Lexara: A User-Centered Toolkit for Evaluating Large Language Models for Conversational Visual Analytics

Srishti Palani
srishti.palani@salesforce.com
Tableau Research, Salesforce
Palo Alto, CA, USA

Vidya Setlur
vsetlur@salesforce.com
Tableau Research, Salesforce
Palo Alto, CA, USA

Abstract

Large Language Models (LLMs) are transforming Conversational Visual Analytics (CVA) by enabling data analysis through natural language. However, evaluating LLMs for CVA remains a challenge: requiring programming expertise, overlooking real-world complexity, and lacking interpretable metrics for multi-format (visualizations and text) outputs. Through interviews with 22 CVA developers and 16 end-users, we identified use cases, evaluation criteria and workflows. We present Lexara, a user-centered evaluation toolkit for CVA that operationalizes these insights into: (i) test cases spanning real-world scenarios; (ii) interpretable metrics covering visualization quality (data fidelity, semantic alignment, functional correctness, design clarity) and language quality (factual grounding, analytical reasoning, conversational coherence) using rule-based and LLM-as-a-Judge methods; and (iii) an interactive toolkit enabling experimental setup and multi-format and multi-level exploration of results without programming expertise. We conducted a two-week diary study with six CVA developers, drawn from our initial cohort of 22. Their feedback demonstrated Lexara's effectiveness for guiding appropriate model and prompt selection.

CCS Concepts

• **Human-centered computing** → **Visualization systems and tools**; **Interactive systems and tools**.

Keywords

Benchmarking, Analytical Conversation, Visual Analytics, Large Language Model Evaluation

ACM Reference Format:
Srishti Palani and Vidya Setlur. 2026. Lexara: A User-Centered Toolkit for Evaluating Large Language Models for Conversational Visual Analytics. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 25 pages. https://doi.org/10.1145/3772318.3790735

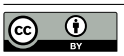
1 Introduction

Recent advances in Large Language Models (LLMs) have enabled a shift toward more natural, conversational interactions with data [52, 56, 69]. Increasingly, LLMs are being integrated into *Conversational Visual Analytics (CVA)* tools, allowing users to generate and refine visualizations through natural language [22, 50, 53, 73, 88]. This democratizes *Visual Analytics (VA)*, traditionally defined as analytical reasoning facilitated by interactive visual interfaces [13, 84], by making it accessible to users without programming or analytical expertise. As CVA tools proliferate, it has become imperative for CVA tool developers and end-user analysts to continually evaluate and adapt to a rapidly growing ecosystem of LLMs and system prompts. These choices directly affect system behavior, output quality, and end-user trust [10]. To understand current evaluation practices and identify gaps in existing approaches, we conducted formative studies with practitioners to investigate the following research questions:

> **RQ1:** What does **practitioners' real-world use of CVA** look like?
> **RQ2:** What **evaluation criteria** do practitioners apply when assessing CVA system outputs?
> **RQ3:** What **evaluation workflows** do practitioners use for CVA interactions, what challenges do they face, and how well do existing tools address these challenges?

Through semi-structured interviews with 22 CVA tool developers and an observational study with 16 end-users (where a browser extension logged real-world CVA interactions), we uncovered significant gaps between practitioner needs and existing approaches. Thematic analysis revealed that real-world CVA usage is inherently multi-turn and multi-format: users engage in iterative conversations where context from earlier exchanges informs later responses, and expect systems to produce integrated text, visualization, and code outputs. Practitioners evaluate both *visualization quality* (e.g., data fidelity, field similarity, chart type, axes, filters and sorting, visual encodings, and interactivity) and *analytical natural language response quality* (e.g., factual grounding, analytical thinking, conversational coherence, and follow-up relevance across turns), emphasizing the need for flexible, multi-granular evaluation that accommodates graded correctness and multiple valid answers. A response may be technically correct but still suboptimal or misleading. For example, a visualization response with swapped axes, choice of `pie` vs. `bar` charts, or semantically inferred fields (e.g., `Profit` vs. `Revenue-Cost`) may be valid, but still differ from expected outputs in ways that could affect interpretability and trust in the analyses. Consequently, practitioners rely on ad-hoc, fragmented CVA evaluation workflows: manually comparing outputs across spreadsheets, adapting ill-suited Natural Language Processing (NLP) metrics, and referencing external benchmark reports.

Yet current evaluation approaches fall short of these requirements. Existing CVA benchmarks' test cases [17, 47, 48] are synthetically generated, focus primarily on single-turn interactions, and require programming expertise for setup and interpretation, limiting accessibility for product managers, designers, and other



CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/2026/04
https://doi.org/10.1145/3772318.3790735

low-code stakeholders. Traditional NLP metrics like BLEU [59], ROUGE [44], or F1, Precision, Recall [20] are limited to n-gram overlap with single references and struggle with multi-format CVA outputs. Even recent visualization-specific metrics [22, 45, 58, 64] focus on isolated aspects rather than the entire CVA pipeline, and rarely accommodate graded correctness and are difficult to interpret [97]. Recent general-purpose LLM evaluation tools [4, 34, 38] offer low-code interfaces but lack native support for CVA-specific aspects like rendered visualizations, visualization grammars, analytical thinking in natural language responses, and data source analysis. These gaps leave practitioners unable to systematically evaluate the LLM-mediated interaction components (e.g., conversational coherence, inferred assumptions, field selection, and visualization correctness) that critically shape whether downstream analytical reasoning is even possible. While visual analytics (VA) research traditionally focuses on assessing end-to-end sensemaking effectiveness, cognitive support, and task-level analytical outcomes [35, 62, 63], evaluating these LLM-mediated layers requires different approaches that complement rather than replace traditional user-centered VA evaluation.

To address these gaps, we present Lexara, a user-centered toolkit for evaluating LLMs for CVA that operationalizes our formative findings into:

- **CVA test cases** derived from logged real-world end-user analyst interactions (RQ1),
- **Interpretable, graded CVA evaluation metrics** aligned with practitioners' evaluation criteria (RQ2), and
- **An interactive low-code CVA-specific benchmarking tool** designed around practitioners' workflow challenges and needs (RQ3), enabling multi-format (text-to-spec visualizations, text-based analytical explanations), multi-turn evaluation and systematic comparison of model-prompt configurations.

We deployed Lexara and engaged six of the 22 CVA tool developers in a two-week diary study. Feebdack demonstrated that Lexara's test cases capture real-world complexity, offer more interpretable metrics than traditional approaches, and enables practitioners to uncover performance patterns, diagnose model and prompt behavior, and make informed deployment decisions. The toolkit is publicly available at https://lexara-6b38293fcdac.herokuapp.com/ with open-source code at https://anonymous.4open.science/r/Lexara-CVA-Eval-280B/README.md. Overall, this work makes progress toward the larger vision of responsible AI development for analytics [29, 91], enabling practitioners to systematically evaluate, compare, and improve LLM-based systems before deployment.

2 Related Work

This paper builds on prior research across three themes: (1) CVA tools, which examine how users generate visualizations via natural language dialogue; (2) CVA evaluation tools, which offer frameworks and interfaces to systematically assess conversational outputs for VA; and (3) visualization and analytical language evaluation methods, which propose both quantitative and qualitative metrics to judge the quality of generated visualizations and analytical explanations.

2.1 CVA Tools

A growing body of work has explored *Conversational Visual Analytics (CVA) tools, i.e., systems that enable users to interact with data and create visualizations through natural language dialogue* [71, 89]. These tools are designed to lower the technical barriers to data exploration by allowing users to issue queries in natural language, which the tool interprets to: retrieve relevant data fields, select appropriate chart types, assign encodings, and generate visualizations.

Early CVA tools used keyword recognition and clarifications [69, 76, 93], interactive widgets [24, 81], and gesture-based input [77] during the conversation so that users can intuitively interact with their data without technical or programming expertise. While these interaction mechanisms help, intent inference, deeper analysis and conversational coherence across turns, ambiguity resolution, etc. remained open challenges [85].

With recent advances in LLMs, there has been a marked shift toward more expressive and capable CVA tools that can comprehend colloquial, flexible queries and generate diverse output formats, including structured code, visualization specifications, rendered charts, and natural language explanations. Tools like Chat2VIS [50] leverage GPT-3.5 to translate user queries into code for visualizations, supporting iterative refinement through multi-turn dialogue. Similarly, pipeline approaches like LIDA [22] decompose the visualization generation process by combining LLMs with visualization rules. Commercial tools have also adopted LLMs for CVA [1, 5, 53, 73, 88]. BaViSitter [19] further expands the CVA landscape by exploring multimodal interfaces, enabling users to issue commands that incorporate both natural language and interactive visual references. Our work complements this by focusing on how to systematically evaluate such interactions, especially when they involve ambiguity, context carryover, and inference.

While these tools demonstrate the potential of LLMs to make data exploration more accessible, questions remain about how practitioners actually use CVA tools in real-world settings and how they assess the quality of generated outputs. In this work, we conduct formative studies to explore practitioners' real-world use of CVA tools, uncovering their evaluation criteria and workflows when assessing CVA tool outputs. Building on these practitioner insights, we present a toolkit that operationalizes these findings into testcases, metrics, and evaluation interfaces to help systematically evaluate, compare, and improve the models and prompts used in LLM-based CVA tools before deployment.

2.2 CVA Evaluation Methods

2.2.1 Benchmarks. Standardized benchmarks (e.g., BIG-Bench [78], HELM [42]) help evaluate LLMs and system prompts at scale, by specifying inputs and expected outputs across diverse use cases, from reasoning puzzles to even basic analytics questions, making benchmarks central to progress in model evaluation. Data-centric benchmarks, like Spider [40, 95] and CoSQL [94], evaluate natural language to SQL generation. For natural language to visualization generation, the nvBench [48], nvBench 2.0 [47] and VisEval [18] test suites provide large-scale natural language to Vega-Lite mappings. However, these benchmarks have important limitations. First, they are synthetically generated rather than derived from real end-user

usage, meaning they do not reflect how practitioners actually interact with CVA tools in practice. For instance, most benchmarks focus on single-turn queries, overlooking the multi-turn conversational dynamics (such as context carryover, iterative refinement, and evolving analytical goals) that characterize real-world data exploration. Second, running and adapting these benchmarks requires programming expertise, computational resources, and technical setup (e.g., configuring databases, managing API calls, writing evaluation scripts), and time, creating barriers for practitioners, such as product managers, tool designers, and other low-code stakeholders, who want to evaluate CVA systems without extensive technical overhead.

Our work addresses these gaps through formative studies that observe and log real-world CVA usage by end-user analysts. From this empirical foundation, we derive Lexara's test case suite, which captures authentic usage patterns including composite questions, ambiguous intents, and multi-turn dynamics like context carryover during iterative analysis. Furthermore, Lexara's interface is designed to democratize CVA evaluation: practitioners can upload their own test cases, configure evaluation experiments, and explore results at multiple levels of granularity—all without writing code. This low-code approach makes systematic evaluation and experimentation accessible to a broader range of users, supporting iterative improvement of CVA tools before deployment.

2.2.2 Interactive Benchmarking Tools. A growing ecosystem of evaluation tools helps developers probe and debug model behavior. OpenAI Evals, Google's AutoSxS, ChainForge [4], EvalLM [38], and LLM Comparator [34] support test case comparison, hypothesis testing, and automated judgment. Other VA tools like PromptIDE [80], LMdiff [79], and Sequence Salience [83] assist with prompt iteration and token-level inspection. To support model evaluation and debugging, tools such as OpenAI's Evals framework [102] and Google Vertex AI AutoSxS [26] offer capabilities like side-by-side output comparison, rule-based and LLM-based judging, and hypothesis testing. However, these tools are built to evaluate single-turn, single-format outputs, usually text.

Lexara builds on these tools to offer native support for CVA-specific aspects like multi-turn analyses, comparing rendered visualizations, visualization grammars, analytical thinking in natural language response, based on datasource analysis. Its design incorporates domain-relevant ontology (e.g., ambiguity type, axis match, interactivity) and conversation-aware diagnostics, enabling more targeted debugging and benchmarking of CVA system behavior.

2.2.3 Human and Automated Evaluation Methods. Given the complexity of CVA interactions, human evaluation remains the gold standard. Experts or users must manually assess nuances and rate responses on a range of criteria. However, this process is labor- and time-intensive, and does not scale well across large prompt sets and model configurations [9, 21, 30]. To scale evaluation beyond manual methods, LLM-as-a-Judge approaches are increasingly used to assess model outputs along dimensions like coherence, correctness, or explanation quality [27, 37, 54]. These methods often correlate better with human judgment than traditional metrics, particularly in open-ended QA, vision-language tasks, and agent reasoning [15, 41]. Yet, studies show systemic biases — including self-preference, verbosity, position, and concreteness biases [72, 90, 92, 98], which can skew results if not addressed. Lexara develops a complementary hybrid human-AI evaluation approach to balance the richness of user-centered values, evaluation criteria and methods, with the scale of LLM-as-a-Judge automated methods. To ensure validity and reliability of evaluation methods, Lexara implements prompt, model, and interface–level safeguards: Evaluation criteria are derived from a formative study with practitioners and end-users. It employs few-shot prompts seeded with end-user–labeled evaluation examples that reflect analyst values and highlight common points of confusion. The LLM-as-a-Judge Recommendation feature recommends models from outside the candidate model family to reduce self-preference, and evaluation runs randomize item positioning and use per-output scoring against a reference rather than pairwise comparisons to limit position bias. Prompts explicitly instruct judges to ignore stylistic flourish and avoid rewarding verbosity by truncating or equalizing answer length. To counter concreteness and stylistic biases, Lexara uses detailed rubrics that specify grounded analytical criteria and provides end-user-annotated examples across multiple orthogonal metrics. The interface further surfaces judge rationales, JavaScript Object Notation (JSON) spec diffs, and rendered charts, enabling human inspection and override in a hybrid workflow. Finally, we validate Lexara's metrics against human raters (e.g., Cohen's κ, Spearman ρ) to ensure alignment.

