 1 PM today?
5	How confident are you that I will find a parking space on 24th August in "Ried House of the Guest" around 12 PM?
6	What is the total capacity of the parking place "Mountain at Hörnle"?
7	How many trucks are expected to occupy the parking space "Heitlern School Center P1" around 6 PM this evening?
8	How crowded will be the tour "The Moor and Hay Meadow Path" on 24th August, 2024 during the day?
9	How full is the parking place "Mountain at Hörnle" going to be on 24th August, 2024 around 2 PM?
10	How full is "Ried Allgäuer Straße Birkenweg" expected to be on 24th August compared to today around 5 PM?
11	How many free parking spaces will be there at "Mountain at Hörnle" on 24th August 10 AM?
Group 2: Peak Time Analysis	
12	How is the average occupancy of "Heitlern School Center P1" on 24th August compared to today?
13	When is the "Heitlern School Center P1" expected to be >95% full?
14	Which of the days among the 20th and 24th August 2024, are the tours in "Starnberg" least occupied?
15	What were the minimum and maximum occupancies at "Ried Allgäuer Straße Birkenweg" on the 20th and the 24th August 2024 around 6 PM?
Group 3: Proximity	
16	Give me 5 tours that start around "Schliersee".
17	How many parking spaces are within 10 km radius around the start point of tour "Grüntalspitze Crossing"?
Group 4: Combined Questions	
18	How many free parking spaces are there 10 km around the start location of tour "Wannenjoch – Iseler" around 10 AM?
19	Find 5 tours whose start points are within 100 km from "Tegernsee" and are not crowded.

- *Occupancy*: Covers questions that help to get an overview of the current and future occupancy of a point of interest, a tour, or parking spaces
- *Peak Time Analysis*: Questions to get insights on peak occupancy within a given time frame
- *Proximity*: Questions that help to understand spatial relationships between a point of interest, a tour, or parking spaces, helping understand how many options are accessible from a specified location
- *Combined Questions*: Questions that are combinations of the above to include even more complex queries

Based on the described data sample, we compare our semantic approach to a zero-shot text-to-SQL baseline (similar to our zero-shot text-to-SPARQL prompt) and to two state-of-the-art text-to-SQL approaches: C3-SQL (Dong et al. 2023), one of the leading approaches in the Spider 1 benchmark with fully available source code, and ReFoRCE (Deng et al. 2025), the leading approach on the much more complex Spider 2.0 benchmarking leaderboard at the time of evaluation, achieving an overall performance of around 37/100. We used the same model (OpenAI GPT-4o) for all evaluations to keep the comparison fair.

In contrast to larger benchmarks such as Spider and BIRD, where automatic evaluation is performed following different evaluation, we follow a more elaborate but more precise approach, evaluating each method by human experts. Since the semantic equivalence of SQL queries is generally undecidable (Imielinski and Lipski 1984), automatic evaluation metrics have known limitations. The execution accuracy metric (i.e., comparing the query results) often yields false positives when different queries return the same result. In contrast, the exact match metric (i.e., only comparing the query with the gold standard) suffers from false negatives due to possible syntactic variation. Therefore, we opted to perform a manual evaluation. Two human experts assess all queries and outcomes that result from applying the text-to-query methods to the CQs specified above. They assign a score $C \in \{0, 1\}$ to each generated query and its corresponding result based on correctness. In cases of disagreement, a short discussion is held to reach a consensus, ensuring a fair and consistent evaluation.

Additionally, we collect the input and output token usage information for each LLM call. This data compares query generation costs for each question across different methods. LLMs are non-deterministic, even if their temperature (i.e., a parameter that controls the randomness in token selection) is set to zero. For this reason, we ran the evaluation multiple times and reported the score that occurred in more than 50% of the runs. Furthermore, we report the number of query executions performed by the method to get an overview of the database's utilization.

4 Results

The overall results of our comparison are shown in Table 4. In both cases (text-to-SQL and text-to-SPARQL), the non-agentic, zero-shot methods utilize the fewest resources in terms of LLM tokens, leading to the lowest cost. The significant difference in the high number of input tokens in both zero-shot approaches is due to the

Table 4 Average results from comparing different text-to-query approaches based on 19 evaluated questions (CQs). Metrics include the proportion of correctly answered questions ($C_\emptyset$), the average number of executed database queries per question ($Q_\emptyset$), the average input/output tokens ($T_{in,\emptyset}$, $T_{out,\emptyset}$), and associated costs ($\$_\emptyset$) (source: authors' own)

	Approach	$C_\emptyset$	$Q_\emptyset$	$T_{in,\emptyset}$	$T_{out,\emptyset}$	$\$_\emptyset$
Text-to-SQL	Zero-Shot SQL	0.1053	1	2,738	136	0.0082
	C3-SQL	0.0000	3.2	2,413	17,200	0.1780
	ReFoRCE	0.3158	84.8	952,495	41,546	2.7967
Text-to-SPARQL	Zero-Shot SPARQL	0.3684	1	6,312	251	0.0183
	Semantic Agent	0.4211	0.8	23,989	1,425	0.0742

increased size of the ontology compared to the relatively small excerpt of the database schema. However, the database schema's size increases when more complex databases are provided, while the ontology size is more use-case-specific and remains relatively constant. Regarding the number of executed database queries, the semantic agent skips query generation in cases where syntactic correctness cannot be ensured, resulting in the lowest average of executed database queries. Since all approaches were executed using the same base model, the method's costs depend on the number of input and output tokens. Regarding accuracy (i.e., the proportion of correctly answered questions), semantic text-to-SPARQL approaches constantly outperformed text-to-SQL approaches in our evaluation. A more detailed analysis of each method's performance can be found below.

