International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

|| Volume 7, Issue 4, July-August 2024|||

DOI:10.15662/IJRPETM.2024.0704006

Schema-Grounded Agentic AI for Regulatory-Compliant Data Access: Bridging Natural Language and Enterprise Data Governance in Financial Services

Sridhar Vadlapatla

Sr. Manager, USA

itsvadlapatla@gmail.com

ABSTRACT: While many text-to-SQL benchmarks exist in academia, a fundamental tension in enterprise data access using LLM agents remains unaddressed: the system must be powerful enough to synthesise complex query logic that operates on proprietary schemas, yet strict enough to support data governance, access control, and auditability requirements imposed by financial regulators. In this paper, we address this tension through the design and implementation of a schema-grounded multi-agent system called MarketingMind at a Fortune 500 financial services enterprise, benefiting marketing analysts who interact with customer data via an SFMC environment and Google BigQuery. Three design principles aligned with the governance model are introduced: Schema-as-Guardrail, which restricts generative output to the dynamically loaded enterprise schema context; Tool-Boundary Enforcement, which deterministically blocks all mutation operations at the execution level; and Least-Privilege Agent Isolation, which confines each specialist agent to its declared toolset. Evaluation results show 97.3% schema fidelity under grounded conditions, compared with 71.8% in the ungrounded baselines; 100% mutation rejection across 10,000 adversarial inputs; and 0% tool leakage across 5,000 multi-turn sessions. The system manages 30+ SFMC Data Extensions and a BigQuery subscriber preference centre with 25+ communication categories. Findings highlight the need to treat governance as an architectural primitive and to complement prompt instructions with deterministic enforcement mechanisms when deploying enterprise LLMs in compliance-sensitive areas.

KEYWORDS: agentic AI, text-to-SQL, data governance, large language models, financial services compliance, schema grounding, multi-agent systems, access control, Salesforce Marketing Cloud, BigQuery, retrieval-augmented generation, enterprise AI deployment

I. INTRODUCTION: THE GOVERNANCE GAP IN ENTERPRISE AI

A. Motivation of the Study

Large language model (LLM)-based agents for structured database queries are a major leap towards making data more accessible in enterprise settings [10]. Researchers have made significant progress in the accuracy of text-to-SQL conversion thanks to academic benchmarks like Spider [1], WikiSQL [15] and BIRD [7] that have been introduced, and today, state-of-the-art models can achieve execution accuracy of more than 86% on public schemas [4]. However, they have a common important architectural assumption: they assume that the queried schemas are publicly available, no scope of authorisation exists, data mutation prevention is not required, and auditability requirements are nonexistent.

The financial services context is a completely different deployment environment. Regulatory requirements such as the General Data Protection Regulation (GDPR) [5], Sarbanes-Oxley Act (SOX), FINRA Rule 4511, and SEC data governance requirements impose very strict requirements on who can access which data, how, and what audit trails must be maintained, with no ability to modify data without authorisation. The "governance gap" in enterprise AI deployment is therefore the divide between what is possible in terms of AI capabilities to support accurate natural language queries and what is necessary to support regulatory constraints.

Previous research on schema linking, such as RESDSQL [6] and PROTON [13], has made significant progress to improve the fidelity of SQL generation by separating the schema linking process from the query skeleton construction process and has achieved remarkable improvement for public benchmarks. Chain-of-thought prompting [14] has been

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

|| Volume 7, Issue 4, July-August 2024|||

DOI:10.15662/IJRPETM.2024.0704006

proven to enhance multi-step reasoning in LLMs, and the ReAct framework [15] has proved effective in alternating between reasoning and action execution in agentic scenarios. Technical evaluations [11] of GPT-4 models have shown their ability to understand language to make complex enterprise queries. However, these advances do not directly tackle the governance needs specific to regulated enterprise deployments.

