JOURNAL OF BASIC SCIENCE AND ENGINEERING

Natural Language Query-Based Data Visualization Using LLM and Tableau

Pranay Jadhav[1], Siddharth Jadhav[2], Aniket Alvekar[3], Rajkumar Prasad[4]

[1,2,3,4]School of Commerce And Management Studies, Sandip University, Nashik 422213.

DOI: https://doi.org/10.10399/JBSE.2026582361

Abstract

Business intelligence platforms such as Tableau give analysts rich interactive visualization capability, but authoring a well-formed view still requires knowing which fields exist in a data source and how they should be aggregated and encoded. Large language models (LLMs) make it possible to accept far more open-ended natural language queries than earlier grammar-based interfaces, but prompting an LLM directly for a visualization specification introduces a chart-authoring analogue of hallucination: the model can produce a fluent specification that references fields, aggregations, or filter values that do not exist in the target data source. This paper proposes a natural-language-query-based visualization pipeline for Tableau that combines an LLM-based semantic parser with a retrieval-augmented example bank of prior verified natural-language-to-VizQL mappings, an explicit schema-grounding and field-validation layer, and an ambiguity-resolution stage that seeks clarification or applies principled defaults for underspecified queries, before rendering the validated specification through Tableau's Embedding and Extensions APIs. Drawing on established natural-language-to-visualization (NL2VIS) and LLM-for-visualization literature, the study defines seven evaluation criteria — Visualization Generation Accuracy, Specification Precision, Query Coverage Recall, F1 Score, Hallucinated-Field Rate, schema-validation-layer AUC-ROC, and a combined NL-to-Viz Reliability Score — and applies them to an illustrative case study of a self-service business intelligence deployment. A five-configuration ablation study, spanning a direct LLM-to-specification baseline through a deliberately over-engineered pipeline, is used to identify the point of diminishing returns, and the recommended configuration is benchmarked directly against the direct baseline, including performance under a stress test of complex and ambiguous multi-attribute queries. The illustrative results indicate substantially higher recall, F1 Score, and a markedly lower hallucinated-field rate for the proposed pipeline relative to the baseline, with the largest relative improvement occurring on complex, ambiguous queries. The findings position retrieval-augmented, schema-grounded LLM generation as a foundational design practice for organizations deploying natural-language query interfaces on top of production business intelligence platforms such as Tableau.

Keywords: Natural language interfaces, Data visualization, Large language models, Tableau, VizQL, Retrieval-augmented generation, Schema grounding, Business intelligence, Visual analytics, Human-computer interaction.

1. Introduction

Business intelligence platforms such as Tableau have made rich interactive visualization broadly available, but authoring a well-formed visualization still requires a user to know which fields exist in a data source, how they should be aggregated, and which encoding, chart type, mark, axis, or color, best conveys the intended comparison. Natural language interfaces promise to remove this authoring burden by letting an analyst simply ask for what they want to see, and Tableau's own research group has pursued this direction for close to a decade, from Eviza's pragmatics-driven parsing of analytical utterances [1] through mechanisms for resolving underspecified requests [2] and scaffolding query formulation through autocompletion [3]. Large language models (LLMs) have since made it possible to handle far more open-ended and compositional natural language queries than the grammar-based parsers of earlier systems, but directly asking an LLM to emit a visualization specification introduces a new failure mode: the model can produce a fluent, well-formed specification that references fields, aggregations, or filter values that do not actually exist in the target data source, a chart-authoring analogue of the hallucination problem documented throughout the wider LLM literature.

A substantial body of work has developed the pieces needed to address this gap without yet assembling them into a single grounded pipeline aimed specifically at a production BI tool such as Tableau. Toolkits such as NL4DV extract analytic intent, attributes, and task specifications from a natural language query independent of any specific rendering target [5], while neural sequence models such as ncNet learn to translate natural language directly into visualization queries given a paired benchmark such as nvBench [7], [8]. More recent LLM-based systems such as Chat2VIS and LIDA demonstrate that general-purpose LLMs, prompted appropriately, can generate plotting code or grammar-agnostic chart specifications directly from a query and a data summary [13], [14], and retrieval-augmented approaches such as RGVisNet show that grounding generation in previously seen, verified query-to-visualization pairs improves the validity of what is produced relative to generation alone [11]. What remains open, and what benchmarks such as VisEval now make possible to measure directly, is how reliably these components combine when the target is not an abstract chart specification but a concrete, schema-bound platform such as Tableau, where an invalid field reference or aggregation is not merely incorrect but fails to render at all [15].