4.1 Model-wise evaluation

Table 5 details the per-question results of the analyzed text-to-SQL approaches. $C \in \{0, 1\}$ indicates whether a query's result is correct, $Q \in \mathbb{N}_0$ refers to the number of queries executed on the database, thus, the load that is put on the database as a result of a single text-to-query execution. $T_{in}, T_{out} \in \mathbb{N}_0$ refer to the number of input and output tokens, respectively, and $\$ \in \mathbb{R}_{\geq 0}$ indicates the costs associated with these tokens (for OpenAI GPT-4o as of 28 May 2025). The tokens and the number of executed database queries for each approach reflect their internal principles. In the zero-shot text-to-SQL approach, a specialized prompt is sent to the LLM once, resulting in a single SQL query as a response, which may be correct or not. Since there are no retries, the number of queries remains constant at 1. Conversely, C3-SQL generates up to five SQL query candidates per question, leading to a higher number of executed queries (3.2 on average) and a significantly higher number of output tokens. While the accuracy of the zero-shot text-to-SQL approach reflects the findings in Neubig et al. (2025), C3-SQL could not answer a single question correctly. While this seems unexpected, it confirms the results of other methods that reached high scores in the Spider 1 benchmark and dropped dramatically in performance when applied to real-world databases (e.g., DIN-SQL scored 85.3/100 in Spider 1 and dropped to 1.46/100 in Spider 2).

ReFoRCE takes a fundamentally different approach. It intensively interacts with the database through repeated exploratory queries, iterative refinement, and parallel candidate evaluation. Consequently, it executes an average of 84.8 SQL statements per question, over 25 times more than C3-SQL, putting a significant load on the data-

Table 5 Results of text-to-SQL approaches, comprising an indicator of whether the CQ could be answered (C), the number of executed database queries (Q), the LLM's input and output tokens (T_{in} , T_{out}), and the associated costs (source: authors' own)

CQ	C	Q	T_{in}	T_{out}	\$	CQ	C	Q	T_{in}	T_{out}	\$
1	0	1	2,735	1,314	0.0081	11	0	1	2,737	156	0.0084
2	0	1	2,735	189	0.0087	12	0	1	2,736	148	0.0083
3	0	1	2,737	131	0.0081	13	0	1	2,733	100	0.0078
4	0	1	2,736	135	0.0082	14	0	1	2,742	138	0.0082
5	0	1	2,742	147	0.0083	15	0	1	2,751	135	0.0082
6	1	1	2,729	56	0.0073	16	0	1	2,726	62	0.0074
7	0	1	2,738	120	0.0080	17	0	1	2,737	112	0.0080
8	0	1	2,739	144	0.0083	18	0	1	2,741	175	0.0086
9	1	1	2,742	158	0.0084	19	0	1	2,735	142	0.0083
10	0	1	2,742	214	0.0090						
$\emptyset$ $C_\phi = 0.1053$, $Q_\phi = 1$, $T_{in,\phi} = 2738$, $T_{out,\phi} = 136$, $\$_\phi = 0.0082$											

CQ	C	Q	T_{in}	T_{out}	\$	CQ	C	Q	T_{in}	T_{out}	\$
1	0	5	3,425	19,998	0.1979	11	0	5	3,469	25,785	0.2281
2	0	5	3,337	19,612	0.2124	12	0	1	1,158	12,904	0.2227
3	0	2	1,756	15,930	0.2162	13	0	1	1,147	11,570	0.1800
4	0	1	1,192	14,047	0.1997	14	0	5	3,484	23,194	0.2269
5	0	3	2,286	14,916	0.2011	15	0	1	1,203	12,593	0.2366
6	0	5	3,363	16,345	0.1567	16	0	1	1,176	11,829	0.1544
7	0	5	3,368	18,624	0.2107	17	0	5	3,467	21,020	0.1942
8	0	5	3,463	20,089	0.2024	18	0	5	3,392	25,023	0.2507
9	0	3	2,286	14,923	0.2177	19	0	2	1,702	15,045	0.2161
10	0	1	1,176	13,355	0.3376						
$\emptyset$ $C_\phi = 0.0000$, $Q_\phi = 3.2$, $T_{in,\phi} = 2,413$, $T_{out,\phi} = 17,200$, $\$_\phi = 0.1780$											

CQ	C	Q	T_{in}	T_{out}	\$	CQ	C	Q	T_{in}	T_{out}	\$
1	1	94	620,669	32,325	0.5733	11	1	80	303,772	19,371	0.7726
2	0	81	1,701,418	66,920	0.2806	12	0	81	1,226,568	51,316	0.5350
3	1	70	408,324	25,229	0.0266	13	0	95	1,815,418	63,043	0.4611
4	1	92	692,994	36,665	0.7652	14	0	109	1,076,267	47,212	0.7566
5	0	86	922,929	39,676	0.3238	15	0	67	1,236,580	51,742	1.7111
6	1	72	151,173	9,711	1.9698	16	0	80	301,798	14,092	0.3163
7	0	101	1,388,000	59,474	1.5660	17	0	71	469,340	22,938	1.0723
8	0	93	1,040,480	43,394	1.5579	18	0	79	1,005,769	51,549	1.1270
9	1	108	141,6042	59,387	0.8147	19	0	86	760,953	33,237	0.2436
10	0	66	1,558,912	62,091	0.0157						
$\emptyset$ $C_\phi = 0.3158$, $Q_\phi = 84.8$, $T_{in,\phi} = 952,495$, $T_{out,\phi} = 41,546$, $\$_\phi = 2.7967$											

base. Moreover, it consumes significantly higher model-specific resources (an average of 952,495 input and 41,546 output tokens per question). While this extensive approach yields substantially improved accuracy (31.58% correct answers), it still leaves many questions unresolved.

Table 6 shows the results of the evaluated text-to-SPARQL approaches. The zero-shot text-to-SPARQL approach could answer seven questions correctly. In contrast

Table 6 Results of text-to-SPARQL approaches, comprising an indicator of whether the CQ could be answered (C), the number of executed database queries (Q), the LLM's input and output tokens (T_{in} , T_{out}), and the associated costs (source: authors' own)