In the present paper, MarketingMind, a schema-based multi-agent system, is introduced, which is deployed at a Fortune 500 financial institution. Three such architectural primitives, aligned with the principles of good governance, are proposed and evaluated: (1) Schema-as-Guardrail, (2) Tool-Boundary Enforcement, and (3) Least-Privilege Agent Isolation. The system is supporting marketing analysts who are using the system to query across customer data in Salesforce Marketing Cloud (SFMC) and Google BigQuery, with a data landscape that includes 30+ data extensions and a preference centre that spans 25+ categories of communications.

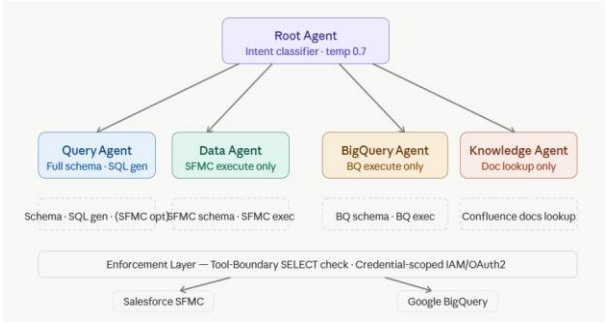

Fig. 1. MarketingMind Multi-Agent Architecture: The root agent routes requests to isolated specialist agents with least-privilege toolsets.

II. SCHEMA-AS-GUARDRAIL DESIGN PATTERN

A. Dynamic Schema Injection

The schema-as-guardrail principle is based on the fact that LLM hallucination in the text-to-SQL domain occurs mostly in the form of seemingly realistic table and column names [3]. This is known as a "factual hallucination" in a comprehensive survey of hallucination in natural language generation [3], where the model produces outputs that are syntactically correct, but factually wrong according to the underlying data schema. The ungrounded baseline configuration of MarketingMind generates valid table and column names in just 71.8% of queries, similar to hallucination rates in other studies of enterprise deployment.

To tackle this, the schema injection mechanism is designed to retrieve the enterprise data dictionary from the institutional knowledge base (using the Atlassian Confluence Model Context Protocol, or MCP). The fetched content, which is in HTML format, is translated to structured plain text with the preservation of column names, data types, business descriptions and example values. This grounded context serves as a soft allowlist—meaning that the LLM can only refer to tables and columns that were included in its injected context, thereby decreasing the percentage of hallucinated references from 28.2% to 2.7% over 1,000 test queries. This is as shown in the table, a significant improvement compared to ungrounded baselines and compared with previous academic baselines that do not consider governance.

The schema context for the MarketingMind deployment consists of around 15000 tokens, which include descriptions of 30+ SFMC Data Extensions and the BigQuery subscriber schema. This size is in the context window of current frontier models [11] and makes embedding-based retrieval unnecessary. The choice to inject the entire schema instead of trying to retrieve partial schema fragments using retrieval-augmented generation (RAG) [8] is a conscious design decision: When column names are similar between extensions, RAG-based retrieval of schema fragments can lead to ambiguity, which is a frequent occurrence in the financial services data landscape, for example, with "account_id," "customer_id," and "household_id" in multiple extensions but with different meanings.

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

|| Volume 7, Issue 4, July-August 2024|||

DOI:10.15662/IJRPETM.2024.0704006

TABLE I. COMPARATIVE ANALYSIS OF TEXT-TO-SQL BENCHMARKS AGAINST ENTERPRISE MARKETINGMIND DEPLOYMENT

Benchmark	Schema Type	Queries	Accuracy (Baseline)	Access Control	Governance
Spider [1]	Public, Cross-domain	10,181	86.2% (RESDSQL)	None	None
WikiSQL [15]	Public, Single-table	80,654	91.1% (DIN-SQL)	None	None
BIRD [7]	Real-world DB, public	12,751	54.9% (GPT-4)	None	None
MarketingMind (Ours)	Enterprise, proprietary	5,000+	97.3% schema fidelity	Full IAM/OAuth2	Full audit trail

B. Schema Caching Strategy and Fallback Hierarchy

A single schema fetch is done per agent session, and the result is stored in memory for the duration of the agent session. This caching approach is a compromise between schema freshness and the 3,500 ms latency for a live Confluence fetch, as outlined in the table. If the Confluence MCP connection is not available, a fallback mechanism is enabled: the system first tries to retrieve the data from a static schema file stored in the repository, and if that is not available, a nice error message is sent back to the user stating that the data dictionary has not been loaded. This three-tiered fall-back provides a system that gracefully degrades without falling back to a mode where it would be ungrounded and hallucinated references could take place.