2.3 Evaluation Metrics for Visualization and Analytical Language

Evaluating visualizations requires balancing correctness with interpretability and usability. Foundational work has emphasized the trade-offs between ecological validity and experimental rigor [63, 103], advocating for mixed methods [31, 39] and cognition-grounded metrics [8, 14, 43]. More recently, automatic evaluation techniques have emerged. VIS-Shepherd [58] employs multimodal or LLM critics to rate visualization quality. Vi(E)va LLM! [64] proposes a layered stack—from code similarity to insightfulness—applying measures like Jaccard similarity, SSIM, and VLAT. SimVecVis [45] encodes chart structure as latent vectors and evaluates reconstruction performance. Song et al. [74] raise critical methodological questions about evaluating LLM-generated visualizations, arguing traditional metrics fail to capture visual design's complexity and subjectivity. They advocate for nuanced, design-aware strategies considering interpretability, expressiveness, and task relevance – aligning with Lexara's graded, hybrid approach.

In parallel, natural language evaluation has matured from n-gram metrics (e.g., F1, Precision, Recall [20], BLEU [59], ROUGE [44]) to semantic measures (e.g., BERTScore [96], BLEURT [68]). Yet, these often fail to capture reasoning quality, contextual coherence, or domain-specific accuracy, especially in open-ended CVA explanations. Addressing these limitations, Lexara develops hybrid visualization and language metrics that accommodate ambiguity, support multiple correct outputs, and integrate rule-based and rubric-guided LLM-as-a-Judge pipelines. Users can inspect, override, and refine judgments, enabling interpretable, scalable automation with human-in-the-loop evaluation.

3 Formative Studies: Eliciting Real-World Use Cases, Evaluation Criteria & Workflows

While prior tools and benchmarks have advanced model evaluation, they often overlook the actual experiences of CVA practitioners, i.e., those building or using these tools in real-world settings. To ground the design of the Lexara toolkit in these practitioner perspectives, we conducted two complementary formative studies: interviews with tool developers to surface design rationales and evaluation workflows, and observational sessions with end-users to capture situated judgment during CVA scenarios. Taken together, these complementary approaches revealed both the considerations shaping system design and the situated practices of use, grounding our toolkit in the perspectives of those who build and those who rely on CVA tools.

3.1 Study 1: Tool Developers' Use Cases, Evaluation Criteria & Workflows

We conducted one-hour semi-structured video interviews with 22 professionals involved in developing CVA tools. Participants included researchers, designers, engineers, and product managers. Using a snowball sampling strategy [6, 11, 55]: we initially reached out to subject matter experts in CVA and LLM evaluation in a large technology company, who then recommended additional colleagues with relevant expertise. This sampling ensured coverage across product, research, and engineering roles engaged in building or evaluating LLM-powered CVA systems. The interviews explored four key areas: (1) use cases of CVA tools, (2) evaluation criteria and (3) workflows used to assess models and system prompts for these use cases, and (4) any challenges they faced in conducting these evaluations. All participants gave informed consent for data collection (audio, video, and usage logs). Sessions were recorded, transcribed, and thematically analyzed using an open-coding approach [16] to identify use cases, evaluation workflows and their challenges, and evaluation criteria. Refer to the supplementary materials for details on our experimental protocol, interview guides and apparatus.

3.2 Study 2: End-Users' Use Cases, Evaluation Criteria & Workflows

We conducted 45-minute lab-based sessions with 16 professional data analysts or end-users ($U1$-$U16$) across diverse domains including finance, education, healthcare, and technology. Participants held roles such as analysts, BI advisors, data architects, research scientists, consultants, and product managers. Recruitment was conducted via a visual analytics conference, supplemented by direct outreach to attendees interested in conversational AI and visualization tools. This strategy provided access to participants who had real-world VA experience but varying familiarity with conversational interfaces (six beginners, seven intermediate, three advanced). We classified participants as beginners if they reported less than one year of regular experience using conversational interfaces for data analysis. Intermediate participants reported approximately 1–3 years of experience with BI tools and occasional authoring of visualizations. Advanced participants had more than three years of experience building or maintaining analytics workflows and routinely authored or reviewed visualizations. Each session had two phases:

Phase 1: Think-Aloud CVA Interaction [15–20 min] To understand their usage of CVA tools, participants used a commonly-used commercial LLM-enabled CVA tool [73] to analyze a datasource of their choice, either from a curated gallery or their own (Appendix Table 3). A Chrome extension (Appendix Figure 5) recorded their multi-turn interactions, capturing prompts, model responses, and in-the-moment reflections. After each response, participants rated its quality using Likert-style criteria and corrected outputs when needed to reflect their expectations. Participants could also suggest custom evaluation criteria or flag inaccuracies. These real-world logs later informed the design of Lexara's test case library (see Supplementary Materials).

Phase 2: Side-by-Side LLM Response Comparison [20–25 min] Participants compared anonymized outputs from multiple models (GPT-4o, Claude-Opus-4, GPT-o3 anonymized for participants to avoid biasing) for the same user utterances (visualizations, natural language responses, and JSON grammar specifications, alongside traditional metrics (F1, Precision, Recall [20]) (Appendix Figure 6). Grammar specifications were shown to expose structural decisions such as field encodings, chart types, filters, and sort logic – details not always visible in the rendered visualizations. By surfacing specifications alongside outputs, we enabled participants to diagnose why two similar-looking charts diverged and express expectations around cross-format consistency [69, 85].

They were asked to think aloud as they compared model outputs, evaluation trade-offs, and reflected on where existing metrics fell short. All sessions worked with the same utterances from the Superstore datasource [82], which is part of the NLVCorpus benchmark [75]. We selected Superstore due to its familiar business context and rich analytical scope, which could elicit ambiguity, multi-turn reasoning, and diverse chart types [69, 70]. Using this shared datasource ensured ecological validity while enabling consistent comparisons across sessions.

3.3 Characteristics of CVA Use Cases

In addition to analyzing developer and end-user interviews, we thematically analyzed utterances from Phase 1 of Formative Study 2 (§3.2), where 16 data professionals from domains, such as finance, education, healthcare, and technology engaged in multi-turn CVA sessions ($\sum = 80$ utterances, $\mu = 5.8, \sigma = 3.1$ turns per conversation). Using a browser extension (Appendix Figure 5), we logged user–system interactions, including in-the-loop ratings, corrections, and labels for each utterance. The resulting conversations spanned a diverse set of utterance types and evaluation challenges, reflecting realistic task complexity and variation across domains (finance (8 conversations), education (3), healthcare (5)). This annotated set of utterances reflects the diversity of analytical intents and reveal key challenges in interpreting user intent during conversational interactions, described as follows:

3.3.1 Visualization Types. Thematic analysis of 64 user utterances revealed requests for a diverse range of chart types: bar chart (n=30), scatter plot (n=6), line chart (n=18), box plot (n=4), histogram (n=3),

multi-line chart (n=3). For example, for the utterance, *"Plot a scatter of* `discount` *vs.* `profit margin`*,"*, $U5$ remarked, *"I expected a scatter plot, but it gave me a bar chart. Technically valid but not what I asked"*. Such examples underscore the importance of semantic precision and visual format alignment in user expectations, highlighting the need for evaluation metrics sensitive to visualization intent and not just syntactic correctness.

3.3.2 Ambiguity in User Utterances. In analyzing the user utterances, we observed that 27 utterances exhibited some form of ambiguity, requiring the system to make context-sensitive inferences. Ambiguity in natural language, defined as the presence of multiple plausible interpretations for a single expression or request, is a well-documented challenge in CVA interfaces [2, 24, 69]. Ambiguity often arises when user intent is underspecified, field references are vague or mismatched, or contextual cues from earlier turns are required for correct interpretation. A single utterance could display multiple forms of ambiguity.

Syntactic Ambiguity. 18 utterances demonstrated syntactic ambiguity, which arises when the structure of a sentence permits multiple grammatical interpretations [32]. For example, $U13$ asked, *"Show top 10 products in furniture by sales region with high profit,"*, and responded to the result: *"I could see at least two ways to interpret that,* [Show the top 10 products in the furniture category, grouped by sales region, but only include those with high profit. or Show the top 10 products in the furniture category, by those sales regions that have high profit.] *and the model picked one* [the latter]."

Semantic Ambiguity. Of the 27 ambiguous utterances, 19 involved semantic ambiguity, wherein an utterance could plausibly map to multiple fields or concepts in the datasource, due to underspecified or imprecise language. For example, $U4$ asked, *"Show me profit over time"*, although the datasource only contained fields `Net Revenue` and `Cost`. The model inferred a plausible mapping and returned a chart plotting Net Revenue over time. The user later reflected, *"I said 'profit,' but in this datasource that could mean revenue minus cost, or maybe net sales. The model guessed* `Net Revenue`.".

Pragmatic Ambiguity. Pragmatic ambiguity arises when the meaning of a user's prompt depends on context, whether from earlier dialogue turns or implicit assumptions that are not explicitly stated in the input utterance [69, 85]. Participants encountered such ambiguity in 37 utterances, and judged systems not only by their final outputs, but by how well they interpreted underspecified prompts in light of surrounding context.

Participants often incrementally elaborated on previous prompts (19 utterances), expecting earlier filters or visual structure to persist. For example, $U7$ began with, *"Show sales by region"*, followed by, *"Now break it down by category."* They appreciated when the model preserved the region filter, noting, *"I like that it remembered my earlier region filter, but sometimes other models just dropped it."*

Ambiguity also arose when users referred back to previously mentioned entities without explicitly naming them again (32 utterances). For example, when $U12$ asked, *"Which of these categories had the highest growth?"* (following a turn which asked about product categories) *"It got confused about what 'these categories' meant and pulled in something else [the model erroneously filtered out all the categories returning a vacuous chart.]."* Participants also expected temporal or categorical filters to persist across related turns. $U2$ explained how this shaped their interpretation when they asked, *"Show only* `2023 sales`*"* next *"Now compare* `East` *vs.* `West`*."* *"When the filter carried through, the analysis made sense. When it didn't, it felt like it didn't get me and I had to keep starting over."*

Another form of pragmatic ambiguity involved both implied concepts and underspecified utterances (64 occurrences), where users referenced fields, filters, sorts, or time units without explicitly naming them, expecting the system to resolve intent through context. For example, $U5$ implicitly assumed descending order for a visualization response to the utterance, *"Top 10 states by profit."* They remarked, *"It didn't sort descending, so the top 10 wasn't actually top."*

3.4 Evaluation Criteria for CVA Use Cases

These utterances provide a foundation for systematic evaluation. Building on this foundation, we examined how practitioners actually judge the quality of CVA outputs in real-world scenarios. Through thematic analysis of the data, we found that participants consistently evaluated responses along three key categories: *Visualization Quality*, *Natural Language Quality*, and *Conversation Quality*.

3.4.1 Visualization Quality. Visualization Quality refers to how accurately and appropriately a generated visualization represents the correctness of data values, the appropriateness of chart types for the given analytical intent, and the presence of filters and sorting operations applied to the underlying data representing the visualization. Based on participant responses, we group visualization quality concerns into four key categories: *Data*, *Chart Type*, *Functionality*, and *Design*.

Data. We begin with criteria related to the data underpinning the visualization, specifically, whether the data is faithfully represented (*Data Fidelity*) and whether the fields selected align semantically with user intent, even when exact matches are absent (*Field Similarity*).

Participants emphasized that charts must truthfully and completely reflect the underlying data; any deviation, such as missing, extra, or incorrectly aggregated rows or columns, was viewed as a major breach of trust (Data Fidelity). As $U11$ points out, *"If it says Profit but it's clearly counting rows, that's a fail for me."*. Participants allowed partial credit when the underlying data was what they expected, but the analytical operation applied to the data (e.g., aggregation type) was not what they expected. For instance, $U2$ remarked, when viewing 'count' as the aggregation type rather than 'sum': *"It's not a big deal but here it seems to have missed* `Sum of Quantity`*"*).