CQ	C	Q	T_{in}	T_{out}	\$	CQ	C	Q	T_{in}	T_{out}	\$
1	1	1	6,310	245	0.0182	11	0	1	6,312	190	0.0177
2	0	1	6,310	227	0.0180	12	0	1	6,311	370	0.0195
3	1	1	6,312	273	0.0185	13	1	1	6,308	130	0.0171
4	0	1	6,311	186	0.0176	14	0	1	6,317	213	0.0179
5	0	1	6,317	203	0.0178	15	0	1	6,326	370	0.0195
6	1	1	6,304	106	0.0168	16	0	1	6,301	179	0.0175
7	1	1	6,313	194	0.0177	17	0	1	6,312	320	0.0190
8	1	1	6,314	308	0.0189	18	0	1	6,316	364	0.0194
9	1	1	6,317	232	0.0181	19	0	1	6,310	340	0.0192
10	0	1	6,317	323	0.0190						
$\emptyset$ $C_\phi = 0.3684$, $Q_\phi = 1$, $T_{in,\phi} = 6,212$, $T_{out,\phi} = 251$, $\$_\phi = 0.0183$											
CQ	C	Q	T_{in}	T_{out}	\$	CQ	C	Q	T_{in}	T_{out}	\$
1	1	1	12,623	560	0.0372	11	0	1	2,304	284	0.0086
2	1	1	12,623	568	0.0372	12	0	1	25,324	1,586	0.0792
3	1	1	6,304	368	0.0194	13	1	1	6,300	165	0.0174
4	0	1	6,303	226	0.0180	14	0	1	57,298	3,275	0.1760
5	0	1	6,309	232	0.0181	15	0	1	77,125	3,962	0.2324
6	1	1	6,296	125	0.0170	16	0	1	12,602	450	0.0360
7	1	1	12,645	441	0.0360	17	0	0	77,509	5,606	0.2498
8	1	1	6,314	316	0.0189	18	0	0	25,408	2,007	0.0836
9	1	1	12,637	567	0.0373	19	0	0	70,868	5,039	0.2276
10	0	1	19,008	1303	0.0606						
$\emptyset$ $C_\phi = 0.4211$, $Q_\phi = 0.8$, $T_{in,\phi} = 23,989$, $T_{out,\phi} = 1,425$, $\$_\phi = 0.0742$											

to its lower token-consuming text-to-SQL twin, the generated SPARQL queries are slightly wordier, which explains the higher number of output tokens. The gap in input tokens is explained by the different sizes of the ontology and the (reduced) database schema. Our semantic agent leverages iterative syntax validation before query execution, ensuring that at most one valid SPARQL query per question is submitted to the endpoint. If no syntactically valid query can be constructed within a defined number of attempts, no query is executed at all, leading to an average of 0.8 query executions. Since neither semantic reflection on query results nor multiple candidate generation occurs, the number of executed queries remains constant. With an average of 23,989 input and 1425 output tokens per question, resource usage is significantly lower than for C3-SQL and ReFoRCE. Despite this modest resource footprint, the semantic agent achieves the highest accuracy (42.11%) among all evaluated approaches.

4.2 Syntax and semantics: error distribution

In the following, we focus on those queries that did not lead to the correct result and analyze their different syntactic and semantic errors. An overview of the error distribution in text-to-SQL approaches is illustrated in Table 7.

Table 7 Error distribution of text-to-SQL approaches (source: authors' own)

Query Generation		Zero-Shot	C3-SQL	ReFoRCE
Skipped		0/19	0/19	1/19
Executable	Correct Result	2/19	0/19	6/19
	Empty Result	9/19	9/19	0/19
	Incorrect Result	7/19	2/19	12/19
Non-Executable	General Syntax Error	1/19	1/19	0/19
	Input Type Error	0/19	7/19	0/19
	Unknown Function	0/19	0/19	0/19

Table 8 Error distribution of text-to-SPARQL approaches (source: authors' own)

Query Generation		Zero-Shot	Semantic Agent
Skipped		0/19	3/19
Executable	Correct Result	7/19	8/19
	Empty Result	6/19	6/19
	Incorrect Result	1/19	2/19
Non-Executable	General Syntax Error	1/19	0/19
	Input Type Error	0/19	0/19
	Unknown Function	4/19	0/19

C3-SQL generated queries for all 19 competency questions, of which only 11 were executable. Among these, none returned correct answers, with nine queries producing empty result sets – the remaining eight faced execution errors, primarily due to type mismatches or incorrect filtering conditions. In contrast, ReFoRCE successfully generated executable queries for 18 questions, though 12 produced incorrect results. Due to ReFoRCE's internal mechanism preventing empty results, all queries yielded non-empty outputs, albeit often incorrect. The zero-shot text-to-SQL approach had only one query fail due to syntax errors, but nine executed queries returned empty results. Additionally, seven queries were executed, but were semantically incorrect or returned inaccurate results, leaving only two entirely correct.

The error distribution of the investigated text-to-SPARQL approaches is summarized in Table 8. The zero-shot text-to-SPARQL method provided 13 executable queries, two of which were incorrect, and five returned the correct answer. The remaining six returned empty results, primarily due to incorrect domain class hierarchy level choice and the absence of OPTIONAL clauses (i.e., query parts that should only be evaluated when the data are present) to handle missing data properties. From the queries that were not executable, there was one query with a syntax error and four that used incorrect or unsupported SPARQL functions. The semantic agent, equipped with a syntax checker, was developed to minimize those errors that are most common execution errors (i.e., syntactic errors). If the syntax errors are not fixed or the LLM does not manage to fix them, the model keeps trying to fix them until a maximum retry limit is reached. Cases where this limit is reached and thus do not have a syntax-error-free query are skipped from being executed. In our experiment, this occurred for three questions. A SPARQL query with no syntax errors was generated for all the remaining 16 questions, eight of which provided the correct answer. The number of incorrect results was reduced to 2, but six queries still provided empty results, mainly due to the same issues mentioned above in the zero-shot approach.

5 Discussion

Our results reveal a clear distinction between classic text-to-SQL and semantically enriched text-to-SPARQL approaches. While zero-shot SQL answers only 10.5% of our competency questions correctly, zero-shot SPARQL improves this accuracy to 36.8%, even outperforming the evaluated leading text-to-SQL systems (C3-SQL and ReFoRCE). Our semantic agent further improves this to 42.1%, eliminating syntactically invalid queries. While C3-SQL attempts to enhance accuracy by generating and evaluating multiple candidate queries, ReFoRCE pushes this strategy to its limits, exemplifying a “more is better” principle. Its extensive, feedback-driven query generation improves accuracy on the Outdooractive database sample. However, none of the text-to-SQL approaches was able to catch up with the semantic text-to-SPARQL approaches. In the following subsections, we further interpret our results, develop their theoretical implications, and derive practical guidance for applied tourism information systems.