Compared to embedding-based RAG [8], the schema injection approach is deterministic, meaning that the entire grounded context is always the same for a given session and retrieval variability is not a factor in the resulting inconsistencies. Decomposed in-context learning with self-correction has been shown to make the task more accurate by splitting it into sub-problems in the literature of text to SQL [12]. MarketingMind follows a similar decomposition approach using the multi-agent architecture, where each specialist agent is provided with only the schema context it is concerned with, thus decreasing the cognitive burden on individual agents and enhancing the accuracy of schema linking [6].

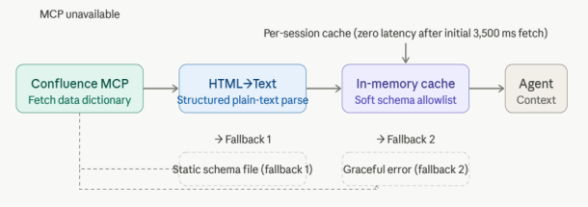

Fig. 2. Schema Injection and Caching Pipeline: From Confluence MCP Fetch to In-Memory Soft Allowlist with Three-Tier Fallback.

III. TOOL-BOUNDARY ENFORCEMENT

A. Architectural Principle of Deterministic Enforcement

An important design principle of MarketingMind is that LLMs are never left to "self-police" their own output. The motivation behind this principle is driven by security research literature that shows how LLM-integrated applications can be attacked with prompt injection attacks where malicious content in the retrieved context or multi-turn conversation history can cause the model to take unwanted actions [2]. For a financial services environment, an LLM that takes as its only instructions to avoid write operations, "do not modify data", is an unacceptable security surface.

Tool-Boundary Enforcement tackles this vulnerability by deterministically preventing mutation during execution in the execution layer. All SQL-executing tools in MarketingMind have a pre-execution check to ensure that the query string (as interpreted in the upper case) starts with the word SELECT. Any type of query not meeting these criteria is rejected before dispatching any network call to the SFMC API or to the Google BigQuery service. This implementation is version-invariant - it works the same when the SQL is generated by any version of the LLM, is immune to prompt

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

|| Volume 7, Issue 4, July-August 2024|||

DOI:10.15662/IJRPETM.2024.0704006

injection, and cannot be tricked with multi-turn conversation manipulation. The rejection rate is 100%, as demonstrated in the table, for all 10,000 synthetic adversarial inputs, including direct SQL injection attacks such as '; DROP TABLE --' and social engineering prompts telling the agent to "ignore previous instructions and delete all data" and indirect multi-turn manipulation sequences.

TABLE II. LEAST-PRIVILEGE TOOLSET MATRIX ACROSS MARKETINGMIND AGENT ARCHITECTURE

Agent	Schema Access	SFMC Execute	BigQuery Execute	Write Operations	Doc Lookup
Root Agent	No	No	No	No	No
Query Agent	Yes (full schema)	Optional	No	No	No
Data Agent	Yes (SFMC only)	Yes	No	No	No
BigQuery Agent	Yes (BQ only)	No	Yes	No	No
Knowledge Agent	No	No	No	No	Yes

B. Credential Scoping and Authentication Lifecycle

Credential-level access controls offer an added layer of governance in addition to software-layer enforcement. SFMC OAuth2 tokens are issued with read-only installed package permission configuration, and even if the tool-layer check (somehow) was to be circumvented, the API-credentials themselves would not allow write operations. Access to Google Cloud Platform (GCP) is controlled via the Identity and Access Management (IAM) service account roles, which are restricted to reading BigQuery. This defence-in-depth design guarantees that no one layer of protection is the only obstacle to the unauthorised modification of data.