Another critical dimension of data correctness is whether the fields appropriately reflect the user's analytical intent. This involves both syntactic matching (e.g., field names) and semantic understanding of the utterance (Field Similarity). Participants emphasized that fields bound to visual elements must align with the task intent, even if the exact names do not match. Near-misses, such as mapping a user's mention of "sales" to an attribute labeled `sales_amount`, or interpreting "date" as `Billing_Timestamp_HFD`, were typically accepted when types were compatible or meanings

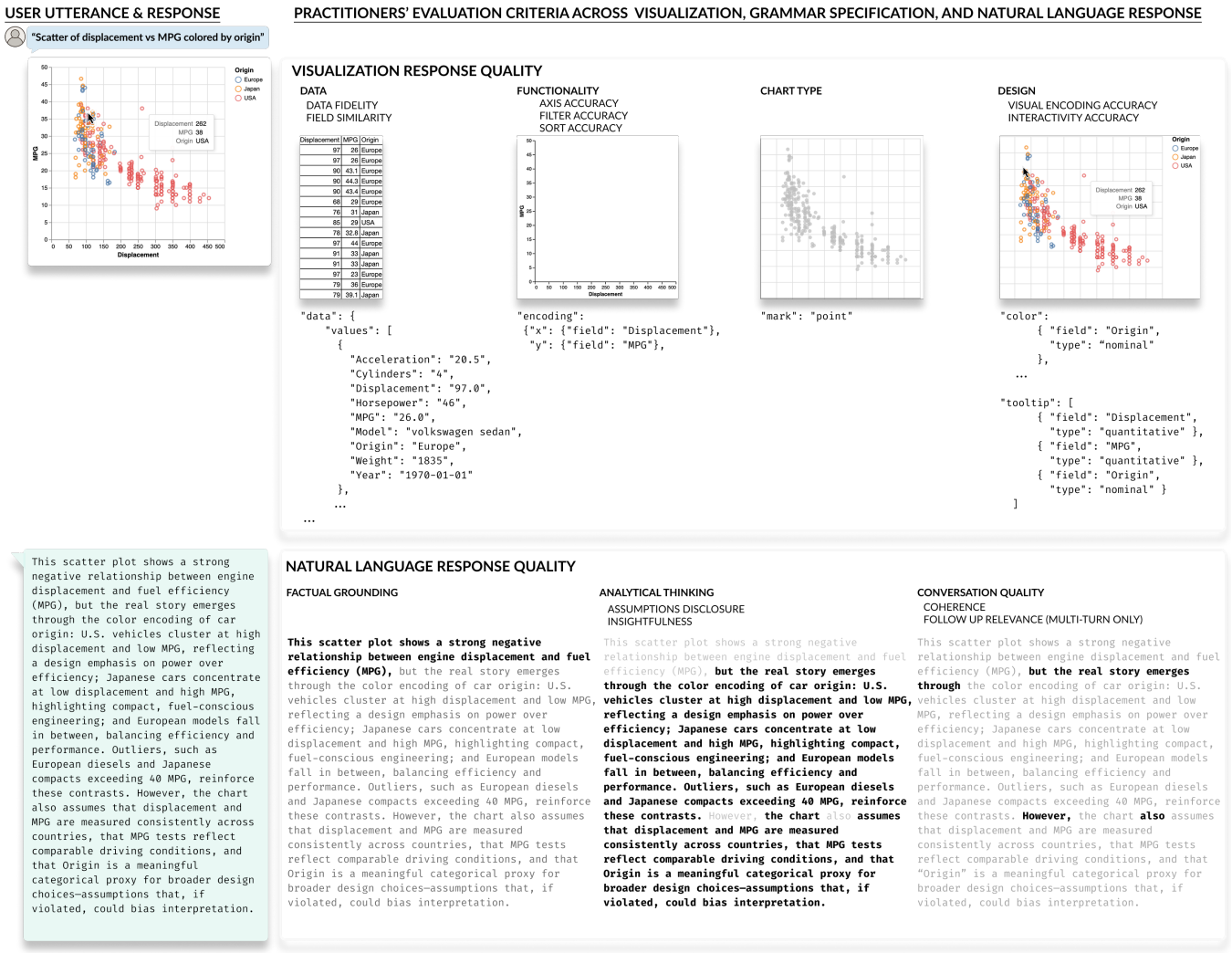


Figure 1: Illustration of how practitioners evaluate the multi-format CVA response. The example shows (left) a user utterance and corresponding model outputs, and (right) the evaluation criteria identified in our formative studies: visualization response quality assessed by looking at both the rendered visualization and grammar specification (data fidelity, chart type, functionality, design), natural language response quality (factual grounding, analytical thinking and conversation quality). This figure provides a conceptual overview and does not reflect the actual UI of any CVA system.

were semantically close. As $U14$ explained, *"in enterprise schemas, fields rarely match user wording exactly and reasonable inferences preserve flow."* Participants also appreciated when the model could make these inferences but strongly preferred transparency in the mappings. $U14$ continued, *"Recognize* `date` *even when the column is* `Billing_Timestamp_HFD`*, but tell me what you picked. Bold the exact field names you chose. "*

Chart Type. Participants consistently emphasized that the appropriateness of the chart type directly impacts how easily they can interpret and trust the visual output. They expected models to follow established visualization best practices [49], such as generating line charts for trends over time or bar charts for categorical comparisons. For instance, $U8$ noted upon seeing a bar chart being generated, *"Two models picked the wrong chart for profit per month: it should be a line. I'll still give them partial credit, but they didn't pick the best chart."*

Functionality. Participants assessed functionality correctness based on whether the visualization functioned as expected in terms of axes, filters, and sorting choices. Specifically, participants expected the axes to reflect correct field mappings, orientation, and scale, including the use of zero baselines when appropriate. $U15$ remarked, *"This one doesn't start at zero. That's misleading"* , and *"Can we get units on the axis?"* Swapped axes (e.g., X and Y reversed) were generally not treated as outright failures, since the data geometry remained valid, but participants felt they deserved a moderate score rather than full credit. However, critical errors like incorrect scale

type or missing baselines, were scored lower. *"Missing titles or units are not deal-breakers, but more like nice-to-haves"* ($U14$).

Participants viewed filter correctness as essential for analytic continuity. They expected the models to clearly indicate applied filters and penalized both over-filtering and under-filtering. $U3$ and $U5$ correspondingly stated, *"Only this model didn't filter. That's the first thing I check"* and *"This one added extra* `Year=2024`*."*

Sorting accuracy was similarly important for data prioritization tasks. Participants expected the sorting behavior to match either explicit instructions or be reasonably inferred from context. When reviewing the model output for *"Show top 10 products in Furniture by Sales"*, $U5$ commented, *"All the other models chose to* `Sales` *descending, but seems like this model chose to not do that. I guess it's ok because it's implied but not expected."*

Design. Participants assessed design quality based on how truthfully and clearly the visualization encoded information, prioritizing functionality and interpretability over any stylistic enhancements.

In particular, participants expected visual encodings, such as color, size, shape, opacity, and text labels to reflect meaningful distinctions in the data. Participants attempted to refine visual encodings through follow-up prompting: *"Color by Region; add data labels."* ($U10$).

Participants also expected interactive affordances, particularly tooltips to reveal accurate, relevant data on demand. When these elements were incorrect or incomplete, it broke their flow and raised concerns about the model's reliability. $U8$ noted, *"I hovered and it showed the wrong value. Tooltip said 'sum' but it was a count."* Participants, such as $U11$ also prompted to explicitly surface additional information - *"Include* `Sales` *and* `Profit` *in the tooltip."*

3.4.2 Natural-Language Response Quality. In addition to visualization correctness, participants carefully evaluated the accompanying natural language responses, particularly when models provided textual explanations or summaries alongside charts.

Factual Grounding. Participants consistently prioritized factual consistency between the chart and the text. They expected that descriptions include all salient facts, such as filters, measures, magnitudes, and directional trends that were encoded visually. If key facts were missing or incorrect, trust was quickly eroded and treated as high-severity errors. As $U12$ emphasized, *"If the chart and the text disagree, I stop trusting either."*

Analytical Thinking. Participants also evaluated how well the system reasoned alongside them, looking for evidence of reasoning or interpretation beyond simple description. In particular, participants appreciated when the model explicitly surfaced filters, timeframes, or aggregation logic, i.e., assumptions disclosure. This transparency helped them understand and verify how results were derived. For instance, $U11$ noted, *"If you assume individual profit values, say so."*

Higher-rated responses that synthesized trends, pointed out anomalies, and suggested comparisons or causes, were considered insightful; rather than restating input queries, participants valued responses that proactively explained what the data meant. As $U8$ shared, *"This one points out that sales dropped in Q4. That's helpful"* and *"I would like to get to a point where these systems just give me rich actionable insights not just say I did what you asked me to"* ($U11$).

3.4.3 Conversation Quality. Beyond isolated responses, participants evaluated how well CVA systems sustained coherent, context-aware conversations over multiple turns. This process included judging whether the system maintained logical flow, preserved contextual intent across prompts, and adapted outputs based on evolving dialogue.

Coherence. Participants valued responses that were logically structured, internally consistent, and free from contradictions. They often praised outputs that maintained a clear reasoning chain and articulated how different observations connected. For example, $U3$ said, *"What its saying makes sense that sales rose in Q4 so inventory dropped. So, this could impact next quarter."*

Follow-up Relevance. Participants emphasized that in multi-turn interactions, the model must retain prior context: including applied filters, selected categories, or inferred user goals. Outputs that failed to carry over context felt disjointed or inattentive. $U6$ highlighted for this utterance, *"Focus on high-growth segments in Q3 only,"* when the model added a filter with Q3's dates instead of the whole year, they commented, *"I like that since we asked about high-growth segments in Q3, this tells me what happened in Q3 only."* In Lexara, we operationalize these concerns by scoring each response in situ with respect to its preceding conversational context, rather than collapsing an entire dialogue into a single scalar score. This per-turn, context-aware design lets practitioners see where multi-turn workflows recover from errors or break down.

3.5 Workflow Challenges and Design Considerations

Through interviews with the CVA practitioners, we identified five core challenges (C1–C5) that they face when evaluating CVA systems. Each challenge is associated with recurring evaluation workflows, which, while common, often fall short of supporting systematic and scalable LLM benchmark evaluation.

C1: Fragmented, ad-hoc comparisons. Practitioners primarily relied on manual side-by-side comparisons of models and prompts. They often tested the same utterance across configurations, consolidated outputs in spreadsheets or slides, and visually inspected screenshots and specs. $T3$ and $T11$ explain their respective workflows, *"I literally had tabs for each model. One with the spec, one with screenshots, and then I'd eyeball which chart dropped categories"* and *"I feel like I'm always just context switching across all these channels, which leaves me not able to have time for really diving into types of behavior I care about."*

C2: Misalignment with domain-specific tasks. While public benchmarks like nvBench [48], Spider [40], and VisEval [18] offered a shared vocabulary, practitioners found them poorly aligned with their specific CVA use cases. They often needed to test ambiguous field references, vague temporal phrases, or domain-specific analytic tasks that benchmarks did not capture. $T17$ stated, *"Aggregate scores rarely tell me if the model will mess up axes when I ask for 'profit by segment' and such"* and $U6$ expressed frustration when describing their workflow: *"need to tailor to own data and use cases."*

C3: Unreliable transfer to actual environments. Practitioners frequently encountered mismatches between public benchmark

performance and real-world reliability. Models that performed well in papers or demos often broke down when applied to internal datasources or production workflows. As *T*20 noted, *"Prompts work for their demo datasource, but failed on ours."* To get more reliable comparisons, some developers used programmatic test suites, running benchmark scripts over curated utterances in notebooks. While this added reproducibility, the gap between test coverage and domain-specific needs remained. *T*7 explains, *"We run our benchmark on real data and look for accuracy, cost, speed columns) and export machine-readable logs."*

C4: Inaccessible and opaque evaluation outputs. Evaluation pipelines often required programming expertise and produced outputs that are typically JSON logs or console traces. This limited collaboration between engineers, PMs, and designers from participating meaningfully in the evaluation, as *T*17 points out: *"We run programmatic scripts from nvBench, but the outputs are JSON logs that PMs can't interpret."* Even for technical users, understanding *why* a case passed or failed remained difficult as few tools supported granular inspection or comparison across meaningful categories - *"No self-service so the benchmarking evaluation dashboard is hard to update and maintain, so results stay opaque to non-engineers"* (*T*7).

C5: Lack of interpretable and graded metrics. Participants reported that standard metrics, such as accuracy, BLEU, F1 rarely captured the graded, nuanced correctness required in CVA. Many cases involved partial correctness (e.g., correct chart but misleading axis), where binary scoring fell short, as *T*14 pointed out, *"In our work, accuracy isn't just yes or no. Sometimes it's close enough to be useful, other times a valid looking chart is misleading."*

4 Design Considerations For A CVA Evaluation Toolkit

Guided by the evaluation workflows and challenges (C1–C5) outlined in §3.5, we define seven design goals (D1–D7) to guide the evaluation of an LLM-based CVA toolkit (i.e., a software system that integrates reusable components for building, testing, and analyzing CVA test cases). Each goal is grounded in observed needs and translated into concrete design strategies, which we realize in the Lexara toolkit (§5).

D1: Lower the barrier to systematic benchmarking. Enable low-code practitioners to set up, run, and interpret benchmarking experiments with minimal effort based on challenges **[C1]** and **[C4]**. By providing templates for datasources, utterance sets, rubrics, and sensible defaults for metrics and comparisons, the toolkit should support reproducible evaluations without requiring programming skills, a need emphasized in HCI literature [36, 66].

D2: Tailor evaluations to real-world CVA use cases. To address challenges of misalignment with real-world CVA tasks (**[C2]**, **[C3]**), the toolkit should support benchmarking on user-specific data, tasks, and prompts, a paradigm advocated for contextualized LLM evaluation aligned with practitioners' tasks and goals [25].

D3: Scale evaluations with speed and reliability. To address the bottlenecks in manual comparison (**[C1]**) and opaque tooling (**[C4]**), the toolkit should support scalable, repeatable experiments across many utterances, prompts, and models, enabling efficient comparisons and rapid iteration [23].

D4: Compare across formats. To address challenges in manual, fragmented comparisons (**[C1]**), the evaluation toolkit should support reasoning on alignment across multiple output formats (i.e., rendered visualizations, natural language explanations, and underlying chart specifications) [51].

D5: Link overviews to instance-level insights. To address limitations in fragmented comparisons and opaque tooling (**[C1]**, **[C4]**), the toolkit should allow practitioners to fluidly navigate between high-level aggregate metrics and fine-grained utterance-level results. This supports practitioners in identifying performance subsets across task types, semantic categories, or failure modes that are most relevant to their analysis goals [46].

D6: Support context-aware diagnostic analysis. Evaluation toolkits should enable practitioners to interpret model behavior in relation to specific analytic contexts (**[C1]**, **[C4]**). Drawing from prior work in model interpretability [23, 66], the toolkit should identify when models succeed or fail (e.g., across utterance types, data domains, or prompt strategies), and how those outcomes emerge through qualitative and quantitative patterns in model and prompt behavior.