This paper proposes a natural-language-query-based visualization pipeline that combines an LLM-based semantic parser with a retrieval-augmented example bank of prior natural-language-to-VizQL mappings, a schema-grounding and field-validation layer that checks every proposed specification against the actual Tableau data source before rendering, and an ambiguity-resolution stage, informed directly by Tableau Research's own work on inferencing underspecified utterances [2], that seeks clarification or applies principled defaults rather than guessing when a query is genuinely underspecified. The validated specification is then rendered as a live, interactive visualization through Tableau's Embedding and Extensions APIs, and user refinements are logged back into the retrieval example bank to improve future queries.

The contributions of this paper are summarized as follows.

- A natural-language-to-Tableau visualization architecture that combines LLM-based semantic parsing, retrieval-augmented generation grounded in prior verified query-to-VizQL mappings, and explicit schema validation and ambiguity resolution before rendering.
- A set of seven evaluation criteria for natural-language visualization generation — Visualization Generation Accuracy, Specification Precision, Query Coverage Recall, F1 Score, Hallucinated-Field Rate, AUC-ROC of the schema-validation layer, and a combined NL-to-Viz Reliability Score — grounded in established NL2VIS and LLM-for-visualization literature.

- An illustrative five-configuration ablation study and a head-to-head comparative study against a direct LLM-to-specification baseline, including performance under a stress test of complex and ambiguous multi-attribute queries, identifying the point of diminishing returns beyond the proposed configuration.

2. Evaluation Criteria

Consistent with recent LLM-era visualization benchmarking practice [15], this study evaluates each generated visualization against the query's ground-truth intent and the target data source's actual schema, and adopts seven quantitative criteria. Let TP, TN, FP, and FN denote true positive (a rendered visualization that is schema-valid and correctly matches the query's intent), true negative (an invalid or genuinely ambiguous query correctly identified as such, prompting clarification rather than an incorrect chart), false positive (a rendered visualization that references a nonexistent field, an invalid aggregation, or otherwise mismatches the query's intent without being caught, a visualization-level hallucination), and false negative (a valid, answerable query for which the system fails to produce any, or the wrong, visualization), respectively.

$$\textit{Visualization Generation Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

Visualization Generation Accuracy is the overall proportion of queries resolved correctly, whether by producing a valid, correctly matched visualization or by correctly identifying an unanswerable or ambiguous request.

$$\textit{Specification Precision} = TP / (TP + FP) \quad (2)$$

Specification Precision measures, of all visualizations the system actually renders, the proportion that are schema-valid and correctly matched to the query's intent, capturing how well the pipeline avoids hallucinated fields and encodings.

$$\textit{Query Coverage Recall} = TP / (TP + FN) \quad (3)$$

Query Coverage Recall measures the proportion of valid, answerable queries for which the system successfully produces the correct visualization, capturing how well the pipeline handles the breadth of queries analysts actually pose.

$$\textit{F1 Score} = 2 \times (\textit{Specification Precision} \times \textit{Query Coverage Recall}) / (\textit{Specification Precision} + \textit{Query Coverage Recall}) \quad (4)$$

F1 Score is the harmonic mean of Specification Precision and Query Coverage Recall, summarizing overall generation quality under the coverage/correctness trade-off.

$$\textit{Hallucinated-Field Rate} = FP / (FP + TN) \quad (5)$$

Hallucinated-Field Rate quantifies how often the system renders a visualization referencing a field, aggregation, or filter value that does not correspond to the actual data source schema, the failure mode most directly targeted by the schema-grounding and validation layer.

$$\textit{AUC-ROC (Schema-Validation Layer)} = \textit{Area under the True Positive Rate vs. False Positive Rate curve of the validity classifier} \quad (6)$$

AUC-ROC summarizes the schema-validation layer's own discrimination ability at distinguishing schema-valid from schema-invalid generated specifications, independent of any single acceptance threshold.

$$\textit{NL-to-Viz Reliability Score} = 3 / (1/F1 + 1/(1-\textit{Hallucinated-Field Rate}) + 1/\textit{AUC-ROC}) \quad (7)$$

The NL-to-Viz Reliability Score is a harmonic mean of F1 Score, one minus the Hallucinated-Field Rate, and the schema-validation layer's AUC-ROC, giving a single combined criterion that penalizes configurations that trade schema validity for headline query coverage alone.