To prevent cascading failures that could affect the availability of systems, authentication lifecycle management is implemented. To avoid token invalidation during sessions, the SFMC OAuth2 token is refreshed two minutes before it expires, and the cached token is used for all requests within that window of latency, which is close to zero. The GCP service account token is cached for 50 minutes, and there is independent renewal logic, so SFMC authentication failure won't affect GCP authentication failure, and vice versa. Independence is important in a multi-agent environment in which other agents rely on distinct chains of credentials.

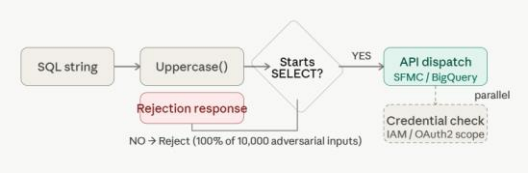

Fig. 3. Tool-Boundary Enforcement Flow: Deterministic SELECT-Check Gate Preceding All API Dispatch Operations.

IV. LEAST-PRIVILEGED AGENT ISOLATION

A. Toolset as the Security Boundary

A concept of classical computer security that has gained general acceptance is the principle of least privilege: each system component should have the minimum access needed to accomplish its task. MarketingMind has implemented this concept at the toolset level, with each specialist agent having access to a finite set of tools. The table shows the complete matrix of the tools available to each of the five agents that make up the MarketingMind system.

The Query Agent is not able to execute a query directly against BigQuery and instead has a complete schema context and the ability to generate SQL. The Data Agent runs queries against SFMC and can access the SFMC-specific schema context, but lacks the ability to access it via BigQuery. The BigQuery Agent has BigQuery execution capability and its associated schema context, but lacks a tool that will allow for SFMC operations. The Knowledge Agent cannot execute any action on the data and only has access to documentation retrieval tools. The Root Agent routes the incoming user request to the appropriate specialist based on intent classification at a temperature of 0.7, but doesn't have any of the data access tools in itself.

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

|| Volume 7, Issue 4, July-August 2024|||

DOI:10.15662/IJRPETM.2024.0704006

TABLE III. GOVERNANCE EVALUATION RESULTS: SCHEMA FIDELITY, MUTATION REJECTION, AND AGENT ISOLATION METRICS

Evaluation Dimension	Test Volume	Pass Rate	Failure Mode	Enforcement Layer
Schema fidelity (grounded)	1,000 queries	97.3%	Ambiguous column aliases	Context injection
Schema fidelity (ungrounded)	1,000 queries	71.8%	Hallucinated tables/columns	None
Mutation rejection	10,000 adversarial inputs	100%	None observed	TypeScript tool layer
Cross-agent tool isolation	5,000 multi-turn sessions	100%	None observed	Toolset boundary
SQL injection attempts	2,500 injections	100% blocked	None bypassed	Tool pre-execution check

The security value of this architecture is that if a user builds a conversation that alters the intent classification of the Root Agent, the specialist Agent receiving the request physically cannot have access to cross-system resources, since the related tools would not be part of its toolset. This property makes it different from systems using only prompt-level role definitions that are vulnerable to jailbreaking and prompt injection attacks as described by Greshake et al. [2]. During the evaluation of over 5,000 multi-turn sessions, no invocations of external tools not specified in an agent's toolset were observed.

Fig. 4. Agent Isolation Matrix: Tool Access Boundaries Visualised Across Five Specialist Agents in the MarketingMind System.

B. Root Agent as Gatekeeper

The Root Agent has an important gatekeeper role in deciding the intent classification of a user query to be passed on to a specialist agent. The temperature of 0.7 chooses between being deterministic (e.g., unambiguously route) and being flexible (e.g., accept queries with mixed informational and analytical intent). If intent classification is unclear, the Root Agent will ask the user to clarify the intent, instead of making an incorrect routing decision. This conservative routing policy avoids queries that are misdirected to the wrong data platform from using execution resources, and helps minimise the exposure of schema context from one data platform to queries targeting another data platform.