5.1 Discussion of results

The empirical picture illustrates both the potentials and current limitations of the investigated methods. Overall, our observations can be grouped into three main areas: a clear dependence of accuracy on query categories, a strong sensitivity of LLMs to variations in prompts, and substantial differences in agent tasks and control flow.

5.1.1 Impact of query categories on accuracy

Many semantic errors can be traced back to vague input questions. Since the competency questions in our evaluation were developed with non-technical tourism experts, they were deliberately designed to reflect how outdoor tourists typically phrase their queries, often with a certain degree of ambiguity and imprecision. Our evaluation reveals apparent differences across the categories of the investigated competency questions (i.e., occupancy, peak time analysis, proximity, and combined questions).

Across all methods, questions in the occupancy category were handled most effectively. Notably, the semantic text-to-SPARQL approaches benefited from the rich structure of the occupancy ontology, achieving an accuracy of 54.5% (zero-shot) and 63.6% (semantic agent), respectively. Questions categorized as peak time analysis represent a more challenging subset of occupancy-related queries, often involving comparisons between different occupancy values. Only the semantic approaches could answer questions from this category, underlining the positive effects of an ontological abstraction.

Unlike the other categories, proximity questions (e.g., searching hiking tours around a specified location) do not include occupancy information. They are thus not covered by the TOO but depend entirely on the imported ODTA ontology. Surprisingly, although these questions may appear simple at first glance, none of the approaches provided an answer to any of them. A similar pattern emerged in the combined category, which included proximity elements and resulted in zero correct answers. For the semantic models, this limitation is likely related to the fact that

ODTA was not adapted for use with large language models. Interestingly, our experiments also suggested that simpler, less expressive ontologies sometimes led to more stable and accurate results than complex, highly descriptive ones.

5.1.2 LLM input sensitivity and prompt engineering

Another important observation concerns large language models' high sensitivity to seemingly minor input variations (Salinas and Morstatter 2024). In our experiments, even slight lexical changes within the ontology, such as rewording labels or comments, led to noticeable fluctuations in query performance. For instance, changing the word order or introducing synonyms in natural-language explanations for concepts such as the confidence of an occupancy measurement (i.e., the probability that a measured or predicted occupancy value is correct) significantly impacted the model's interpretation and the resulting queries. Moreover, we found that LLMs frequently ignored formulated instructions, regardless of their position within the prompt, which is contrary to earlier observations (Liu et al. 2024). For example, recommendations to include specific OPTIONAL clauses or to avoid certain hierarchy levels were often disregarded, resulting in empty query outputs.

5.1.3 Agent tasks and control flow

Our experiments further highlight a fundamental challenge in dealing with semantic errors. Even though all queries generated by the semantic agent were syntactically valid (i.e., they can be executed by the database without raising technical errors), the overall accuracy improved only to 42.1%, highlighting that ensuring syntactic validity does not guarantee semantic accuracy. As part of the error correction and refinement task, we initially attempted to mitigate this with an external critic component, acting as an evaluator both before execution (to reduce the number of database calls) and after execution (to catch implausible answers). While this agent often articulated sound reasoning and identified relevant issues, the critic mechanism frequently led to overthinking: the agent would get stuck in repetitive loops of critique, repeatedly offering the same suggestions despite explicit instructions not to do so. Similarly, overly critical feedback often misjudges the complexity of a situation. For instance, when asked to determine the occupancy of a point of interest, our initial critic component unnecessarily suggested analyzing nearby parking places instead of simply checking whether the point of interest provides its own occupancy data.

5.2 Theoretical implications

These findings bring up several theoretical implications. First, the consistent superiority of semantic approaches in terms of accuracy in our evaluation suggests that abstracting technical details and focusing on domain-specific concepts simplifies the creation of correct queries. This mirrors the original motivation of the OBDA paradigm for humans and appears to extend naturally to LLMs. This holds even though we provided the database schemas with natural language comments of comparable detail. However, the following still applies to current LLMs: simpler ontolo-

gies likely outperform highly expressive ones in terms of understanding through the LLM. For tourism, this implies that complex domain constructs (e.g., multi-level occupancy with uncertainty, seasonality, or proximity) must be presented as simplified as possible, which requires a high level of domain expertise. This also reinforces the rationale of some existing ontology-engineering methods in tourism that emphasize lightweight design and the involvement of domain experts (Şimşek et al. 2023), while adding a new view through the lens of LLM-based understanding.

Second, natural language annotations become first-class modeling elements. Our findings suggest that annotations (natural language labels and comments originally intended for documentation purposes) are particularly relevant for LLMs' understanding of ontologies, which remains consistent with Hertling and Paulheim (2023). The lack of an LLM-optimized annotation design in ODTA likely contributed to poor performance on proximity and combined questions. This adds a new perspective to the discussion of touristic ontology engineering: as LLMs gain importance, natural language annotations function as first-class modeling elements. However, approaches such as Chessa et al. (2023) illustrate that tourism ontologies still regard comments primarily as metadata, rather than as natural-language descriptions for semantic interpretation – a perspective that has yet to be incorporated.

Third, input sensitivity of LLMs presents both a challenge and an opportunity. Our findings confirm earlier insights (Salinas and Morstatter 2024) that even seemingly small changes in prompts can have a decisive influence on the outcome of an LLM. With ontologies effectively becoming part of the prompt, this sensitivity is amplified through the impact of free-text annotations. Although this sensitivity complicates reproducibility and robustness, it also constitutes a flexible point of intervention: ontologies can be adapted and finetuned without altering the underlying database schema (Sequeda et al. 2023; Xiao et al. 2019). By contrast, text-to-SQL approaches remain tightly coupled to static schemas that are far less adaptable. Thus, semantic approaches create a broader space for prompt-level variation, supporting controlled flexibility and conceptual reframing in LLM-based query generation.

Fourth, agentic architectures can introduce elements of instability. While agents often enhance reasoning power and adaptability compared to purely prompt-based approaches, their increased autonomy interacts with input sensitivity and the inherent non-determinism of LLMs (Ouyang et al. 2025), which can produce unstable control flows. In this context, we observed that LLMs sometimes ignore explicit instructions. In contrast to previous observations (Liu et al. 2024), this did not depend on the placement within the prompt. Moreover, in line with previous work (Chiang and Lee 2024), our experiments revealed a tendency to overthink and overestimate situation complexity, posing a significant challenge for agentic systems. In tourism contexts, such instability is particularly critical, given the increasing decision support offered by touristic chatbots (Cruz et al. 2025) to visitor safety, congestion management, and route planning, where a balance between adaptability and reliability is essential (Mich and Garigliano 2023).