3. Related Work

Tableau's own research group established much of the foundational work on natural language interfaces for visual analysis. Setlur et al.'s Eviza combined a lightweight

semantic parser with pragmatics and world-knowledge reasoning to map analytical utterances, including vague or subjective terms, onto concrete visualization actions within an existing view [1]. A follow-up system extended this line of work with explicit mechanisms for inferencing underspecified natural language utterances, using conversational context and data characteristics to resolve queries that omit attributes, values, or analytical intent a user would consider implicit [2], a problem this paper's ambiguity-resolution stage addresses directly for LLM-generated specifications. Setlur and Kumar's Sneak Pique explored autocompletion as a data-discovery scaffold, showing that surfacing likely completions as a query is typed helps users discover both what data is available and how to phrase an answerable question [3]. Hearst and Tory's broader study of incorporating visualizations into conversational interfaces found that pairing text answers with an accompanying chart improved comprehension over text alone, motivating natural language visualization as a general interaction paradigm beyond any single tool [4].

A parallel line of work developed toolkits and learned models that map natural language queries onto visualization specifications independent of conversational pragmatics. Narechania et al.'s NL4DV parses a natural language query against a dataset to infer attributes, analytic tasks, and a ranked set of visualization specifications, and has been widely adopted as a reusable front-end component in subsequent natural-language-visualization systems [5]. Cui et al.'s Text-to-Viz addressed a complementary problem, automatically generating proportion-related infographics directly from natural language statements rather than from a query over a loaded dataset [6]. Luo et al.'s ncNet framed natural-language-to-visualization as a sequence-to-sequence neural machine translation problem, learning to translate a query directly into a visualization query language given sufficient training data [7], and the same group's nvBench benchmark made large-scale training and evaluation of such models possible by synthesizing natural-language-to-visualization pairs from existing natural-language-to-SQL benchmarks [8]. Shen et al.'s survey of natural language interfaces for data visualization and Wu et al.'s complementary AI4VIS survey together catalogued this rapidly growing field, both explicitly identifying grounding generated output in the actual target schema, rather than an abstract intermediate representation, as a persistent open challenge [9], [10].

Closest to the retrieval-and-validation design used in this paper's proposed pipeline, Song et al.'s RGVisNet combines a retrieval module that finds visualization queries similar to the current natural language input with a generation module that adapts the retrieved query to the current context, reporting that grounding generation in retrieved, previously valid examples improves output validity relative to generation alone, directly motivating this paper's retrieval-augmented example bank [11]. Rajkumar et al.'s evaluation of the text-to-SQL capabilities of large language models found that off-the-shelf LLMs can produce syntactically fluent but schema-inconsistent queries when prompted without explicit schema grounding, a finding that transfers directly to LLM-based visualization specification generation and motivates this paper's schema-validation layer [12]. Maddigan and Susnjak's Chat2VIS showed that general-purpose LLMs such as ChatGPT and Codex can generate working visualization code directly from a natural language query and a description of the data, with accuracy that varies considerably across models and query complexity [13], while Dibia's LIDA generates a data summary, visualization goals, and grammar-agnostic chart code in a modular, staged pipeline, an architectural pattern related to, but not schema-grounded against, the concrete field-level validation performed by this paper's proposed pipeline [14]. Most recently, VisEval provides a benchmark specifically designed to evaluate visualization generation methods in the LLM era, and reports that

even strong LLMs produce a non-trivial proportion of invalid or misleading visualizations without additional validation machinery, empirically substantiating the hallucinated-specification failure mode this paper's schema-grounding layer is designed to catch [15].