D7: Make metrics interpretable and actionable for handling multiple plausible answers. To address limitations in binary metrics (**[C5]**), CVA evaluations should employ transparent, graded metrics that can accommodate multiple valid outputs. Metrics should clearly communicate what is being measured, why it matters, and how the score was derived to support trust and actionability for model and prompt selection [25, 36].

5 Lexara: A User-Centered CVA Evaluation Toolkit

Building on the design considerations, we developed Lexara, a user-centered evaluation toolkit that operationalizes the findings from our formative studies. The toolkit comprises three complementary components:

- **Test Cases from Real-world CVA Conversations** [D2, D6] [§5.1]: A curated suite of multi-turn user queries annotated with expected outputs and labeled for prevalent CVA challenges such as ambiguity, inferred fields, and context carryover.
- **CVA Evaluation Metrics** [D7][§5.2]: A set of interpretable metrics derived from practitioners' evaluation criteria covering visualization quality, analytical natural language response quality, and conversational coherence, designed to handle graded correctness and multiple plausible answers.
- **CVA-Specific Interactive Evaluation Tool** [D1, D3, D4-6] [§5.3]: A low-code interface for setting up and running evaluations, comparing multi-format outputs across models, prompts, and turns, and linking aggregate metrics to fine-grained diagnostics based on structured test case templates.

5.1 Lexara's Test Cases from Real-world CVA Conversations

Each test case is based on a specific datasource and is represented in YAML/JSON format. The datasource file includes the following fields: `title`, `data-source-fields` the data attributes, each with a `name` and `fieldValues`, a column-vector of data values. Each test case includes: `Conversation ID`, set of `Utterances` with canonical phrasing (i.e., a representative or standard formulation of the intent) and participant-authored variations, `Labels` denoting chart type, ambiguity type, context-handling, and inferencing types. The `expected response` is provided in two formats: (i) a visualization specification (in a JSON schema similar to Vega-Lite [87], covering fields, encodings, transforms, filters, and sorts) and (ii) a natural language explanation that faithfully describes the chart while surfacing assumptions made by the model (e.g., inferred fields or grouping logic). We expose these templates in the User Interface (UI) to give practitioners fine-grained control over datasources and labels at the cost of some upfront schema familiarity.

To ensure these `expected response` reflect real-world practitioner or expert consensus, test cases were sourced either from (i) end-user analyst interactions during the formative study, where participants corrected model outputs during real-world usage, or (ii) from prior literature and benchmarks [18, 48, 75]. To ensure quality, each expected response was independently reviewed by two CVA domain experts. In cases of disagreement, a third expert adjudicated the final answer. Inter-rater reliability analysis (Cohen's Kappa = 0.81) confirms a high level of agreement across annotators. Since many tasks often have multiple plausible answers, we mitigate subjectivity by explicitly labeling the sources of ambiguity (i.e., syntactic, semantic, and pragmatic) and supporting multiple acceptable expected answers when justified. Lexara's evaluation framework applies graded metrics that capture partial correctness on a continuum, enabling finer-grained diagnostic feedback beyond binary judgments. Because the metrics compute scores by comparing model outputs against these expected CVA responses, the current toolkit is explicitly designed for curated test suites, which as mentioned in the Formative study, many teams already maintain. The full test suite is included in the Supplementary Materials and can be directly uploaded into the Lexara toolkit for benchmarking.

5.2 Lexara's User-Centered Evaluation Metrics

The practitioner evaluation criteria described in detail in §3.4 are implemented in code and their graded nature is illustrated through examples in this section. For examples of how Lexara evaluates model responses using these metrics see Table 1. The metrics are designed to operate over structured visualization grammars and natural language text, and do not assume vision inputs or explicit tool-calling support. To reflect the nuanced judgments practitioners make in evaluating CVA outputs, the metric scores are scaled to express partial correctness on a continuum rather than as binary outcomes. For instance, a model output that captures the general intent of the query but omits a key filter or misrepresents an encoding might receive a score of 70 rather than 50. This scaling is informed by formative user studies, where practitioners rated outputs on Likert-style scales and provided justifications reflecting degrees of acceptability. We anchor partial credit scores to thresholds that align with rubric-based assessment theory [12] and graded benchmarking techniques [7, 28, 60] for supporting diagnostic debugging. This graded approach to benchmarking enables practitioners to distinguish between responses that are technically plausible but incomplete, versus those that are outright misleading or irrelevant.

5.2.1 Visualization Quality Metrics. Metrics systematically measure the quality of visualization responses and range from 0-100%. Calculated by comparing the expected and actual visualization grammar spec and accommodating for partial credit or multiple plausible answers.

Data. Refers to whether the visualization truthfully represents the underlying datasource. This includes checking for fidelity of rows, columns, and aggregations, as well as semantic alignment of selected fields with user intent.

- **Data Fidelity:** Checks if the visualization faithfully represents the underlying datasource (rows, columns, and processing like aggregations or means).

  function Score_Data_Fidelity(expected, actual)
  if expected equals actual **then**
  return 100
  else if same rows and fields(expected, actual) AND minor aggregation difference(expected, actual) **then**
  return 70
  else
  return 0
  end if
  end function

  Examples: Expected: Sum of Quantity = 200 vs Actual: Count of Quantity = 200 → Score = **70%** (aggregation mismatch). Expected: 100 rows vs Actual: 80 rows → Score = **0%** (data missing).
- **Field Similarity:** Checks if bound fields match intended fields; partial credit if fields are semantically equivalent or have matching data types.

  function score_field_similarity(expectedSpec, actualSpec, datasourceMeta)
  $E_x \leftarrow$ expectedSpec.encoding.x.field
  $E_y \leftarrow$ expectedSpec.encoding.y.field
  $A_x \leftarrow$ actualSpec.encoding.x.field
  $A_y \leftarrow$ actualSpec.encoding.y.field
  function canon(f, meta)
  names $\leftarrow \{ meta[f].name \} \cup$ meta[f].aliases
  return stem(lowercase(join(names)))
  end function
  function cosSimStems(s1, s2)
  $\vec{v}_1 \leftarrow$ bow(s1); $\vec{v}_2 \leftarrow$ bow(s2)
  if $\|\vec{v}_1\| = 0$ or $\|\vec{v}_2\| = 0$ **then**
  return 0
  end if
  return $\frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\|\|\vec{v}_2\|}$
  end function
  $S_x \leftarrow$ cosSimStems(canon(E_x, datasourceMeta), canon(A_x, datasourceMeta))

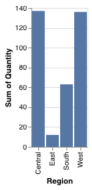
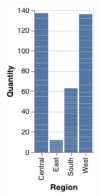
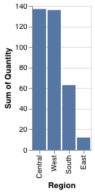
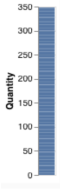
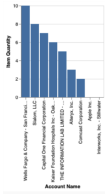
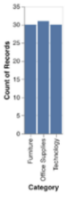
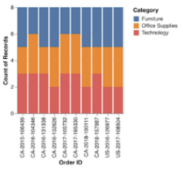
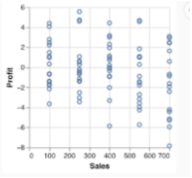
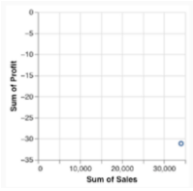
Test	User Utterance	Expected Response	Model Response	Metrics
1 Turn 1	<Quantity> on y-axis and <Region> on x-axis	 A vertical bar chart with Region on the x-axis and the sum of Quantity on the y-axis (range 0–140). The Central region has the highest total (137), with other regions lower.	 A bar chart comparing Quantity (y-axis) across Regions (x-axis), allowing a quick scan of how quantities differ by region.	Data Fidelity = 100% ; Field Similarity = 100% ; Chart Type Similarity = 100% ; Axis Accuracy = 100% ; Filter Accuracy = 100% ; Sort Accuracy = 100% ; Visual Encoding Accuracy = 100% . **Overall Viz** = 100% . NL: Factual Grounding 70% ; Assumptions Disclosure = 40% ; Insightfulness = 40% ; Coherence 90% . **Overall NL** 65% .
1 Turn 2	Sort by <Quantity>	 A vertical bar chart of Regions, sorted in descending order by the sum of Quantity, so the highest-quantity region (Central, 137) appears first.	 A bar chart ordered by Quantity values for each Region; the description notes sorting by quantity but does not specify whether the order is ascending or descending.	Data Fidelity = 100% ; Field Similarity = 100% ; Chart Type Similarity = 100% ; Axis Accuracy = 50% ; Filter Accuracy = 100% ; **Sort Accuracy** = 0% ; Visual Encoding Accuracy = 100% . **Overall Viz** 65% . NL: Factual Grounding 60% ; Assumptions Disclosure = 40% ; Insightfulness = 20% ; Coherence 80% . **Overall NL** 50% .
2	show me top accounts by attendees	 A horizontal bar chart of the Top 10 Account Names, ranked by total (sum) Item Quantity and sorted from highest to lowest, making it easy to spot the highest-volume accounts.	 A horizontal bar chart listing the Top 10 Account Names ranked by sum(Item Quantity), with bars sorted descending to highlight the largest contributors. (Accounts with null Item Quantity don't contribute to the totals.)	Data Fidelity = 100% ; Field Similarity 100% ; Chart Type Similarity = 100% ; Axis Accuracy = 100% ; Filter Accuracy = 100% ; Sort Accuracy = 100% ; Visual Encoding Accuracy 95% . **Overall Viz** 98% . NL: Factual Grounding = 100% ; Assumptions Disclosure = 100% ; Insightfulness = 20% ; Coherence 95% . **Overall NL** 85% .
3	count orders by categories	 A bar chart that counts orders per Category—the x-axis lists categories and the y-axis shows the count of Order IDs.	 A bar chart breaking down Order IDs by Category, with each category represented by a different color to distinguish groups.	Data Fidelity = 100% ; Field Similarity = 100% ; Chart Type Similarity 50% ; Axis Accuracy = 100% ; Filter Accuracy = 100% ; Sort Accuracy = 100% ; Visual Encoding Accuracy 85% . **Overall Viz** 90% . NL: Factual Grounding 60% ; Assumptions Disclosure = 20% ; Insightfulness = 20% ; Coherence 90% . **Overall NL** 50% .
4	revenue versus earnings	 A scatterplot with Sales on the x-axis and Profit on the y-axis, one point per record. Plotting raw values reveals how profit changes with sales.	 A scatterplot comparing total Sales (x) to total Profit (y) so it collapses to a single point representing Sum of Sales vs. Sum of Profit.	Data Fidelity = 0% ; Axis Accuracy = 95% ; All other viz metrics = 100% **Overall Viz** = 85% ; NL: Factual Grounding 25% ; Assumptions Disclosure = 0% ; Insightfulness = 20% ; Coherence 90% . **Overall NL** 30% .

Table 1: Worked examples showing how LEXARA metrics assess visualization and natural language quality responses interpretively.

$S_y \leftarrow$ cosSimStems(canon(E_y, datasourceMeta), canon(A_y, datasourceMeta))
$S \leftarrow (S_x + S_y)/2$
$T_x \leftarrow$ [meta[E_x].dataType = meta[A_x].dataType]
$T_y \leftarrow$ [meta[E_y].dataType = meta[A_y].dataType]
bonus $\leftarrow$ 10 if ($T_x \wedge T_y$) else 0
return min(100, $100 \times S + bonus$)
end function

Examples: Related data type and semantic fields: `Order Date vs Ship Date` → base semantic similarity 0.77; both temporal ⇒ + 10% type bonus. Score: min(100%, 77% + 10%) = **87%**
Unrelated data types and semantic fields: `Region vs Category` → base semantic similarity 0.29; both discrete dimensions ⇒ +10% type bonus. Score: min(100%, 29% + 10%) = **39%**
Missing axis: `Sales vs (missing Y axis)` → no comparable field, similarity = 0.0; no type bonus. Score: **0%**
Note: For two-axis charts, compute S_x and S_y separately via stemmed cosine similarity, set $S = (S_x + S_y)/2$, then add a single 10% bonus only if both axis data types match; cap at 100%.

Chart Type. measures how well the model's chosen mark type aligns with the mark types that Tableau's Show Me engine [49] recommends for the same set of data fields. We (i) run Show Me on the expected fields to obtain a ranked list of recommended chart/mark types, then (ii) compare the model's chosen mark against that list:

function SCORE_CHART_SIMILARITY(expectedSpec, actualSpec, datasourceMeta)
$F \leftarrow$ { *expectedSpec.encoding.x.field*, *expectedSpec.encoding.y.field* }
$R \leftarrow$ ShowMeRecommend(F, datasourceMeta) ▷ ranked list of mark types
$M \leftarrow$ normalizeMark(actualSpec.mark) ▷ e.g., "line", "bar", "area", "point", "table"
if $R = \emptyset$ **then**
return 0%
else if $M = R[1]$ **then** ▷ top recommendation
return 100%
else if $M \in R$ **then** ▷ other Show Me recommendation
return 50
else
return 0%
end if
end function

Examples: Best recommended chart: Data = time series, Model = `line` → Score = **100%**. Plausible but not best: Data = time series, Model = `area` → Score = **50%**. Alternate plausible: Data = category + measure, Model = `pie` → Score = **50%**. Poor choice: Data = two measures, Model = `table` → Score = **0%**.

Functionality Captures whether the visualization operates correctly in terms of axes, filters, and sorts. A functionally correct chart reflects accurate axis assignments, scale and baselines, appropriate filtering, and expected ordering.