5.3 Practical implications

From a practical perspective, grounded in our results, three key areas emerge. First, ontology engineering processes should evolve for the LLM era in tourism information systems. Ontologies should no longer be viewed as purely logical constructs consisting of concepts, relations, and constraints. LLMs primarily interpret unstructured text, which means that annotations, which were originally intended for documentation purposes, are becoming more operationally relevant. Consequently, the early involvement of LLM experts should become standard practice. At the same time, validation and verification processes should be adapted to include early LLM-based checks, since even small wording choices in annotations can influence model behavior. Simpler, well-structured ontologies often yield more reliable outcomes than complex and deeply nested ones in LLM-based systems.

Second, the greater the degree of freedom an agent has, the more unpredictable its behavior becomes, reinforced by non-determinism and input sensitivity. Practitioners should translate our theoretical findings into actionable countermeasures. For example, agents should be constrained as far as the use case reasonably allows; multiple internal checks and guardrails within the agent architecture may be useful. To further increase the reliability of an agent, finetuning approaches should also be considered. Ultimately, extensive testing is essential to prevent unwanted looping or overthinking.

Third, efficiency and cost management should guide system design. Feedback-driven text-to-SQL systems, such as ReFoRCE, achieve higher accuracy only at extreme computational costs. In our experiment, ReFoRCE generated 84.8 database queries per question on average, requiring nearly one million tokens. In practice, such resource demands are rarely sustainable. Not only do they lead to high costs, but they also cause long response times and require substantial computing resources. What may work in edge cases (e.g., analytical workloads) is unsuitable for real-world, visitor-facing use cases, where tourists use a chatbot for fast and efficient information retrieval. By contrast, our results indicate that ontology-centric architectures may offer a more favorable balance between accuracy, operational cost, and latency and should be considered as an alternative. While currently observed accuracies are far from production-ready, we expect that accuracies can be significantly improved through further tailoring to the specific use case.

6 Conclusion, limitations, and future research

This paper investigated natural language querying of tourism occupancy data, enabling tourists and tourism professionals to efficiently access complex occupancy and overcrowding information through large language models. Performing a comparative evaluation of performance and cost trade-offs of traditional and semantically enriched text-to-query systems, our results indicate that while zero-shot SQL achieved only 10.5% accuracy, zero-shot SPARQL increased this to 36.8%, and the semantic agent further improved it to 42.1%, outperforming evaluated leading text-to-SQL systems like ReFoRCE at a fraction of the cost. In addition, the semantic

agent generated fewer queries, produced no syntax errors, and imposed a lower database load.

Like other work, this study has several limitations to consider when interpreting the results. First, the evaluation is based on a single ontology and 19 expert-crafted competency questions that reflect common patterns in outdoor tourism. While grounded in domain knowledge, such a setup cannot capture the full variety of queries encountered in real-world systems, which would require a deployed interface to collect user queries at scale as well as broader ontological coverage. Second, while synthetic occupancy data improves the coverage of relevant cases, it comes at the cost of reduced realism. Third, we evaluated all methods based on a static database snapshot, with minor changes to ontology labels or schema annotations leading to substantial shifts in accuracy. Despite efforts to harmonize helper annotations across both text-to-SQL and text-to-SPARQL methods, perfect equivalence was not possible, as each family of methods consumes guidance fundamentally differently. The comparison is, therefore, indicative rather than definitive.

Since one of the main challenges lies in the input sensitivity of LLMs, future work should focus on identifying factors that increase or reduce prompt robustness in terms of LLM interpretability, with particular attention to annotation design and ontology simplification. Finetuning could be explored as a complementary strategy to address this issue. Moreover, balancing critical reasoning with pragmatic performance remains a key challenge, especially in agentic systems. Future work should explore strategies that enable LLMs to optimize both aspects effectively while maintaining predictable control flows. Lastly, integrating query generation with advanced methods such as RAG could be promising, potentially improving entity recognition (e.g., walking tours, POIs) even when ontological descriptions are limited. Ultimately, we view this direction as a promising foundation for advancing toward scalable, intelligent tourism information systems that provide real-time, personalized, and semantically grounded access to quantitative data.

Author contributions S.N. conceptualized the research, wrote the main manuscript text, and did part of the literature review; S.N. and R.R. developed the semantic agent and investigated the results, whereby L.G. did the text-to-SQL experiments with valuable support from S.N. and R.R.; M.W. did part of the literature review; A.H. and R.K. supported by editing and reviewing the text; H.K. supported the study by reviewing and supervision.

Funding Open Access funding enabled and organized by Projekt DEAL. This study was supported by the AIR research project (67KI21005B), funded by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV).

Data availability The data supporting the findings of this study is available from the authors upon reasonable request. However, access is restricted due to licensing agreements with Outdooractive AG, and therefore the data is not publicly available. Requests for access will require approval from Outdooractive AG.