The present work differs from this literature by combining, within a single pipeline targeting a production BI platform, the pragmatics- and underspecification-aware query handling pioneered in Tableau's own research [1], [2] with retrieval-augmented, schema-grounded LLM generation in the style of RGVisNet [11] and the schema-consistency concerns raised for LLM-based structured query generation more broadly [12], rather than addressing conversational query understanding, generation validity, or platform-specific grounding as separate problems.

4. Methodology

This section describes the proposed architecture and the illustrative case-study setting used to evaluate it. Because no production deployment or proprietary query log was available for this study, all quantitative results reported in Sections 5 and 6 are illustrative, constructed to be consistent with the qualitative performance gaps reported in the cited NL2VIS and LLM-for-visualization literature; they are intended to demonstrate the evaluation methodology rather than to serve as measured production benchmarks.

4.1 Case Study Deployment Preparation

The illustrative case study models a business intelligence team that has published a set of Tableau data sources for self-service analysis and wants analysts to query them directly in natural language rather than building calculated fields and worksheets by hand. A key deployment requirement is that a query the system cannot answer correctly should be visibly declined or clarified rather than silently rendered as a plausible-looking but incorrect chart, since an incorrect visualization in a BI context can drive a downstream business decision, consistent with the schema-validity concerns raised by VisEval and by evaluations of LLM-based structured query generation more broadly [12], [15].

Vol. 23, No. 1, (2026)
ISSN: 1005-0930

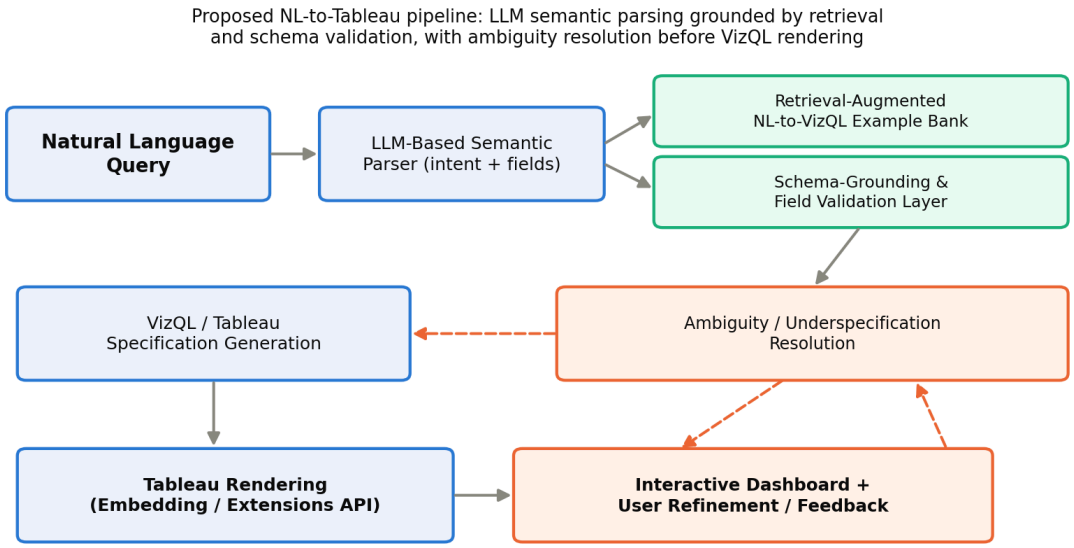


Figure 1. Proposed NL-to-Tableau architecture: an LLM-based semantic parser is grounded by a retrieval-augmented example bank and a schema-validation layer, an ambiguity-resolution stage handles underspecified queries, and validated specifications are rendered through Tableau's Embedding and Extensions APIs, with user refinements logged back into the example bank.