- **Axis Accuracy.** Evaluates whether the X/Y axes use the *intended fields* and *parameters*, giving graded credit for semantic proximity of field names and incorporating data-type compatibility directly into the similarity score.

function SCORE_AXIS_ACCURACY(expectedSpec, actualSpec, datasourceMeta)
$E_x \leftarrow$ expectedSpec.encoding.x.field; $E_y \leftarrow$ expectedSpec.encoding.y.field
$A_x \leftarrow$ actualSpec.encoding.x.field; $A_y \leftarrow$ actualSpec.encoding.y.field
$S_x \leftarrow$ cosSimStems(canon(E_x, datasourceMeta), canon(A_x, datasourceMeta))
$S_y \leftarrow$ cosSimStems(canon(E_y, datasourceMeta), canon(A_y, datasourceMeta))
$T_x \leftarrow \mathbf{1}_{(meta[E_x].dataType = meta[A_x].dataType)}$
$T_y \leftarrow \mathbf{1}_{(meta[E_y].dataType = meta[A_y].dataType)}$
$S'_x \leftarrow 0.9 \times S_x + 0.1 \times T_x$
$S'_y \leftarrow 0.9 \times S_y + 0.1 \times T_y$
$S \leftarrow (S'_x + S'_y)/2$
$score \leftarrow 100 \times S$
if axesSwapped(expectedSpec, actualSpec) **then**
$score \leftarrow 0.5 \times score$
end if
if wrongScaleOrBaseline(actualSpec) **then**
$score \leftarrow 0.7 \times score$
end if
return min(100, $score$)
end function

- **Filter Accuracy** Measures how well applied filters match the expected set, allowing partial credit when *field names* are semantically close and *values* normalize to the same concept; adds a small bonus when matched fields share data types. *Examples:* Extra filter: expected {Year=2023}, actual {Year=2023, Region=West} → **50%** (one match over soft union), +10% if types align; Semantic field/value match: expected {Region=West}, actual {SalesRegion=West} → high semantic similarity and added value equivalence ⇒ **100%**; Wrong value: expected {Month=Jan}, actual {Month=Feb} → **0%**

function SCORE_FILTER_ACCURACY(expectedSpec, actualSpec, meta)
$E \leftarrow$ normalizeFilters(expectedSpec.transform.filter)
$A \leftarrow$ normalizeFilters(actualSpec.transform.filter)
matched $\leftarrow$ 0
usedA $\leftarrow \emptyset$
usedE $\leftarrow \emptyset$
for each $e \in E$ **do**
best $\leftarrow$ 0
bestIdx $\leftarrow$ none
for each $a \in A$ not in usedA **do**
$sf \leftarrow$
cosSimStems(
canon($e.field$, meta),
canon($a.field$, meta))
if $sf \geq \tau_f$ **and** valuesEquivalent(e, a) **then**
$sim \leftarrow (sf + opMatch(\text{e,a}))/2$
if $sim > best$ **then**

```
                    best ← sim
                    bestIdx ← a
                end if
            end if
        end for
        if bestIdx ≠ none then
            matched ← matched + best
            usedA ← usedA ∪ {bestIdx}
            usedE ← usedE ∪ {e}
        end if
    end for
    matchCount ← |usedE|
    unionSize ← |E| + |A| − matchCount
    if unionSize = 0 then
        return 100
    end if
    base ← 100 × matched / unionSize
    base ← min(100, base)
    typesAgree ←
        allMatchedTypesEqual(usedE, usedA)
    if typesAgree then
        bonus ← 10
    else
        bonus ← 0
    end if
    return min(100, base + bonus)
end function
```

- **Sort Accuracy:** Assesses whether sort *fields* and *directions* match, granting graded credit for semantic proximity of the sort key and a type-consistency bonus.

```
function SCORE_SORT_ACCURACY(expectedSpec, actual-
Spec, datasourceMeta)
    E ← expectedSpec.sort          ▷ (field, direction)
    A ← actualSpec.sort            ▷ (field, direction)
    if E is none and A is none then
        return 100
    end if
    if E is none xor A is none then
        return 0
    end if
    S_f ← cosSimStems(canon(E.field, meta), canon(A.field,
meta))
    D ← 1.0 if E.dir = A.dir;
        0.5 if E.dir ≠ A.dir;
        0 if A.dir missing when E.dir specified
    base ← 100 × S_f × D
    bonus ← 10 if meta[E.field].dataType = meta[A.field].dataType
else 0
    return min(100, base + bonus)
end function
```

Examples: Right field, wrong direction: Sales `desc` vs. Sales `asc` → $S_f = 1.0$, $D = 0.5$; 50% (+10% if type matches) ⇒ up to **60%** ; Semantic key: Revenue `desc` vs. SalesAmount `desc` (treated as same concept) → $S_f = 0.8$, $D = 1.0$, type bonus = 10 ⇒= **100%** ; Missing sort: expected sort `desc` , actual none → **0%**

Design. Focuses on how clearly and meaningfully information is encoded. This involves assessing the accuracy of visual encodings (e.g., color, size, labels), the appropriateness of design choices for interpretability, and whether interactive elements (e.g., tooltips) provide correct contextual information.

- **Visual Encoding Accuracy.** Measures how truthfully and clearly the chart maps data to visual channels, using a graded score per channel (*color, shape, opacity, text, size*), then averaging across channels. For each channel we combine: (i) semantic match of bound field, (ii) data-type consistency, and (iii) design best-practice adherence (e.g., hue for nominal, gradient for quantitative, contrast, legibility).

```
function SCORE_ENCODING_ACCURACY(expectedSpec, ac-
tualSpec, meta)
    channels ← {color, shape, opacity, text, size}
    scores ← [ ]
    for each c in channels do
        E ← expectedSpec.encoding.c
        A ← actualSpec.encoding.c
        if E is none and A is none then
            append 100 to scores
            continue
        end if
        presence ← 0
        if E and A present then
            presence ← 100
        else if A present then
            presence ← 50
        end if
        sem ← 0
        if E and A present then
            sem ← 100×
                cosSimStems(
                    canon(E.field, meta),
                    canon(A.field, meta))
        end if
        typeOK ← 0
        if E and A present then
            if meta[E.field].dataType = meta[A.field].dataType
then
                typeOK ← 100
            end if
        end if
        practice ← 100×
            bestPracticeScore(c, E, A, meta)
        s_c ←
            0.3 · presence+
            0.4 · sem+
            0.1 · typeOK+
            0.2 · practice
        append s_c to scores
    end for
    return mean(scores)
end function
```

Examples: Exact, well-designed: Color = Region (nominal) with categorical palette, text labels on bars, size not used → Score **100%**; Good semantics, minor design gap:Color = Sales (quantitative) but uses categorical palette (should be gradient) → Score **80%**; Alternative acceptable: Expected no opacity, actual uses opacity to mitigate overplotting in dense scatter (quantitative) → Score **80%**; Mismatched channel: Size = Region (nominal) with many categories → Score **40%**; Missing expected channel: Expected color by Region, actual has no color encoding → Score = **0%** for color (averaged across channels)

- **Interactivity Accuracy** Scores how well interactive affordances (e.g., tooltips, selection, zoom/pan, drill-down) support accurate, relevant, and usable reading. Allows *multiple correct answers*: if the actual design includes an alternate but reasonable set of fields or interactions that covers the required information, it receives partial credit. We combine: (i) *coverage* of required info, (ii) *correctness* of shown values/aggregations, and (iii) *usability* best practices (formatting, units, concise lists, interaction consistency).

  function SCORE_INTERACTIVITY_ACCURACY(expectedSpec, actualSpec, datasourceMeta)
  $R \leftarrow$ requiredTooltipFields(expectedSpec) ▷ fields bound to x,y,color,size + key filters/measures
  $A \leftarrow$ actualTooltipFields(actualSpec)
  $M \leftarrow$ matchFieldsSemantically(R, A, datasourceMeta) ▷ one-to-one best semantic matches
  coverage $\leftarrow 100 \times \frac{|M|}{\max(1, |R|)}$ ▷ fraction of required info covered
  correctness $\leftarrow$ 100× mean(AGGOK(m) for $m \in M$) ▷ aggregations/values align with spec
  extras $\leftarrow$ 100× normalized count of additional relevant fields in $A \setminus M$
  redundancyPenalty $\leftarrow$ 100× penalty for duplicates/noise (e.g., repeating same measure twice)
  tooltipScore $\leftarrow$ clamp$\Big($ 0.6·coverage +0.3·correctness +0.1·extras −0.1·redundancyPenalty $\Big)$
  interactionsExpected $\leftarrow$ interactionsFrom(expectedSpec) ▷ e.g., selection, zoom, drill
  interactionsActual $\leftarrow$ interactionsFrom(actualSpec)
  interMatch $\leftarrow$ 100× softJaccard(interactionsExpected, interactionsActual) ▷ partial credit for alternates
  consistency $\leftarrow$ 100× INTERACTIONCONSISTENCYOK() ▷ tooltips/selection respect filters/sorts/encodings
  usability $\leftarrow$ 100× avg(FORMATTINGOK(), UNITSSHOWN(), BREVITYOK()) ▷ avoid long lists, show units
  $w_t \leftarrow 0.6,\ w_i \leftarrow 0.2,\ w_c \leftarrow 0.1,\ w_u \leftarrow 0.1$
  return w_t·tooltipScore +w_i·interMatch +w_c·consistency +w_u·usability
  end function

  Examples.
  Complete & correct tooltips: Show X, Y, color field, and measure with correct aggregation, formatted with units; selection highlights series consistently → Score **100%**. Alternate but acceptable: Expected {Month, Sales, Region}, actual shows {Month, SalesAmount, RegionName, Profit}; semantic matches for required fields plus a relevant extra; formatting OK → Score **95%**. Partially correct: Shows X/Y but omits filter context and uses count instead of sum in tooltip → Score **60%**. Redundant/noisy: Long tooltip lists with duplicate fields, inconsistent units → Score **50%**. Missing: No tooltips or interactive affordances when expected → Score = **0%**.

5.2.2 Natural Language Response Quality. These metrics evaluate natural language responses in CVA. Four out of five of them are implemented as LLM-as-a-Judge metrics grounded in human-in-the-loop evaluations from Formative Study 2 [§3.2], where participants rated model outputs turn-by-turn and articulated why certain responses were accurate, incomplete, or misleading. These qualitative judgments provided a rich, annotated dataset used to craft few-shot rubrics for automated scoring. For example, vague or contradictory explanations were labeled by participants as "low coherence," while detailed responses naming filters, time frames, and assumptions were judged "high on assumptions disclosure." By embedding these annotated examples into the judge prompts, we ensured that LLM-based ratings would better align with practitioner expectations, capture graded correctness, and remain interpretable. This approach also gave end-users a traceable influence on the design of automated evaluation, supporting transparency and trust in the resulting metrics.

Factual Grounding: This is calculated programmatically and ensures the explanation conveys the semantically similar facts as the visualization (measures, magnitudes, directions).

function SCORE_FACTUAL_GROUNDING(expectedText, actualText)
  $e \leftarrow$ embedding(expectedText)
  $a \leftarrow$ embedding(actualText)
  similarity $\leftarrow$
    cosine(e, a)
  if contradiction(expectedText, actualText) **then**
    return 0
  else
    return 100 × *similarity*
  end if
end function

Examples:

- Expected: "Profit climbed 8% year-over-year" vs Actual: "Profit up eight percent year-over-year" → Score = **100%**.
- Expected: "Profit climbed 8% year-over-year" vs Actual: "Profit improved year-over-year" → Score = **70%** (magnitude missing).
- Expected: "Profit climbed 8% year-over-year" vs Actual: "Revenue grew 8%" → Score = **0%** (wrong measure).

Analytical Thinking:

- **Assumptions Disclosure:** [1] Evaluates whether the response surfaces relevant assumptions (filters, time frames, aggregation choices). *Examples:*

[1] This is an LLM-as-a-Judge Metric, please see the human annotated few-shot examples used to prompt this in the supplementary materials.

  - Actual: "This assumes the Region filter is set to North America and values are aggregated monthly" → Score = 4 (relevant assumptions).
  - Actual: "These insights assume data excludes returns, is filtered to 2023, and that Sales reflects total revenue, not net" → Score = 5 (comprehensive).
- **Insightfulness:**[1] Captures depth of analysis, identifying trends, exceptions, and actionable implications. *Examples:*
  - Actual: "Sales increased" → Score =2 (basic observation).
  - Actual: "From Q1 to Q4, Electronics in the West grew 25%, Apparel in the South fell 10%, suggesting a shift in seasonal demand" → Score=5 (rich, actionable).

Conversation Quality:

- **Coherence:**[1] Evaluates whether the response is internally consistent and logically structured. *Examples:*
  Actual: "Sales are up, but that means profit is lower, so we should cut inventory" → Score = 1 (contradictory). Actual: "Inventory is down. Sales are good. Profit is low. Maybe a trend?" → Score = 2 (disorganized). Actual: "Sales increased, possibly leading to higher profit. Inventory dropped, which might be a concern" → Score = 3 (mostly coherent). Actual: "Sales rose in Q4, contributing to higher profits. Inventory dropped significantly, which could create supply issues next quarter" → Score = 4 (well-structured). Actual: "Q4 sales increased 20%, profits rose 15%, while inventory declined 30%, raising fulfillment concerns for Q1" → Score = 5 (clear and precise).
- **Follow-up Relevance:**[1] Checks whether the response remains grounded in prior turns of the conversation (multi-turn only). Examples:
- Previous user utterance: "Focus on high-growth segments."
  Response: "I included segment data like you asked earlier."
  → Score = 2 (minimal linkage).
- Previous user utterance: "Focus on high-growth segments in Q3 only."
  Response: "This line chart filters to Q3 only, shows Technology outperforming all others, continuing the trend we saw last week."
  → Score = 5 (fully grounded).