Declarations

Competing interests Stefan Neubig, Linus Göhl, and Rahul Radhakrishnan work for Outdooractive AG, a company that, among other things, offers visitor management services. The remaining authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abu N, Syahidah WN, Afif MM, Nordin SZ (2021) SARIMA and Exponential smoothing model for forecasting ecotourism demand: a case study in national Park Kuala Tahan, Pahang. *J Phys Conf Ser* 1988(1):012118. <https://doi.org/10.1088/1742-6596/1988/1/012118>
- Askari A, Poelitz C, Tang X (2024) MAGIC: generating self-correction guideline for in-context text-to-SQL (No. arXiv: 2406.12692). arXiv. <https://doi.org/10.48550/arXiv.2406.12692>
- Banerjee A, Mahmudov T, Adler E, Aisyah FN, Wörndl W (2025) Modeling sustainable city trips: integrating CO2e emissions, popularity, and seasonality into tourism recommender systems. *Inf Technol Tour* 27(1):189–226. <https://doi.org/10.1007/s40558-024-00303-1>
- Benner M (2020) Overcoming overtourism in Europe: towards an institutional-behavioral research agenda. *Z Für Wirtschaftsgeographie* 64(2):74–87. <https://doi.org/10.1515/zfw-2019-0016>
- Bi J-W, Li C, Xu H, Li H (2021) Forecasting daily tourism demand for tourist attractions with big data: an ensemble deep learning method. *J Travel Res*. <https://doi.org/10.1177/00472875211040569>
- Bollenbach J, Neubig S, Hein A, Keller R, Krcmar H (2024) Enabling active visitor management: local, short-term occupancy prediction at a touristic point of interest. *Inf Technol Tour* 26(3):521–552. <https://doi.org/10.1007/s40558-024-00291-2>
- Butler R (2018) Challenges and opportunities. *Worldwide Hosp Tour Themes* 10(6):635–641. <https://doi.org/10.1108/WHATT-07-2018-0042>
- Butler RW, Dodds R (2022) Overcoming overtourism: a review of failure. *Tour Rev* 77(1):35–53. <https://doi.org/10.1108/TR-04-2021-0215>
- Calvanese D, Cogrel B, Komla-Ebri S, Kontchakov R, Lanti D (2015) Ontop: answering SPARQL queries over relational databases. *Semant Web J* <https://doi.org/10.3233/SW-160217>
- Capocchi A, Vallone C, Pierotti M, Amaduzzi A (2019) Overtourism: a literature review to assess implications and future perspectives. *Sustainability* 11(12):3303. <https://doi.org/10.3390/su1123303>
- Cen J, Liu J, Li Z, Wang J (2025) SQLFixAgent: towards semantic-accurate text-to-SQL parsing via consistency-enhanced multi-agent collaboration (No. arXiv: 2406.13408). arXiv. <https://doi.org/10.48550/arXiv.2406.13408>
- Chawathe SS (2019) Using historical data to predict parking occupancy. In 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp 0534–0540. <https://doi.org/10.1109/UEMCON47517.2019.8993084>
- Chen X, Lin M, Schärli N, Zhou D (2023) Teaching large language models to Self-debug (No. arXiv: 2304.05128). arXiv. <https://doi.org/10.48550/arXiv.2304.05128>
- Chessa A, Fenu G, Motta E, Osborne F, Reforgiato Recupero D, Salatino A, Secchi L (2023) Data-driven methodology for knowledge Graph generation within the tourism domain. *IEEE Access* 11:67567–67599. <https://doi.org/10.1109/ACCESS.2023.3292153>
- Chiang C-H, Lee H (2024) Over-reasoning and redundant calculation of large language models (Version 2). arXiv. <https://doi.org/10.48550/ARXIV.2401.11467>
- Cossatin AG, Mauro N, Ferrero F, Ardissono L (2025) Tell me more: integrating LLMs in a cultural heritage website for advanced information exploration support. *Inf Technol Tour* 27(2):385–416. <https://doi.org/10.1007/s40558-025-00312-8>
- Cruz M, Jardim B, Neto MDC (2025) Lisa: a touristic chatbot for Lisbon. *Information Technology & Tourism*. <https://doi.org/10.1007/s40558-025-00339-x>
- Deng M, Ramachandran A, Xu C, Hu L, Yao Z, Datta A, Zhang H (2025) ReFoRCE: a text-to-SQL agent with Self-refinement, consensus enforcement, and column exploration. arXiv. <https://doi.org/10.48550/arXiv.2502.00675>

- Deng X, Awadallah AH, Meek C, Polozov O, Sun H, Richardson M (2021) Structure-grounded pretraining for text-to-SQL. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 1337–1350. <https://doi.org/10.18653/v1/2021.naacl-main.105>
- Dong X, Zhang C, Ge Y, Mao Y, Gao Y, Chen L, Lin J, Lou D (2023) C3: zero-shot text-to-SQL with ChatGPT (No. arXiv: 2307.07306). arXiv. <https://doi.org/10.48550/arXiv.2307.07306>
- Dwivedi YK, Pandey N, Currie W, Micu A (2024) Leveraging ChatGPT and other generative artificial intelligence (AI)-based applications in the hospitality and tourism industry: practices, challenges and research agenda. *Int J Contemp Hosp Manag* 36(1):1–12. <https://doi.org/10.1108/IJCHM-05-2023-0686>
- Edge D, Trinh H, Cheng N, Bradley J, Chao A, Mody A, Truitt S, Metropolitansky D, Ness RO, Larson J (2025) From Local to global: a Graph RAG approach to query-focused summarization (No. arXiv: 2404.16130). arXiv. <https://doi.org/10.48550/arXiv.2404.16130>
- Ehrlinger L, Wöß W (2016) Towards a definition of knowledge graphs 4
- FIWARE Foundation (2022) Smart data models. <https://www.fiware.org/smart-data-models/>
- Fu H, Liu C, Wu B, Li F, Tan J, Sun J (2023) *CatSQL*: towards real world natural language to SQL applications. *Proc The VLDB Endow* 16(6):1534–1547. <https://doi.org/10.14778/3583140.3583165>
- Gao D, Wang H, Li Y, Sun X, Qian Y, Ding B, Zhou J (2023) Text-to-SQL empowered by large language models: a benchmark evaluation (No. arXiv: 2308.15363). arXiv. <https://doi.org/10.48550/arXiv.2308.15363>
- Gao Y, Liu Y, Li X, Shi X, Zhu Y, Wang Y, Li S, Li W, Hong Y, Luo Z, Gao J, Mou L, Li Y (2025) A preview of XiYan-SQL: a multi-generator ensemble framework for text-to-SQL (No. arXiv: 2411.08599). arXiv. <https://doi.org/10.48550/arXiv.2411.08599>
- Gao Y, Xiong Y, Gao X, Jia K, Pan J, Bi Y, Dai Y, Sun J, Wang M, Wang H (2024) Retrieval-augmented generation for large language models: a survey (No. arXiv: 2312.10997). arXiv. <https://doi.org/10.48550/arXiv.2312.10997>
- Guarino N, Oberle D, Staab S (2009) What is an ontology? In: Staab S, Studer R (Hrsg.), *Handbook on ontologies*. Springer, Berlin Heidelberg, pp S. 1–17. https://doi.org/10.1007/978-3-540-92673-3_0
- Guo Z, Ma C, Sun Y, Zhao T, Wang G, Huang H (2025) Evaluating and enhancing LLMs for multi-turn text-to-SQL with multiple question types (No. arXiv: 2412.17867). arXiv. <https://doi.org/10.48550/arXiv.2412.17867>
- Guo Z, Xia L, Yu Y, Ao T, Huang C (2025) LightRAG: simple and fast retrieval-augmented generation (No. arXiv: 2410.05779). arXiv. <https://doi.org/10.48550/arXiv.2410.05779>
- Hertling S, Paulheim H (2023) OLaLa: ontology matching with large language models. In Proceedings of the 12th Knowledge Capture Conference 2023, pp 131–139. <https://doi.org/10.1145/3587259.3627571>
- Hogan A, Blomqvist E, Cochez M, D’amato C, Melo GD, Gutierrez C, Kirrane S, Gayo JEL, Navigli R, Neumaier S, Ngomo A-CN, Polleres A, Rashid SM, Rula A, Schmelzeisen L, Sequeda J, Staab S, Zimmermann A (2022) Knowledge graphs. *ACM Comput Surv* 54(4):1–37. <https://doi.org/10.1145/3447772>
- Şimşek U, Kärle E, Angele K, Huaman E, Opdenplatz J, Sommer D, Umbrich J, Fensel D (2023) A knowledge Graph perspective on knowledge engineering. *SN Comput Sci* 4(1). Scopus. <https://doi.org/10.1007/s42979-022-01429-x>
- Imielinski T, Lipski W (1984) On the undecidability of equivalence problems for relational expressions. In: Gallaire H, Minker J, Nicolas JM (Hrsg.), *Advances in data base theory*, vol 2. Springer US, pp 393–409. https://doi.org/10.1007/978-1-4615-9385-0_13
- Jiao EX, Chen JL (2019) Tourism forecasting: a review of methodological developments over the last decade. *Tour Econ* 25(3):469–492. <https://doi.org/10.1177/1354816618812588>
- Kang S-E, Kim MJ, Kim JS, Olya H (2024) Can I trust GenAI to Plan My Next Trip? A multi-method approach to optimizing media mix. *J Travel Res* 00472875241305630. <https://doi.org/10.1177/00472875241305630>
- Keller R, Neubig S, Rebholz D, Schmücker D (2024) How many tourists Next Weekend? Conceptualizing AI-based forecasts for visitor management in tourism. *Mark Rev St. Gallen*
- Kärle E, Simsek U, Akbar Z, Hepp M, Fensel D (2017) Extending the Schema.Org vocabulary for more expressive accommodation annotations. In: Schegg R, Stangl B (Hrsg.), *Information and communication technologies in tourism 2017*. Springer International Publishing, pp S. 31–41. https://doi.org/10.1007/978-3-319-51168-9_3