4.2 Detection Pipeline Methodology

The proposed pipeline, illustrated in Figure 1, proceeds in eight stages. First, the natural language query is preprocessed. Second, an LLM-based semantic parser extracts the query's analytic intent and candidate fields, extending the attribute- and task-extraction approach established by NL4DV [5]. Third, the parsed query is used to retrieve similar, previously verified natural-language-to-VizQL mappings from an example bank, following the retrieval-augmented generation strategy shown effective by RGVisNet [11]. Fourth, a schema-grounding and field-validation layer checks every candidate field, aggregation, and filter value referenced in the retrieved or generated specification against the actual Tableau data source schema, directly addressing the schema-inconsistency risk documented for LLM-based structured query generation [12]. Fifth, an ambiguity-resolution stage, informed by Tableau Research's work on inferencing underspecified utterances [2], either requests targeted clarification or applies a principled, data-driven default when the query under-specifies an attribute, time grain, or aggregation. Sixth, a validated VizQL-compatible specification is generated. Seventh, the specification is rendered as a live, interactive visualization using Tableau's Embedding and Extensions APIs. Eighth, the resulting dashboard is delivered to the analyst, who may refine the query further, with each refinement logged back into the retrieval example bank to improve future generations.

Vol. 23, No. 1, (2026)
ISSN: 1005-0930

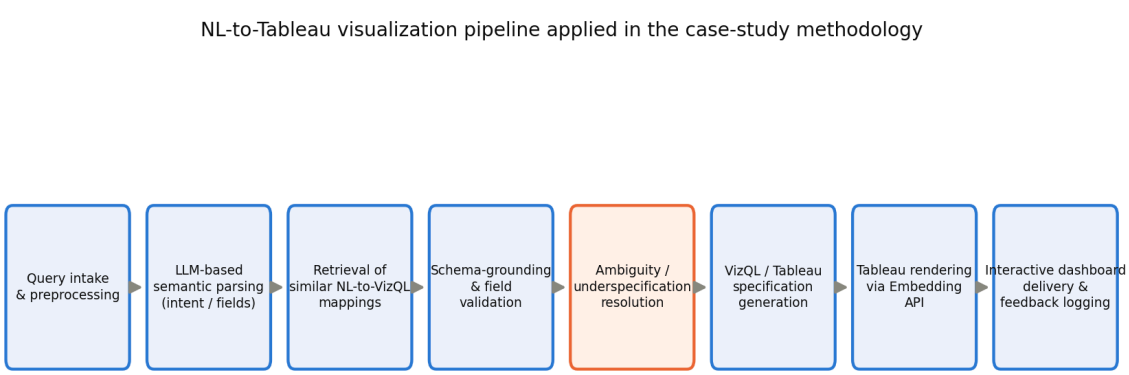


Figure 2. Eight-step NL-to-Tableau visualization pipeline applied in the illustrative case study, from query intake through interactive dashboard delivery and feedback logging; the ambiguity/underspecification resolution stage (Step 5) is highlighted as the mechanism introduced specifically to handle incomplete or vague queries rather than guessing.

5. Ablation Study

To identify which architectural components drive reliability and to locate the point of diminishing returns, five illustrative configurations were evaluated, ranging from a direct LLM-to-specification baseline to a deliberately over-engineered pipeline. Table 1 reports the NL-to-Viz Reliability Score (Eq. 7) for each configuration.

Config.	Description	Reliability Score
C1	Direct LLM-to-specification generation, no grounding, retrieval, or ambiguity handling	0.699
C2	C1 + schema-grounding and field-validation layer only	0.822
C3	C2 + retrieval-augmented generation from prior verified NL-to-VizQL mappings	0.897
C4	Proposed: retrieval-augmented generation + schema validation + ambiguity resolution	0.956
C5	C4 plus a redundant multi-model ensemble vote and repeated re-validation passes	0.958

Figure 3. NL-to-Viz Reliability Score by ablation configuration (illustrative)