5.3 Lexara's Interactive CVA Evaluation Tool

Building on the real-world test cases [§5.1] and user-centered metrics [§5.2], effective CVA evaluation also requires an interactive tool that simplifies setup, supports low-/no-code use, and surfaces actionable insights. The Lexara interface is designed to address fragmented workflows, opaque outputs, and the disconnect between aggregate metrics and specific failures, supporting our design considerations [D1–D7]. It enables users to run benchmarks on custom data and prompts, compare multi-format outputs side by side, and drill down from high-level metrics to turn-level diagnostics.

5.3.1 Evaluation Setup. (Figure 2) Lexara enables low-code practitioners to run systematic CVA benchmarking experiments across multiple models and system prompts **[D1–D3]**. The setup workflow includes intentional defaults, helpful templates, and guardrails for error handling. Each run treats a configuration of one or more models and system prompts applied to a shared set of test cases as the object of evaluation. That is, the toolkit focuses on evaluating model–prompt behavior in the CVA backend (e.g., datasource interpretation, visualization specification, explanation quality), rather than instrumenting every component of a deployed CVA application such as the UI, logging pipeline, or enterprise orchestration.

Upload datasource and test case files. Practitioners can upload their own datasources and test case files, or select from a sample in the 'Select Test Case' dropdown. To create their own files, they are given guidance on template with required fields, structure, and an example file stub. When the uploaded files do not match requirements, the error messages are specific in explaining how to fix the issue. Upon successful upload, they can preview the datasource table and test cases in the Evaluate Test Cases table where each test case ID is a conversation, each row is a user utterance in the conversation with labels, clicking on '+' next to the test case ID unfurls all the rows/utterances in multi-turn conversations.

Specify system prompts. To compare prompt variants, practitioners can define multiple system prompts using an example prompt template with required variables (e.g., `datasource`, `utterance`, and expected `JSON visualization grammar structure`). Prompts are numbered for traceability in results.

Select Models. The toolkit currently supports 10 models: the latest OpenAI (`GPT-5, GPT-5-mini, GPT-5-nano, o3, o4-mini`), Anthropic (`claude-opus-4, claude-3.7-sonnet`, and Deepseek models (`r1`). These were selected as commonly used models by participants in the formative studies. Practitioners are encouraged to bring their own API keys for models. In our current implementation, Lexara treats all evaluated models as text-only chat endpoints: we send natural language utterances and receive JSON visualization specifications plus textual explanations. Extending the toolkit to exercise models' full multimodal and tool-use capabilities is left as future work (see §7.2).

Select Metrics. [D7] Practitioners can select specific visualization and natural response quality metrics, or traditional natural language metrics (F1, Precision, Recall). Each metric has a tooltip with the definition (see §5.2 for details). **LLM-as-a-Judge Recommendation:** For metrics that require LLM-as-a-Judge, Lexara recommends one by annotating the model in the drop-down with `(recommended)` next to its name. This recommendation is made heuristically by following best practices to reduce bias [98]: selecting the strongest model outside the LLM families of models getting evaluated to reduce self-bias. The judge models are instructed to ignore style or truncate or equalize answer length to avoid verbosity bias. Furthermore, to align the Judge with practitioners' evaluation criteria, we apply few-shot learning by sharing examples of ratings distilled from the formative study with end-users.

Specify Test Cases. Practitioners can specify individual test cases or contiguous ranges of test case IDs. Leaving this blank will execute all test cases in the file.

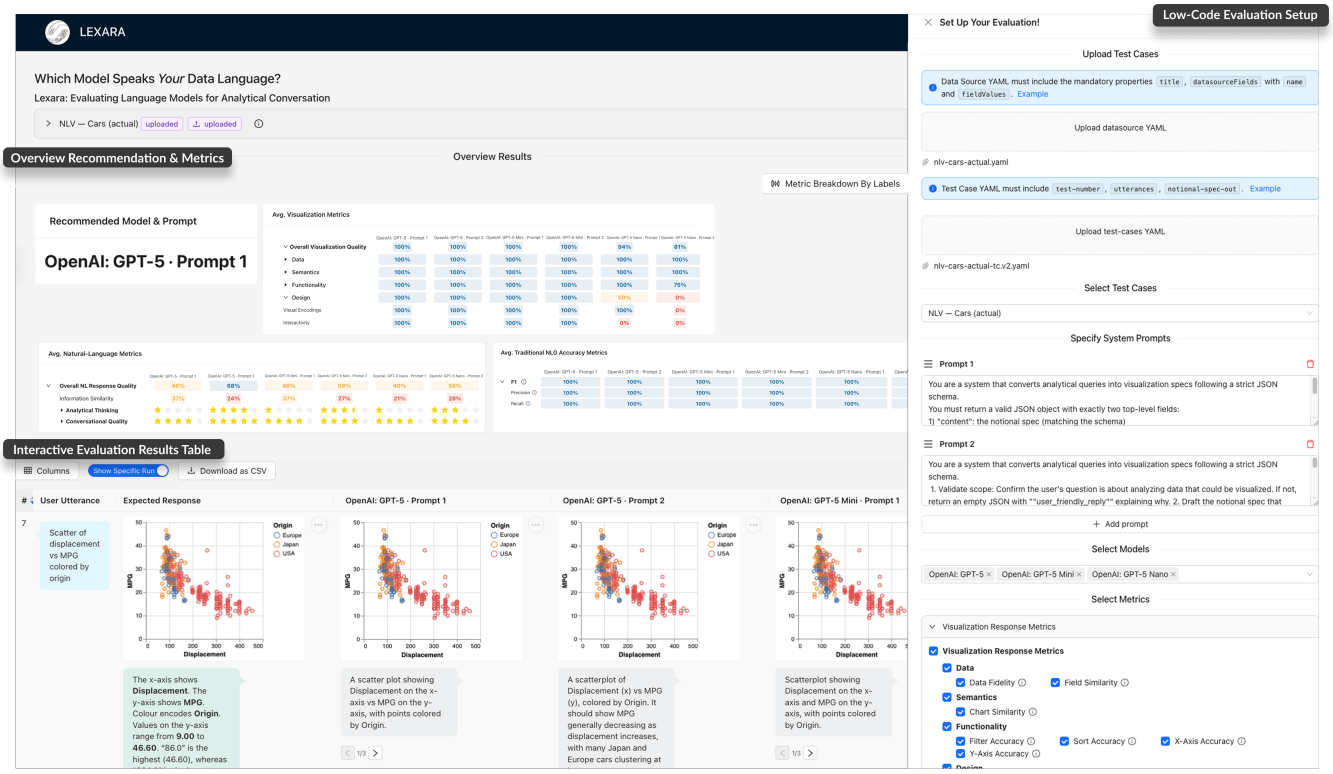


Figure 2: LEXARA's interactive CVA evaluation interface supports two core workflows: (1) an Evaluation Setup Panel where practitioners upload datasources, define test cases, specify prompts, models, expected outputs, and configure CVA-specific metrics; and (2) an Interactive Results Table that streams model outputs—visualizations, structured specs, and natural language—side-by-side. The table enables multi-granular inspection, with expandable metric categories, on-hover explanations, and tools to trace divergences between expected and actual outputs.

Specify Number of Runs. By default, LEXARA executes three replications per (model × system-prompt × judge) configuration to reduce run-to-run variance. Practitioners may adjust the number of replications to 1–5: set 1 for exploratory spot checks and up to 5 for increased reliability during benchmarking. Practitioners can examine the results from each run as they stream in to the Evaluation Results table.

Pressing the *Evaluate* button initiates an evaluation experiment, while the Stop button immediately terminates the ongoing evaluation.

5.3.2 Interactive Test Cases Table. **[D4-6]** As evaluations run, the table dynamically populates with multi-format responses, visualizations, natural language outputs, and JSON specs for each model × prompt combination. Corresponding metrics for visualization quality, language accuracy, and traditional Natural Language Generation (NLG) scores are computed in real time. To support focused analysis, the table offers spreadsheet-like features: columns can be filtered, hidden, or frozen, enabling flexible, side-by-side comparisons.

Multi-format Response Cells (Figure 3) display each model's response as an interactive Vega-Lite chart and accompanying natural language explanation. These are rendered using a custom engine that transforms high-level JSON specs into Vega-Lite, allowing direct visual comparison with expected outputs.

Hierarchical Metrics Drill-Down Cells (Figure 4) Each column represents a metric category: Visualization Response, Natural Language Response, and Traditional NLG. A color-coded scale (red = low, yellow = mid, blue = high) highlights performance at a glance. The drill-down follows an overview + detail-on-demand pattern. For example, the Visualization column initially shows an overall quality score. Expanding reveals subcategories like *Data*, *Semantics*, *Functionality*, and *Design*, which further break down into granular metrics (e.g., *Data Fidelity*, *Sort Accuracy*, *Visual Encodings*). This layered design supports rapid scanning with selective deep dives. Interactive explanations enhance interpretability: hovering on a score reveals the expected vs. actual output (e.g., *Sort: Expected descending*, *Model: none*). For LLM-as-a-Judge ratings, hover text includes the model's justification.

Clicking **Examine Viz Grammar Differences** (Figure 3 (Right)) opens a JSON spec diff viewer, enabling side-by-side comparison of visualization grammars to trace structural discrepancies like missing filters or mismatched encodings.

This combination of overview, detail, and contextual explanations transforms the metric table into an interactive evaluation workspace, enabling practitioners to interpret scores, and understand how and why they were assigned.

5.3.3 Overview Panel. **[D5-6]** The Overview panel provides an entry point into evaluation by surfacing progress, recommendations,

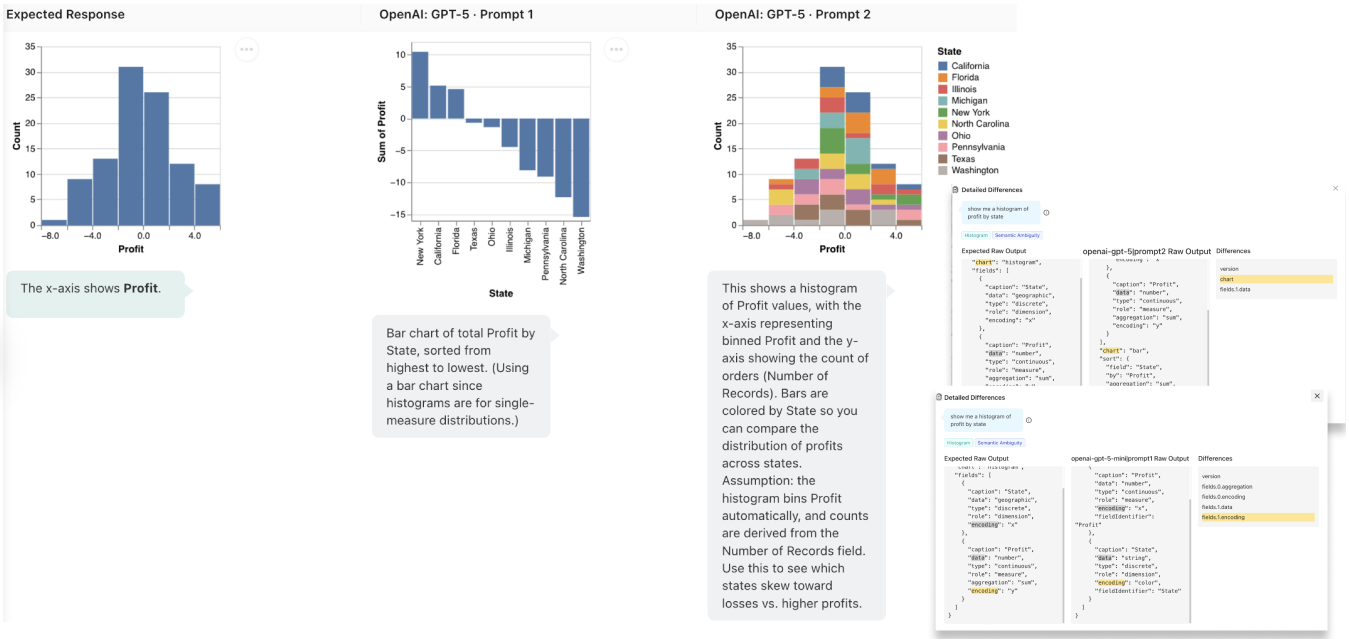


Figure 3: For each user request, the system aligns expected and actual outputs across three formats: visualizations, natural language explanations, and JSON specifications. By surfacing detailed differences (e.g., encodings, aggregations, chart types), the interface enables practitioners pinpoint divergences, understand model behavior, and diagnose strengths or failure modes for various analytic tasks.