- Lee D, Park C, Kim J, Park H (2024) MCS-SQL: leveraging multiple prompts and multiple-choice selection for text-to-SQL generation (No. arXiv: 2405.07467). arXiv. <https://doi.org/10.48550/arXiv.2405.07467>
- Lei F, Chen J, Ye Y, Cao R, Shin D, Su H, Suo Z, Gao H, Hu W, Yin P, Zhong V, Xiong C, Sun R, Liu Q, Wang S, Yu T (2025) Spider 2.0: Evaluating language models on real-world Enterprise text-to-SQL Workflows (No. arXiv: 2411.07763). arXiv. <https://doi.org/10.48550/arXiv.2411.07763>
- Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih W, Rocktäschel T, Riedel S, Kiela D (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. In Proceedings of the 34th International Conference on Neural Information Processing Systems
- Li H, Zhang J, Liu H, Fan J, Zhang X, Zhu J, Wei R, Pan H, Li C, Chen H (2024) CodeS: towards building open-source language models for text-to-SQL. *Proc The ACM Manag Data* 2(3):1–28. <https://doi.org/10.1145/3654930>
- Li J, Hui B, Qu G, Yang J, Li B, Li B, Wang B, Qin B, Geng R, Huo N, Zhou X, Ma C, Li G, Chang KCC, Huang F, Cheng R, Li Y (2023) Can LLM already serve as a database interface? A big bench for large-scale database grounded text-to-SQLs. In Proceedings of the 37th International Conference on Neural Information Processing Systems
- Liu NF, Lin K, Hewitt J, Paranjape A, Bevilacqua M, Petroni F, Liang P (2024) Lost in the middle: how language models use long contexts. *Trans Assoc Comput Linguistics* 12:157–173. https://doi.org/10.1162/tacl_a_00638
- Luo X, Xu D, Li Y (William), & Wan LC (2025) Advancing information search through GenAI: the roles of search type, travel motive and GenAI customization level. *Int J Contemp Hosp Manag* 37(5):1725–1743. <https://doi.org/10.1108/IJCHM-06-2024-0941>
- Maamari K, Abubaker F, Jaroslawicz D, Mhedhbi A (2024) The death of schema linking? text-to-SQL in the age of well-reasoned language models (No. arXiv: 2408.07702). arXiv. <https://doi.org/10.48550/arXiv.2408.07702>
- Mai C, Tal R, Mohamed T (2024) Learning metadata-agnostic representations for text-to-SQL In-context example selection (No. arXiv: 2410.14049). arXiv. <https://doi.org/10.48550/arXiv.2410.14049>
- Mich L, Garigliano R (2023) ChatGPT for e-tourism: a technological perspective. *Inf Technol Tour* 25(1):1–12. <https://doi.org/10.1007/s40558-023-00248-x>
- Neubig S, Radhakrishnan R, Göhl L, Loges R, Polgar M, Hein A, Krcmar H (2025) Boosting the Querying Accuracy of Multi-Level Occupancy Data with Ontology-Guided LLMs. In: Nixon L, Tuomi A, O'Connor P (eds) *Information and Communication Technologies in Tourism 2025*. Springer Proceedings in Business and Economics. Springer, Cham. https://doi.org/10.1007/978-3-031-83705-0_5
- Neubig S, Rebholz D, Hein A, Keller R, Krcmar H (2023) To graph or not to graph: the missing pieces for knowledge graphs in sustainable tourism. *INFORMATIK (NaWerSys), Bonn. INFORMATIK 2023*
- Ouyang S, Zhang JM, Harman M, Wang M (2025) An empirical study of the non-determinism of ChatGPT in code generation. *ACM Trans Softw Eng Methodol* 34(2):1–28. <https://doi.org/10.1145/3697010>
- Phumchusri N, Suwatanapongched P (2021) Forecasting hotel daily room demand with transformed data using time series methods. *J Revenue Pricing Manag*. <https://doi.org/10.1057/s41272-021-00363-6>
- Pourreza M, Rafiei D (2023) DIN-SQL: decomposed In-context learning of text-to-SQL with self-correction (No. arXiv: 2304.11015). arXiv. <https://doi.org/10.48550/arXiv.2304.11015>
- Pourreza M, Rafiei D (2024) DTS-SQL: decomposed text-to-SQL with small large language models (No. arXiv: 2402.01117). arXiv. <https://doi.org/10.48550/arXiv.2402.01117>
- Pourreza M, Sun R, Li H, Miculicich L, Pfister T, Arik SO (2024) SQL-GEN: bridging the dialect gap for text-to-SQL via synthetic data and model merging (No. arXiv: 2408.12733). arXiv. <https://doi.org/10.48550/arXiv.2408.12733>
- Qin Y, Chen C, Fu Z, Chen Z, Peng D, Hu P, Ye J (2025) ROUTE: robust multitask tuning and collaboration for text-to-SQL (No. arXiv: 2412.10138). arXiv. <https://doi.org/10.48550/arXiv.2412.10138>
- Salinas A, Morstatter F (2024) The butterfly effect of altering prompts: how small changes and jailbreaks affect large language model performance. *Findings Assoc Comput Linguist ACL 2024*:4629–4651. <https://doi.org/10.18653/v1/2024.findings-acl.275>
- Schema.org Community Group (2022) Schema.org-Schema.org. <https://schema.org/>
- Schmücker D, Reif J, Horster E, Engelhardt D, Höftmann N, Naschert L, Radlmayr C (2023) The INPREs intervention escalation framework for avoiding overcrowding in tourism destinations. *Tour Hosp* 4(2):282–292. <https://doi.org/10.3390/tourhosp4020017>
- Sequeda J, Allemang D, Jacob B (2023) A benchmark to understand the role of knowledge graphs on large language Model's accuracy for question answering on enterprise SQL databases (Version 1). arXiv. <https://doi.org/10.48550/ARXIV.2311.07509>