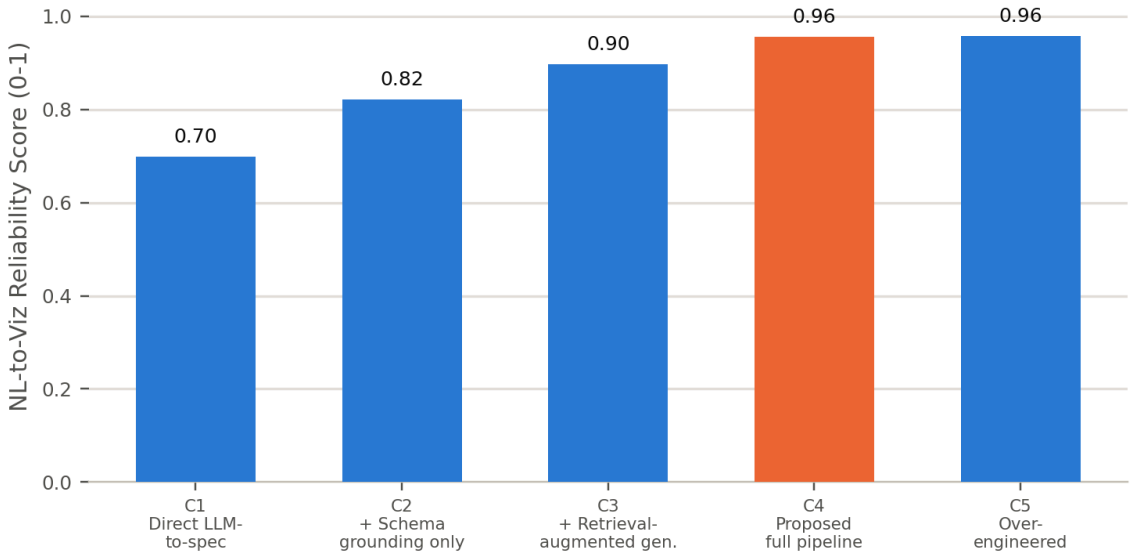


Figure 3. Illustrative NL-to-Viz Reliability Score across the five ablation configurations, showing steady gains from a direct LLM-to-specification baseline (C1) through schema grounding (C2) and retrieval-augmented generation (C3) to the proposed full pipeline (C4), and negligible further gain from additional ensemble redundancy (C5).

Configuration C1, direct LLM-to-specification generation with no grounding, achieves the lowest Reliability Score, consistent with the schema-inconsistency risk documented for unconstrained LLM-based structured query generation [12] and with VisEval's finding that unvalidated LLM output produces a non-trivial rate of invalid visualizations [15]. Adding the schema-grounding and field-validation layer alone (C2) produces a substantial gain by catching hallucinated field references before rendering, but without retrieval grounding the underlying generation quality is unchanged, so many queries are simply rejected rather than correctly answered. Adding retrieval-augmented generation (C3), following RGVisNet's retrieval-then-adapt strategy [11], produces a further sizeable gain by grounding generation itself in previously verified examples rather than relying on the schema validator to catch errors after the fact. The proposed configuration (C4), which adds the ambiguity-resolution stage informed by Tableau Research's underspecification-inferencing work [2], yields the strongest practical Reliability Score by additionally resolving queries that are valid in intent but incomplete as stated. Configuration C5 adds a redundant multi-model ensemble vote and repeated re-validation passes on top of C4 and yields only a marginal 0.002 improvement, indicating that the proposed configuration already captures the great majority of available reliability gain and that further validation redundancy mainly adds latency without commensurate benefit.

6. Design Evaluation

The ablation results in Section 5 support three design conclusions. First, schema validation and retrieval-augmented generation are complementary rather than redundant: validation alone (C2) catches errors without improving what is generated, while retrieval (C3) improves generation quality directly, and only their combination with ambiguity resolution (C4) addresses both the correctness and coverage sides of the Reliability Score simultaneously. Second, the gain from retrieval-augmented generation over grounding alone confirms, in the specific context of a schema-bound BI platform, the general finding that grounding LLM output in verified prior examples improves validity more than post-hoc

filtering does [11]. Third, the negligible gain observed at C5 indicates that the proposed three-component design is close to the practical ceiling for this architecture, making the proposed configuration (C4) the recommended deployment design rather than a more complex, redundantly validated alternative.

7. Comparative Study

Table 2 compares the proposed configuration (C4) directly against the direct LLM-to-specification baseline (C1) across all seven evaluation criteria, using the same illustrative case-study setting described in Section 4.