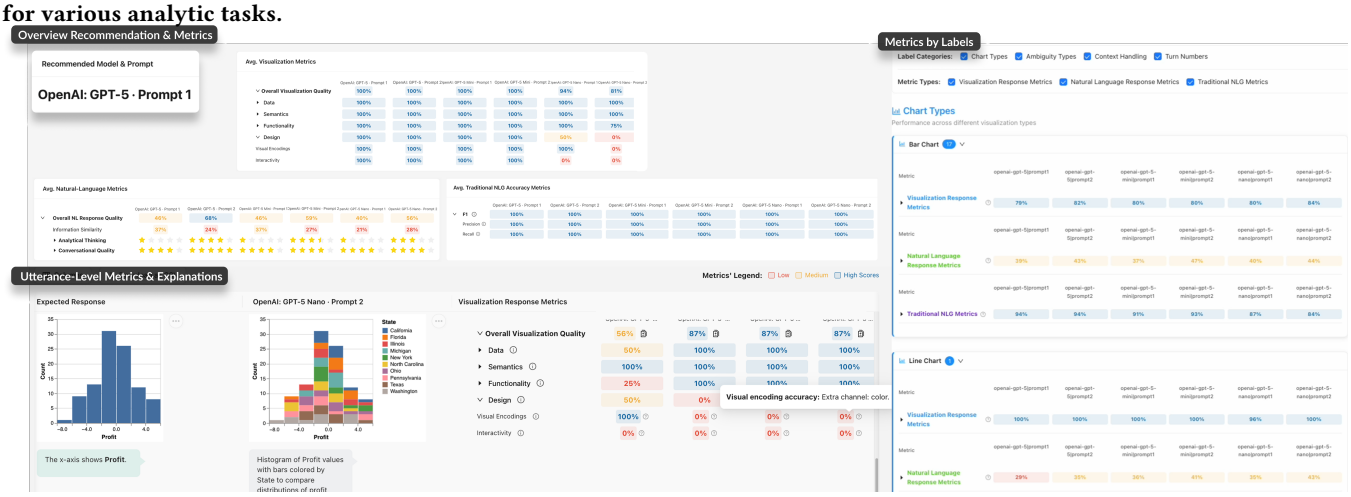


Figure 4: The overview panel (top left) highlights recommended model–prompt pairs and aggregated metrics. The label view (top right) breaks down results by chart type, ambiguity, and context-handling. The utterance-level view (bottom) contrasts expected vs. actual responses with detailed metric explanations.

and aggregate insights. A real-time progress bar tracks completion across models, prompts, and test cases. To mitigate the risk that headline scores anchor practitioners on partial or unstable results, the overview panel is made visible once all utterance-level evaluations for a given run have completed. Each recommendation card links back to the underlying metric table and per-utterance views, encouraging users to treat the overview as a starting hypothesis rather than a definitive judgment. Once evaluation is complete, the system highlights a data-driven recommendation for the best-performing model–prompt pair, based on aggregated metrics.

To support interpretability, the panel includes **Overview Metric Cards** (Figures 2, 3) summarizing performance across three key dimensions: visualization, natural language, and traditional NLG metrics. Each card supports drill-down inspection, enabling practitioners to trace how high-level scores emerged from individual test case dimensions, mirroring the overview + detail pattern in the hierarchical metrics table.

The panel also features a **Metrics-by-Label** view (Figure 4), which breaks down results by test case annotations such as chart type (e.g., bar, line, scatter), ambiguity class (semantic, syntactic,

pragmatic), and contextual intent (e.g., slot-filling, reference resolution, filter carryover). This layered, faceted view helps practitioners move from global trends to specific breakdowns, clarifying *why* a model–prompt pairing was recommended, and where it succeeds or fails.

5.3.4 Implementation. The Lexara toolkit is implemented as a distributed web application with a React frontend [65] using TypeScript [86], Ant Design components [3], and a Flask backend [57]. The system follows a microservices architecture with asynchronous job processing and real-time streaming. A Redis Queue library handles background tasks [67], while connection pooling and semaphore-based concurrency control manage API rate limits and prevent resource contention.

5.3.5 Toolkit Deployment. The toolkit is deployed at https://lexara-6b38293fcdac.herokuapp.com/ and has been iteratively refined based on feedback from engineers, designers, PMs, and researchers across multiple CVA teams at a large technology company. It is also available as an open-source project on GitHub https://anonymous.4open.science/r/Lexara-CVA-Eval-280B/README.md to support broader adoption and experimentation within the CVA research and practitioner communities.

6 Field Deployment Diary Study: Method & Findings

To explore how practitioners use Lexara in real-world settings, we deployed Lexara within a large technology company that develops a range of CVA products, and recruited a subset of the CVA tool developers from our earlier formative study cohort to participate in a two-week structured diary study [§3.2].

6.1 Study Setup

We recruited six CVA tool developers (two engineers, one designer, and three product managers) with prior experience evaluating LLMs and prompt strategies for visualization tasks. Before the study, participants joined a 30-minute orientation introducing Lexara's core features and potential evaluation use cases (e.g., testing prompts, comparing models, authoring test cases).

During the two-week study, participants completed daily evaluation tasks using their own data and prompts, submitting structured logs detailing datasources, test cases, model and prompt selections, rationale, outputs, observations, and confidence levels. A 60-minute debrief interview followed to review experiments and workflows.

We collected orientation and debrief transcripts, daily diary logs, Lexara exports, and participant-authored test cases. All materials were pseudonymized. We conducted thematic analysis using a hybrid inductive–deductive coding approach [16] to identify patterns in real-world evaluation practices and tool gaps.

We now report findings from the two-week diary study with six participants ($P1$–$P6$). Our goal was to understand how practitioners used Lexara in their daily evaluation workflows, what value they derived from its test cases, metrics, and interactive features, and remaining gaps. Participants conducted 38 evaluation experiments across 57 uniquely authored test cases (rest from existing test case suite), comparing 10 LLMs and 6 system prompts (see Table 2).

6.2 Lexara's Test Cases Captured Real CVA Use

Participants valued the realism and variety in the curated test cases, especially the inclusion of multiple chart types and multi-turn follow-ups that mirrored real CVA workflows. As $P6$ noted, "*one prompt and then the next, then remove nulls, felt like how someone might actually interact.*" The presence of expected reference outputs also helped participants calibrate model responses; $P3$ reflected, "*It was helpful to have the reference of what the expected output would have been.*"

However, the current YAML-based authoring workflow posed challenges, particularly for participants with non-engineering roles. PMs and designers found it difficult to contribute, with $P1$ remarking, "*The YAML barrier makes it harder for PMs to contribute new cases.*" Participants suggested more accessible authoring tools, such as a point-and-click interface to define utterances, labels, and expected outputs without needing to write structured files.

6.3 Lexara's Metrics Were Nuanced and Interpretable

Participants appreciated that Lexara's metrics were not black-box scores but came with drilldowns and on-hover explanations. These features clarified why a score was given, making the results more actionable. As $P2$ noted, "*Hovering over visual encodings told me it added this extra channel color, which was the fundamental difference.*" Even those who did not use the hovers extensively still emphasized the value of the metric suite. $P5$ remarked, "*What I would like to keep [are] the metrics for sure … this was the part that is missing from other evaluation tools.*"

Participants also valued the hierarchical, collapsible structure of the metrics. $P3$ appreciated being able to expand only the relevant sections, while $P2$ filtered by specific dimensions of interest, such as axis or sort accuracy. The interpretability helped participants connect high-level scores with concrete differences in outputs.

Importantly, participants recognized the value of graded correctness and support for multiple plausible outputs. $P4$ explained, "*Accuracy isn't just yes or no. Sometimes it's close enough to be useful; other times a valid-looking chart is misleading.*" Some participants wanted more customization, such as evolving the metrics to better reflect visualization best practices ($P3$), readability or tone ($P4$), or performance measures like latency and cost ($P2$).

6.4 Lexara Supported Running a Variety of Experiments at Scale

Participants conducted a variety of evaluation experiments by holding some variables constant while probing others. Common comparisons included large vs. compact models, cross-family competitors, and different prompting strategies, such as persona-prompting (e.g., "*as a data visualization expert*" $P3$), few-shot examples (e.g., "*learn color coding based on these more engaging visualizations*" $P6$), or prompts in different languages to test tone and formality ($P4$). Some explored edge cases, like how multi-turn interactions handled filters and sorts.

Workflows generally followed an overview-to-detail pattern aligned with Lexara's design: selecting test cases, running model - prompt combinations, reviewing summary metrics and system recommendations, inspecting rendered outputs, and drilling into JSON

PID	# of Evaluation Experiments	# of Test Cases Executed	# of Models	# of System Prompts
P1	5	62	4	3
P2	10	103	10	2
P3	3	16	5	1
P4	8	45	5	1
P5	2	32	4	2
P6	10	85	10	2

Table 2: Summary of evaluation experiments conducted by participants (*P*1–*P*6) during the two-week diary study. Each participant ran multiple experiments across varying test cases, models, and prompts. These figures provide a quantitative overview of how Lexara was appropriated in practice, complementing our qualitative insights.

diffs. *P*2 valued the concise summary: "*I appreciate the top-level recommendation.*" P4 praised the structured comparison: "*Great for like-for-like comparisons.*" *P*6 added, "*It's cool to see the model outputs side by side and compare how they generated the viz.*" Some participants ran parallel experiments in separate tabs to test multiple hypotheses simultaneously.

6.5 Lexara Facilitated Granular, Multi-Format Evaluation

Lexara enabled participants to inspect model differences across multiple granularities, formats, turns, and runs, aligning with design goals (*D*4–*D*6). The side-by-side view of expected and actual outputs emerged as the most intuitive entry point. *P*3 appreciated that it surfaced divergences clearly: "*I liked the side-by-side, and having the notional spec JSON to see exactly where differences came from.*" *P*6 emphasized it made follow-up failures obvious: "*Seeing it next to the reference chart made that obvious.*" Compared to spreadsheets or tab-switching, participants found this interface reduced cognitive load.

The JSON diff viewer added a deeper diagnostic lens, helping explain score mismatches when charts looked similar. *P*3 noted: "*Two vizzes looked the same, but the score wasn't 100. JSON showed a tooltip difference.*" They used it to uncover hidden mismatches in encodings: "*Empty graphs still got high scores, but the axis binding was off.*" In one sequence, *P*1 initially tested GPT-4o-mini but noticed misaligned encodings when inspecting results. Lexara's JSON diff viewer highlighted the mismatch, prompting them to switch to Claude Opus 4, saying "*The Viz Grammar Diff is pretty handy!… More confident than before! The difference in scores do seem to correlate better with the observed differences in the viz response.*"

Participants also used the overview metrics and recommendation cards as starting points. While some appreciated the concise summaries (*P2*: "*I appreciate the top-level recommendation*"), others cross-checked them with their own assessments. *P*6 challenged a suggestion: "*It recommended the mini, but my tally favored another.*" *P*1 noted subtle visual flaws not reflected in the metrics. P1 said, *"I feel quite confident. I saw quantitatively a stark difference in the performance of the models and also by clicking through the outputs could tell qualitatively that Claude Sonnet was matching the expected outputs more often."* While initial use presented a learning curve (*P*4: "*The breakdown was almost too much at first*"), participants ultimately integrated the interface into sensemaking workflows, validating model and system prompt outputs, interpreting discrepancies, and reasoning across CVA's multi-format outputs.

7 Validating Lexara's Metrics

To assess whether Lexara's metrics align with expert judgment, we conducted a quantitative validation study comparing metric outputs against human ratings of CVA responses.

7.1 Method

We sampled $N = 120$ CVA responses from the formative and diary-study experiments (§3, §6), stratified with coverage across: different metrics and score ranges (lower, medium, higher thirds of scores); different ambiguity labels (e.g., syntactic, semantic, pragmatic); different task types (e.g., descriptive vs. comparative vs. trend analysis). Each sampled response included: the datasource schema, the user utterance (and conversational context where applicable), and the model-produced visualization, JSON specification, and natural language response.

Two raters (R1, R2), diary study participants familiar with Lexara scored each CVA response on all the metrics defined in §5.2, using the native scale associated with that metric. Raters completed a training phase of 10 pilot items, where they could clarify how to interpret the rubric with the authors, but not discuss individual items or specific scores. These pilot items were not considered in the final analysis.

For each metric, we computed (1) inter-rater reliability between the two raters using linear-weighted Cohen's κ, quantifying how consistently raters applied the rubric to the same set of responses and (2) metric–human alignment between the mean human score per response (simple average of the two raters) and metric, and then calculated Spearman's rank correlation ρ.

7.2 Results

7.2.1 Inter-Rater Reliability. Human raters showed moderate to high agreement on most metrics (see Appendix Figure 7a). Across visualization metrics, linear-weighted κ ranged from 0.45 to 0.78 (median $\kappa = 0.65$), highest for Data Fidelity, Field, Chart Type, and Axis, Filter, Sort Accuracy, and lowest for Interactivity, reflecting the greater subjectivity of interaction design. For natural language response metrics, κ ranged from 0.46 to 0.80 (median $\kappa = 0.63$), with Factual Grounding and Coherence exhibiting higher agreement

than Insightfulness and Follow-up Relevance. These results suggest that human raters can apply Lexara's metrics reliably. Some fluctuation is expected in metrics that capture subjective or experiential qualities like interactivity judgments as they depend on evaluators' expectations about analytic workflows, prior tool experience, and task context. Rather than treating these metrics as decisive indicators of overall system quality, we recommend interpreting them as diagnostic signals. These metrics are intended to complement, not override, more objective correctness measures, guiding targeted debugging and design iteration rather than serving as pass or fail criteria.

7.2.2 Metric–Human Correlation. Lexara's metrics aligned well with human judgments: Data Fidelity, Field Similarity, and Chart Type Similarity showed strong rank correlations with human ratings (Spearman's ρ in the range 0.68˜0.79, see Appendix Figure 7b). In natural language response metrics: Factual Grounding exhibited the strongest alignment ($\rho = 0.82$). The remainder of the natural language response metrics correlated at ($\rho = 0.57$–0.71, see Appendix Figure 7b), lower than Factual Grounding but comparable to human–human agreement.

While Lexara incorporates multiple safeguards to reduce LLM-as-a-Judge biases, disagreements between automated judges and human evaluators still occur, particularly on subjective dimensions. For example, in one evaluation instance, a model-generated bar chart included correct data mappings and filters but omitted interactive tooltips. Human raters penalized this omission due to its impact on exploratory analysis, whereas the automated judge assigned a relatively high interactivity score based on the presence of a rendered chart and valid specification structure. Lexara surfaces such disagreements explicitly in the interface by exposing per-metric scores, judge rationales, and underlying visualization specifications alongside rendered outputs. This allows practitioners to inspect where and why judgments diverge, override automated scores when appropriate, and treat automated evaluations as assistive rather than authoritative. By supporting this human-in-the-loop workflow, Lexara enables users to balance scalability with contextual judgment, reinforcing trust in the evaluation process rather than obscuring uncertainty.