- Shen K, Kejriwal M (2024) SeLECT-SQL: self-correcting ensemble Chain-of-Thought for text-to-SQL (No. arXiv: 2409.10007). arXiv. <https://doi.org/10.48550/arXiv.2409.10007>
- Suárez de Figueroa MDC (2010) Neon methodology for building ontology networks: specification, scheduling and reuse. Universidad Politécnica de Madrid. <https://doi.org/10.1016/j.landurbplan.2011.04.007>
- Talaei S, Pourreza M, Chang Y-C, Mirhoseini A, Saberi A (2024) CNESS: contextual harnessing for efficient SQL synthesis (No. arXiv: 2405.16755). arXiv. <https://doi.org/10.48550/arXiv.2405.16755>
- Tsang WK, Benoit DF (2020) Gaussian processes for daily demand prediction in tourism planning. *J Forecasting* 39(3):551–568. <https://doi.org/10.1002/for.2644>
- Wang Y, Lipka N, Rossi RA, Siu A, Zhang R, Derr T (2024) knowledge Graph prompting for multi-document question answering. *Proc The AAAI Conf Artif Intel* 38(17):19206–19214. <https://doi.org/10.1609/aaai.v38i17.29889>
- Xia H, Jiang F, Deng N, Wang C, Zhao G, Mihalcea R, Zhang Y (2024) SQL-CRAFT: text-to-SQL through interactive refinement and Enhanced reasoning (No. arXiv: 2402.14851). arXiv. <https://doi.org/10.48550/arXiv.2402.14851>
- Xiao G, Ding L, Cogrel B, Calvanese D (2019) Virtual knowledge graphs: an overview of systems and use cases. *Data Intel* 1(3):201–223. https://doi.org/10.1162/dint_a_00011
- XLANG Lab (2025) Spider 2.0. <https://spider-sql.github.io/>
- Yale-LILY (2025) Spider: Yale semantic Parsing and text-to-SQL challenge. <https://yale-lily.github.io/spider>
- Yu T, Zhang R, Yang K, Yasunaga M, Wang D, Li Z, Ma J, Li I, Yao Q, Roman S, Zhang Z, Radev D (2018) Spider: a large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp 3911–3921. <https://doi.org/10.18653/v1/D18-1425>
- Zhang H, Cao R, Chen L, Xu H, Yu K (2023) ACT-SQL: In-context Learning for text-to-SQL with automatically-generated Chain-of-Thought (No. arXiv: 2310.17342). arXiv. <https://doi.org/10.48550/arXiv.2310.17342>
- Zhang Z, Fang M, Chen L, Namazi-Rad M-R, Wang J (2023) How do large language models capture the ever-changing world knowledge? A review of recent Advances. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp 8289–8311. <https://doi.org/10.18653/v1/2023.emnlp-main.516>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Stefan Neubig^{1,2}  · **Linus Göhl**² · **Rahul Radhakrishnan**²  · **Matthias Weirich**¹ · **Andreas Hein**³  · **Robert Keller**⁴  · **Helmut Krcmar**¹ 

✉ Stefan Neubig
stefan.neubig@tum.de

Linus Göhl
linus.goehl@outdooractive.com

Rahul Radhakrishnan
rahul.radhakrishnan@outdooractive.com

Matthias Weirich
matthias.weirich@tum.de

Andreas Hein
andreas.hein@unisg.ch

Robert Keller
robert.keller@hs-kempten.de

Helmut Krcmar
helmut.krcmar@tum.de

¹ School of Computation, Information and Technology, Technical University of Munich, Boltzmannstraße 3, 85748 Garching, Germany

² Outdooractive AG, Missener Straße 18, 87509 Immenstadt, Germany

³ University of St. Gallen, Müller-Friedberg-Strasse 6/8, 9000 St. Gallen, Switzerland

⁴ Kempten University of Applied Sciences, Bahnhofstraße 61, 87435 Kempten (Allgäu), Germany