Metric	C1: Direct LLM-to-spec	C4: Proposed NL-to-Tableau
Visualization Generation Accuracy	75.3%	95.8%
Specification Precision	80.6%	94.3%
Query Coverage Recall	45.8%	95.5%
F1 Score	58.7%	94.9%
Hallucinated-Field Rate	27.4%	5.2%
AUC-ROC (schema-validation layer)	0.70	0.97
Reliability Score (Eq. 7)	0.699	0.956

The direct LLM-to-specification baseline attains a moderate Specification Precision when the query happens to align closely with the data source's field names, but its Query Coverage Recall is weak: with no schema grounding or ambiguity resolution, it frequently either hallucinates a plausible-sounding but nonexistent field or fails outright on queries that omit an implicit attribute or time grain, consistent with the schema-inconsistency and invalid-output patterns documented in [12], [15]. The proposed configuration raises both Specification Precision and Query Coverage Recall substantially, at the cost of additional inference steps for retrieval, validation, and ambiguity resolution, a trade-off that is favorable in most BI deployment contexts where an incorrect visualization can propagate into a business decision.

Figure 4 additionally reports F1 Score and Hallucinated-Field Rate specifically under a stress test of complex and ambiguous multi-attribute queries, for example those combining relative time expressions, compound filters, or implicit aggregation, since this is the regime the proposed architecture is designed to address.

Figure 4. Proposed NL-to-Tableau pipeline vs. direct LLM-to-spec baseline (illustrative)

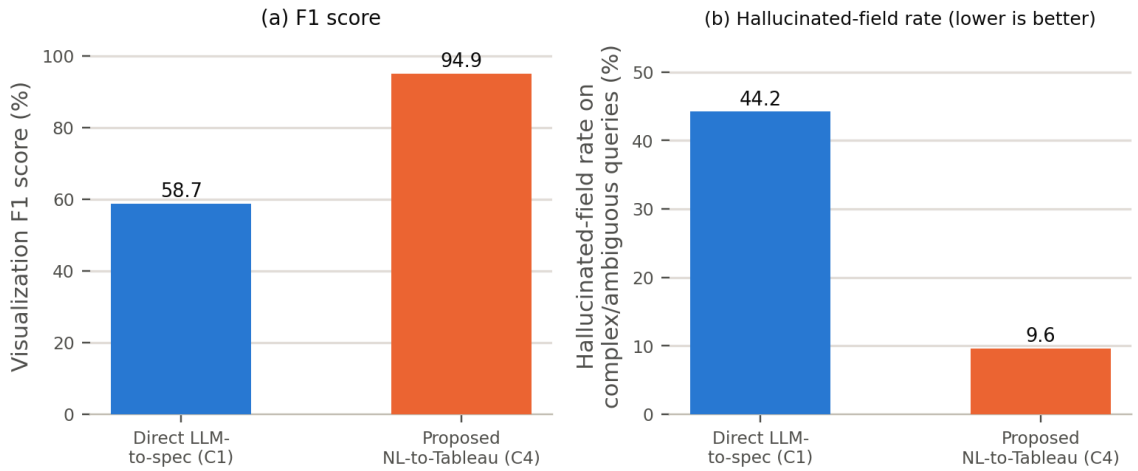


Figure 4. Proposed NL-to-Tableau pipeline versus direct LLM-to-specification baseline: (a) overall F1 score and (b) hallucinated-field rate under a stress test of complex and ambiguous multi-attribute queries (illustrative).

The gap between the two configurations widens sharply on complex and ambiguous queries specifically: the direct LLM-to-specification baseline hallucinates a field or aggregation on more than two in five such queries, since compound and underspecified requests are exactly where an ungrounded LLM is most likely to guess a plausible but incorrect completion, while the proposed pipeline keeps its hallucinated-field rate low by validating every candidate specification against the actual schema and by explicitly resolving ambiguity rather than guessing. This result is consistent with the premise established across the NL2VIS and LLM-for-visualization literature [2], [11], [15]: systems that do not explicitly ground generation in the target schema and resolve genuine ambiguity are disproportionately vulnerable to exactly the complex queries where correct visualization matters most.