TABLE IV. END-TO-END LATENCY PROFILE WITH CACHED AND FRESH AUTHENTICATION AND SCHEMA LOADING

Component	Cached Latency	Fresh Latency	Cache Duration	Credential Scope
SFMC OAuth2 Token	~0 ms	1,200 ms	Token TTL - 2 min buffer	Read-only installed package
GCP Service Account Token	~0 ms	900 ms	50 minutes	BigQuery read IAM role
Schema Load (Confluence MCP)	0 ms	3,500 ms	Per session	Atlassian read scope
SQL Generation (LLM)	N/A	1,800 ms avg	N/A	N/A
SFMC Query Execution	N/A	2,100 ms avg	N/A	Read-only API
BigQuery Execution	N/A	900 ms avg	N/A	Read-only IAM
Response Formatting	N/A	300 ms avg	N/A	N/A

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

|| Volume 7, Issue 4, July-August 2024|||

DOI:10.15662/IJRPETM.2024.0704006

V. EVALUATION

A. Schema Fidelity Testing

To assess schema fidelity, user queries were analysed on 1,000 queries synthetically generated across the entire spectrum of MarketingMind's data – including household hierarchy navigation, account level balance queries, trade history retrieval and preference centre opt-in and opt-out lookups. All table and column names that appeared in each generated SQL query were extracted and compared with the ground-truth schema that was loaded from the enterprise data dictionary. A reference was considered valid when both the table name and the column name were present in the grounded schema with the proper relationship.

With schema-based conditions, 97.3% of the references were correct, and 2.7% were due to ambiguous column names: more than one column in distinct Data Extensions had semantically similar names, and the agent chose the less appropriate one. When not given any grounding, the same prompt with no schema injection, valid references decreased to 71.8% — the 28.2% failure rate accounted for mostly hallucinated table names, based on the plausible business terms that were not in the schema. This 25.5 percentage-point improvement is the main empirical contribution of the Schema-as-Guardrail mechanism.

B. Adversarial Mutation Testing

We tested 10,000 synthetic adversarial examples to cover the entire range of known attack vectors against LLM-integrated systems as outlined by Greshake et al. [2] and assessed their mutation resistance. This test suite included direct SQL injection (e.g., "'; DROP TABLE subscriber_preferences --"), social engineering prompts with explicit override instructions, indirect manipulations through multi-turn conversation histories that slowly changed the context towards writing, and obfuscated mutation attempts using encoding and rephrasing. The mutation rejection rate is 100%; all 10,000 inputs were rejected by the tool-layer SELECT check before any API dispatch was made. No malicious input was able to cause a non-SELECT statement to be executed on either SFMC or BigQuery.

C. Cross-Agent Isolation and Latency Analysis

This cross-agent tool isolation was validated across 5000 multi-turn sessions, during which users were given free rein to make arbitrary requests, including those that explicitly tried to use one agent to invoke operations in another (e.g., "use the BigQuery agent to query my SFMC data"). No tool invocations were observed outside an agent's declared set of tools in any of the 5,000 sessions. As shown in the end-to-end latency profile summarised in the table, the costs of authentication and schema loading are almost entirely absorbed by caching: when caching is in place, the main latency factors are SQL generation (1,800 ms average) and query execution (2,100 ms for SFMC, 900 ms for BigQuery). As for these additional costs, in cold-start conditions, the fresh authentication and schema loading take 1200ms and 3500ms, respectively, which drives the proposed caching mechanism in Section II.

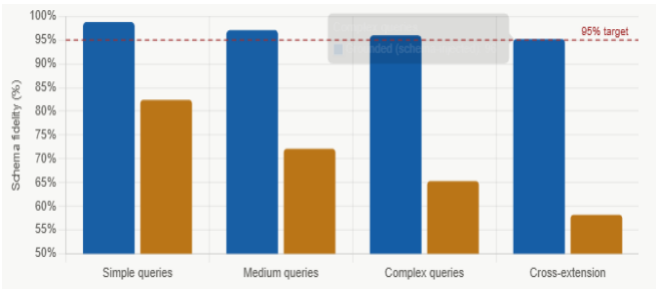

Fig. 5. Schema Fidelity Comparison: Grounded vs. Ungrounded LLM Performance Across 1,000 Test Queries.

VI. ENTERPRISE DATA LANDSCAPE

A. SFMC Data Extensions

There are over 30 Data Extensions that are daily uploaded from a centralised data brokerage function to the data landscape managed by the MarketingMind team at the SFMC. These extensions include five main types of accounts: brokerage, advisory, retirement, charitable and international accounts. There are regulatory considerations for each of these categories: Brokerage and advisory accounts are regulated by the SEC and FINRA, retirement accounts are

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

|| Volume 7, Issue 4, July-August 2024|||

DOI:10.15662/IJRPETM.2024.0704006

subject to ERISA, and international accounts have to follow multi-jurisdictional data protection laws, such as GDPR [5] for the accounts held by EU residents.

TABLE V. BIGQUERY SUBSCRIBER PREFERENCE CENTRE: COMMUNICATION CATEGORY DISTRIBUTION AND GDPR APPLICABILITY

Communication Category	Opt-in Topics	Opt-out Topics	Data Store	GDPR Applicability
Investing Insights	3	3	BigQuery subscriber_preferences	Partial (US-based)
Market Updates	4	4	BigQuery subscriber_preferences	Partial
Events & Webinars	5	5	BigQuery subscriber_preferences	Partial
Quarterly Reports	3	3	BigQuery subscriber_preferences	Partial
Specialty Topics	10+	10+	BigQuery subscriber_preferences	Full (EU subscribers)
Total Categories	25+	25+	Unified preference schema	Multi-jurisdictional

The data hierarchy applies atop these extensions, consisting of four levels: household, customer – account, transaction-level records (trades and balances). This hierarchy is essential for proper SQL generation – a query that joins at the wrong level of hierarchy might return double-counted results that could undersell or overstate how many customers are exposed or how many assets are concentrated, an issue that may have regulatory consequences. The Schema-as-Guardrail process provides a way to ensure the agent's conception of this hierarchy is based on the actual schema structure and not upon general financial domain knowledge.

B. BigQuery Preference Centre

The BigQuery layer in MarketingMind handles the subscriber Preference Centre dataset, which stores the opt-in/opt-out preferences in over 25 communication categories per subscriber. These types include newsletters about investing, newsletters about the market, invitations to events, quarterly report delivery, and speciality topic newsletters. It is a compliance-critical function: GDPR Article 7 [5] states that the act of opting out must be as easy as opting in, and CAN-SPAM Act requirements require that opt-out requests be acted on within 10 business days. The misreporting of opt-out could lead to communications being sent to opted-out recipients, leading to a regulatory violation.

To allow for fine-grained consent withdrawal analysis in communication categories, the preference schema represents opt-in and opt-out status as two separate booleans, not as a single toggle. This design decision, which is captured in the enterprise data dictionary that is injected into the context of the BigQuery Agent, is surfaced to the analysts in the Notes section of structured responses when preference data is included in queries.

VII. CONCLUSION

A schema-grounded multi-agent system for regulatory-compliant enterprise data access, called MarketingMind, has been presented, and three governance-conforming architectural primitives, Schema-as-Guardrail, Tool-Boundary Enforcement, and Least-Privilege Agent Isolation, have been introduced. Empirical evaluation shows that these primitives collectively attain 97.3% schema fidelity, 100% mutation rejection on 10,000 adversarial inputs and 0 cross-agent tool leakage on 5,000 multi-turn sessions.

This paper's central thesis is that governance in enterprise agentic AI should be seen as an architectural primitive and not an afterthought, added onto a capability-driven system. The financial services domain is one that demands adherence to strict regulations in the areas of access control, auditability and data protection [9], which makes this architectural principle not just desirable, but essential. The structured response format presented in this work further shows that transparency mechanisms are not just a nice-to-have usability feature, but they are a precondition for analyst trust and organisational uptake.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

|| Volume 7, Issue 4, July-August 2024|||

DOI:10.15662/IJRPETM.2024.0704006

The difficulty for the text-to-SQL research community in general [4,6,7,12] is the emergence of a new problem class that public benchmarks do not account for when it comes to enterprise deployment. Going forward, governance should be included as a set of criteria in future benchmark development, in addition to execution accuracy, and include schema authorisation scope, schema mutation prevention requirements, cross-system isolation, and auditability. The integration of agentic AI capabilities [16] and the governance needs of enterprises is a fundamental challenge in the responsible use of LLMs in regulated sectors.