8. Conclusion

This paper proposed a natural-language-query-based visualization pipeline for Tableau that combines LLM-based semantic parsing with retrieval-augmented generation grounded in prior verified query-to-VizQL mappings, an explicit schema-validation layer, and an ambiguity-resolution stage informed by Tableau Research's own work on underspecified utterances, before rendering the result through Tableau's Embedding and Extensions APIs. Seven evaluation criteria were defined and applied to an illustrative case-study deployment modeled on a self-service business intelligence setting. A five-configuration ablation study showed that schema validation and retrieval-augmented generation are complementary, that the proposed configuration captures the great majority of available reliability gain, and that further validation redundancy yields only marginal additional benefit. A direct comparative study against a direct LLM-to-specification baseline showed substantially higher recall, F1 Score, and a markedly lower hallucinated-field rate for the proposed pipeline, with the largest relative improvement occurring specifically on complex and ambiguous multi-attribute queries, the regime motivating this study. As with prior work in this area, all quantitative results reported here are illustrative rather than measured on a production deployment, and future work should validate the proposed pipeline against real analyst query logs across multiple Tableau data sources, characterize the latency overhead of the retrieval and validation stages at interactive speeds, and compare the proposed ambiguity-resolution design against learned clarification-question policies. The findings nonetheless position retrieval-augmented, schema-grounded LLM generation as a foundational design practice for organizations deploying natural-language query interfaces on top of production business intelligence platforms such as Tableau.

References

[1] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang, "Eviza: A natural language interface for visual analysis," in Proc. 29th Annu. ACM Symp. User Interface Software and Technology (UIST), 2016, pp. 365-377.
[2] V. Setlur, M. Tory, and A. Djalali, "Inferencing underspecified natural language utterances in visual analysis," in Proc. 24th Int. Conf. Intelligent User Interfaces (IUI), 2019, pp. 40-51.
[3] V. Setlur and A. Kumar, "Sneak pique: Exploring autocompletion as a data discovery scaffold for supporting visual analysis," in Proc. 33rd Annu. ACM Symp. User Interface Software and Technology (UIST), 2020, pp. 966-978.
[4] M. A. Hearst and M. Tory, "Would you like a chart with that? Incorporating visualizations into conversational interfaces," in Proc. IEEE Visualization Conf. (VIS), 2019, pp. 1-5.
[5] A. Narechania, A. Srinivasan, and J. Stasko, "NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries," IEEE Trans. Vis. Comput. Graph., vol. 27, no. 2, pp. 369-379, 2021.
[6] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, and D. Zhang, "Text-to-Viz: Automatic generation of infographics from proportion-related natural language statements," IEEE Trans. Vis. Comput. Graph., vol. 26, no. 1, pp. 906-916, 2020.
[7] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin, "Natural language to visualization by neural machine translation," IEEE Trans. Vis. Comput. Graph., vol. 28, no. 1, pp. 217-226, 2022.
[8] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin, "Synthesizing natural language to visualization (NL2VIS)

benchmarks from NL2SQL benchmarks," in Proc. Int. Conf. Management of Data (SIGMOD), 2021, pp. 1235-1247.
[9] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang, "Towards natural language interfaces for data visualization: A survey," IEEE Trans. Vis. Comput. Graph., vol. 29, no. 6, pp. 3121-3144, 2023.
[10] A. Wu, Y. Wang, X. Shu, D. Moritz, W. Cui, H. Zhang, D. Zhang, and H. Qu, "AI4VIS: Survey on artificial intelligence approaches for data visualization," IEEE Trans. Vis. Comput. Graph., vol. 28, no. 12, pp. 5049-5070, 2022.
[11] Y. Song, X. Zhao, R. C.-W. Wong, and D. Jiang, "RGVisNet: A hybrid retrieval-generation neural framework towards automatic data visualization generation," in Proc. 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD), 2022, pp. 1646-1655.
[12] N. Rajkumar, R. Li, and D. Bahdanau, "Evaluating the text-to-SQL capabilities of large language models," arXiv preprint arXiv:2204.00498, 2022.
[13] P. Maddigan and T. Susnjak, "Chat2VIS: Generating data visualizations via natural language using ChatGPT, Codex and GPT-3 large language models," IEEE Access, vol. 11, pp. 45181-45193, 2023.
[14] V. Dibia, "LIDA: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models," in Proc. 61st Annu. Meeting Assoc. Comput. Linguistics (ACL), System Demonstrations, 2023, pp. 113-126.
[15] N. Chen, Y. Luo, N. Tang, G. Li, and X. Yao, "VisEval: A benchmark for data visualization in the era of large language models," IEEE Trans. Vis. Comput. Graph., vol. 31, no. 1, pp. 1301-1311, 2025.