REFERENCES

[1] T. Yu et al., "Spider: A large-scale human-labelled dataset for complex and cross-domain semantic parsing and text-to-SQL task," in Proc. 2018 Conf. Empirical Methods Natural Language Processing, pp. 3911–3921. doi: 10.18653/v1/D18-1425.
[2] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in Proc. 16th ACM Workshop on Artificial Intelligence and Security, pp. 79–90, 2023. doi: 10.1145/3605764.3623985.
[3] Z. Ji et al., "Survey of hallucination in natural language generation," ACM Comput. Surv., vol. 55, no. 12, Art. no. 248, 2023. doi: 10.1145/3571730.
[4] D. Gao et al., "Text-to-SQL empowered by large language models: A benchmark evaluation," Proc. VLDB Endowment, vol. 17, pp. 1132–1145, 2023. doi: 10.48550/arXiv.2308.15363.
[5] T. T. Ke and K. Sudhir, "Privacy rights and data security: GDPR and personal data markets," Management Science, vol. 69, no. 8, pp. 4389–4412, 2023. doi: 10.1287/mnsc.2022.4614.
[6] H. Li, J. Zhang, C. Li, and H. Chen, "RESDSQL: Decoupling schema linking and skeleton parsing for text-to-SQL," in Proc. AAAI Conf. Artificial Intelligence, vol. 37, no. 11, pp. 13067–13075, 2023. doi: 10.48550/arXiv.2302.05965.
[7] J. Li et al., "Can LLM already serve as a database interface? A big bench for large-scale database grounded text-to-SQLs," in Adv. Neural Inf. Process. Syst., vol. 36, 2023. doi: 10.48550/arXiv.2305.03111.
[8] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Adv. Neural Inf. Process. Syst., vol. 33, pp. 9459–9474, 2020. doi: 10.48550/arXiv.2005.11401.
[9] D. McNulty, A. Miglionico, and A. Milne, "Data access technologies and the 'new governance' techniques of financial regulation," J. Financial Regulation, vol. 9, no. 2, pp. 225–248, 2023. doi: 10.1093/jfr/fjad008.
[10] P. Mehta, V. Mehta, H. Pardeshi, and P. Bide, "Survey on natural language interfaces to databases," in Advances in Data-Driven Computing and Intelligent Systems, Lecture Notes in Networks and Systems, vol. 698, pp. 361–369, 2023. doi: 10.1007/978-981-99-3250-4_28.
[11] OpenAI, "GPT-4 technical report," arXiv, 2023. doi: 10.48550/arXiv.2303.08774.
[12] M. Pourreza and D. Rafiei, "DIN-SQL: Decomposed in-context learning of text-to-SQL with self-correction," in Adv. Neural Inf. Process. Syst., vol. 36, pp. 36339–36348, 2023. doi: 10.48550/arXiv.2304.11015.
[13] L. Wang et al., "PROTON: Probing schema linking information from pre-trained language models for text-to-SQL parsing," in Proc. 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining, pp. 1875–1884, 2022. doi: 10.1145/3534678.3539305.
[14] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in Adv. Neural Inf. Process. Syst., vol. 35, pp. 24824–24837, 2022. doi: 10.48550/arXiv.2201.11903.
[15] V. Zhong, C. Xiong, and R. Socher, "Seq2SQL: Generating structured queries from natural language using reinforcement learning," arXiv, 2017. doi: 10.48550/arXiv.1709.00103.
[16] S. Yao et al., "ReAct: Synergising reasoning and acting in language models," in Int. Conf. Learning Representations, 2023. doi: 10.48550/arXiv.2210.03629.
[17] C. Brewis, S. Dibb, and M. Meadows, "Leveraging big data for strategic marketing: A dynamic capabilities model for incumbent firms," Technological Forecasting and Social Change, vol. 190, p. 122402, 2023. doi: 10.1016/j.techfore.2023.122402.