7.2.3 Model Alignment with Human Preferences. For each of the ten LLMs evaluated in the diary study (§6.1), we computed the mean score by averaging all visualization metrics and all natural language response metrics across all test cases and system prompts that participants executed with that model. Independently, at the end of the diary study we had asked participants rankings of each model they had interacted with: (i) an overall 1–5 quality rating for CVA tasks and (ii) a rough rank ordering of models from best overall for CVA to worst (allowing ties). We converted these into per-model human preference scores by (a) normalizing each participant's ranks to [0, 1], (b) averaging across participants for each model (ignoring models a participant had not used), and (c) using these averages as the human judgment baseline. We then computed Spearman rank correlations between each model's human preference score and its mean visualization and natural language response score, quantifying how well the model performance aligns with practitioners' preferences. Models that participants perceived as stronger for CVA tasks generally obtained higher mean scores across both visualization and natural language response metrics (see Appendix Figures 7c, 7d). The rank correlation between human preferences and Lexara's overall visualization score was $\rho = 0.79$ ($p < 0.01$), and $\rho = 0.74$ ($p < 0.05$) for the natural language response score. These results do not constitute a full comparative benchmark, but they provide a sanity check that Lexara's metrics track practitioners' qualitative impressions at the coarse model level, complementing the per-metric validation against expert ratings.

8 Limitations and Future Work

Lexara contributes to a growing body of research on evaluating LLMs, with a particular focus on the unique demands of CVA. Prior work has offered important building blocks: large-scale text benchmarks for reasoning and language quality [42, 101]; visualization-specific test suites [18, 48]; and interactive LLM evaluation toolkits [4, 33, 38]. However, these efforts typically focus on single-turn, text-only outputs or require significant programming effort to evaluate, making them less suitable for evaluating multi-format, multi-turn, and ambiguity-rich CVA workflows. Lexara addresses these limitations by integrating interpretable metrics, grounded real-world test cases, and an accessible low-code interface tailored for CVA evaluation.

8.1 Broadening Scope for Sustained Use of LLM-based CVA Evaluation Toolkits

While our diary study demonstrates Lexara's usefulness for systematic LLM evaluation, several limitations suggest future directions. The test suite's coverage, though designed for multi-turn, multi-format CVA conversations, remains bounded by datasources, domains, and intents from our formative studies and existing benchmarks. The toolkit currently assumes access to expected CVA responses for each test case, reflecting its diagnostic benchmarking role. Extending the interface for ad-hoc exploratory use remains an open design challenge. Lexara currently supports common chart types (bar, line, scatter, histogram, box plots, multivariate line, pie charts). Building on Vega-Lite's expressive grammar, the toolkit is extensible to broader visualizations (maps, Sankey diagrams, heatmaps) by authoring new test cases using declarative Vega-Lite specifications. Additional contributions may uncover new test cases as conversational intents evolve from analytic questions to dashboard authoring or data stories.

Broader adoption could reveal how sustained use impacts trust, model selection, and deployment practices. Future iterations should expand the test suite across more domains, user types, and data modalities. As users upload custom test cases, features should support ethically grounded, opt-in contribution mechanisms, raising questions around consent, credit, and data quality.

The current YAML/JSON authoring workflow poses challenges for non-technical stakeholders despite offering transparency and control. The diary study revealed desire for easier pathways (CSV templates, point-and-click builders), but simplification risks sacrificing precision and reproducibility critical for benchmarking. A promising direction is collaborative authoring: engineers specify formal test logic while designers and analysts contribute natural language utterances and qualitative labels, aligning with HCI research on participatory evaluation and mixed-expertise workflows.

This work does not address operational concerns like cost, latency, prompt/model drift—critical for large-scale deployment. Incorporating these aspects could enable more holistic, real-world CVA tool evaluations. We have open-sourced the project and hope the community continues to develop this.

A limitation of our validation of LLM-as-a-Judge metrics is that the human raters were Lexara-experienced and trained by the authors. While this familiarity may bias judgments toward the toolkit's rubrics and increase alignment with automated metrics, we intentionally adopted this setup for an initial quantitative validation to reduce labeling noise when applying nuanced, graded CVA criteria; this is reflected in high inter-rater reliability (Cohen's $\kappa = 0.81$). Establishing this calibrated baseline allows us to characterize metric behavior before introducing additional sources of variance. Future work should evaluate generalizability by involving independent domain experts and blinded crowd raters, comparing inter-group agreement, and using randomized and double-blind rating protocols to detect systematic bias and assess transfer beyond this expert-curated setting.

8.2 Designing CVA Metrics for More Nuanced Evaluation Strategies

Lexara extends beyond existing visualization benchmarking efforts [22, 58, 64], by introducing user-centered, graded metrics that move beyond binary checks of validity, legality, and readability. These include finer-grained measures of visualization specification fidelity (e.g., field, axis, sort accuracy) and natural language response quality (e.g., insightfulness, grounding), designed to support multi-format, multi-turn CVA tasks. However, several limitations remain.

To focus on evaluating the baseline capabilities and limitations of LLMs, our current metrics evaluate model outputs derived from text prompts and JSON specifications; they do not yet assess models' native multimodal perception of rendered charts or their performance when using external tools.

Evaluation metrics often encode subjective judgments: thresholds for specification similarity and heuristics for matching may reflect implicit normative biases[72, 90]. Over-optimization toward these metrics could obscure genuine analytic quality. We view the scores as providing a baseline check on whether CVA outputs are eligible to support analytic reasoning (e.g., correct data, appropriate encodings, factually grounded explanations), rather than as direct proxies for the quality of the human sensemaking process itself. While Lexara visualizes full conversational sequences through the metrics-by-label and drill-down views, we do not yet provide explicit trajectory measures such as the number of turns required to reach an acceptable chart or the frequency of successful self-correction.

Ecological validity also introduces variability. Real-world utterances yield multiple plausible answers, complicating reproducibility and inter-rater consistency. While Lexara's graded metrics offer partial interpretability, evaluating under ambiguity remains a broader methodological challenge in HCI and NLP evaluation [8, 9, 21].

8.3 Supporting Actionable Sensemaking in CVA Benchmarking Workflows

Our work combines interactive visualization rendering, JSON spec diffs, hierarchical metric breakdowns, and progress overviews to support more nuanced diagnosis of model and prompt configurations. However, the downside of these interface enhancements is the learning curve; rich outputs can overwhelm new users, and auto-generated recommendations occasionally diverge from practitioners' qualitative judgments, prompting additional manual review. YAML-based authoring also persists as a bottleneck, and integration with enterprise tools for collaboration, orchestration, or data authoring remains limited.

More fundamentally, Lexara functions as a diagnostic CVA benchmarking toolkit; it reveals *where* and *why* models fall short, but does not yet close the loop to support more actionable sensemaking. Future extensions could support semi-automated prompt repair [61, 99] or training data augmentation [100] based on failure patterns, transforming evaluation from a retrospective analysis into a forward-looking, feedback-driven improvement loop. This would align evaluation more closely with iterative development workflows [42, 101].

9 Conclusion

As LLMs increasingly mediate analytical reasoning and visual exploration, rigorous and user-centered evaluation becomes critical. Through formative studies with practitioners, we identified key challenges in evaluating LLMs for CVA, including test cases misaligned with real-world use cases, a lack of interpretable graded metrics, and ad-hoc fragmented evaluation workflows. We operationalize these insights into Lexara, a user-centered CVA evaluation toolkit including test cases grounded in real-world CVA use cases, interpretable metrics that account for multiple or partially-correct responses, and supporting low-code benchmarking balancing human and automatic evaluation methods. By enabling scalable, nuanced, and CVA-specific evaluation, our work contributes both conceptual and practical advances toward more transparent, trustworthy, and user-centered assessment of LLM behavior in CVA systems. The toolkit is publicly available at https://lexara-6b38293fcdac.herokuapp.com/ with open-source code at https://anonymous.4open.science/r/Lexara-CVA-Eval-280B/README.md, to support broader adoption and extension by the HCI and visual analytics communities.

Acknowledgments

We thank Yash Tyagi, Kate Mann, Jon Hendrich, and Edouard Picot for their support and feedback. We are grateful to all study participants for their time, insights, and engagement. We also thank the anonymous reviewers for their constructive feedback and suggestions, which helped strengthen this paper.

References

[1] 2025. Talk with your data — GenAI-Powered Business Intelligence (AI/BI Genie). https://www.databricks.com/product/business-intelligence/ai-bi-genie. Databricks product page; Accessed: 2025-09-05.

[2] Omar Alharbi, Shaidah Jusoh, and Norita Md Norwawi. 2012. Handling Ambiguity Problems of Natural Language Interface for Question Answering. *IJCSI International Journal of Computer Science Issues* 9 (05 2012), 17–25.

[3] Ant Design Team. 2025. Ant Design: A UI Design Language and React UI Library. https://ant.design/ Accessed: 2025-09-10.

[4] Ian Arawjo, Calvin Swoopes, Pao Siangliulue Vaithilingam, Martin Wattenberg, and Elena Glassman. 2024. ChainForge: A Visual Toolkit for Prompt Engineering and LLM Hypothesis Testing. In *CHI*. https://doi.org/10.1145/3613904.364201

[5] Sumeet Arora. 2025. Introducing Spotter: ThoughtSpot's AI Analyst for Everyone. https://www.thoughtspot.com/blog/introducing-spotter-ai-analyst. ThoughtSpot blog; Accessed: 2025-09-05.

[6] Rowland Atkinson and John Flint. 2001. Accessing Hidden and Hard-to-Reach Populations: Snowball Research Strategies. *Social Research Update* 33 (2001), 1–4.

[7] Thea Atwood and Rebecca Reznik-Zellen. 2018. Using the Visualization Software Evaluation Rubric to explore six freely available visualization applications. *Journal of eScience Librarianship* 7 (01 2018), e1122. https://doi.org/10.7191/jeslib.2018.1122

[8] Leilani Battle, Marco Angelini, Carsten Binnig, Tiziana Catarci, Philipp Eichmann, Jean-Daniel Fekete, Giuseppe Santucci, Michael Sedlmair, and Wesley Willett. 2018. Evaluating Visual Data Analysis Systems: A Discussion Report. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics* (Houston, TX, USA) *(HILDA '18)*. Association for Computing Machinery, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3209900.3209901

[9] Anya Belz, Craig Thomson, Ehud Reiter, and Simon Mille. 2023. Non-Repeatable Experiments and Non-Reproducible Results: The Reproducibility Crisis in Human Evaluation in NLP. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 3676–3687. https://doi.org/10.18653/v1/2023.findings-acl.226

[10] Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2023. Prompting Is Programming: A Query Language for Large Language Models. *Proc. ACM Program. Lang.* 7, PLDI, Article 186 (June 2023), 24 pages. https://doi.org/10.1145/3591300

[11] Patrick Biernacki and Dan Waldorf. 1981. Snowball Sampling: Problems and Techniques of Chain Referral Sampling. *Sociological Methods & Research* 10, 2 (1981), 141–163. https://doi.org/10.1177/004912418101000205

[12] Susan Brookhart. 2018. Appropriate Criteria: Key to Effective Rubrics. *Frontiers in Education* 3 (04 2018). https://doi.org/10.3389/feduc.2018.00022

[13] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman (Eds.). 1999. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[14] Sheelagh Carpendale. 2008. *Evaluating Information Visualizations*. Springer Berlin Heidelberg, Berlin, Heidelberg, 19–45. https://doi.org/10.1007/978-3-540-70956-5_2

[15] Yunfei Chang et al. 2024. A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology* 15, 3 (2024), 1–45. https://doi.org/10.1145/3641289

[16] Kathy Charmaz. 2014. Constructing grounded theory. (2014).

[17] Nan Chen, Yuge Zhang, Jiahang Xu, Kan Ren, and Yuqing Yang. 2024. VisEval: A Benchmark for Data Visualization in the Era of Large Language Models. *IEEE Transactions on Visualization and Computer Graphics* (2024).

[18] Nan Chen, Yuge Zhang, Jiahang Xu, Kan Ren, and Yuqing Yang. 2025. VisEval: A Benchmark for Data Visualization in the Era of Large Language Models. *IEEE Transactions on Visualization and Computer Graphics* 31, 1 (Jan. 2025), 1301–1311. https://doi.org/10.1109/TVCG.2024.3456320

[19] Jiwon Choi, Jaeung Lee, and Jaemin Jo. 2024. Bavisitter: Integrating Design Guidelines into Large Language Models for Visualization Authoring. In *2024 IEEE Visualization and Visual Analytics (VIS)*. 121–125. https://doi.org/10.1109/VIS55277.2024.00032

[20] Peter Christen, David J. Hand, and Nishadi Kirielle. 2023. A Review of the F-Measure: Its History, Properties, Criticism, and Alternatives. *ACM Comput. Surv.* 56, 3, Article 73 (Oct. 2023), 24 pages. https://doi.org/10.1145/3606367

[21] Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A Smith. 2021. All That's Human is not Gold: Evaluating Human Evaluation of Generated Text. *arXiv preprint arXiv:2107.00061* (2021).

[22] Victor Dibia. 2023. LIDA: A Tool for Automatic Generation of Grammar-agnostic Visualizations and Infographics using Large Language Models. *arXiv preprint arXiv:2303.02927* (2023).

[23] Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show Your Work: Improved Reporting of Experimental Results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 2185–2194. https://doi.org/10.18653/v1/D19-1224

[24] Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G Karahalios. 2015. Datatone: Managing Ambiguity in Natural Language Interfaces for Data Visualization. In *Proceedings of the 28th annual ACM Symposium on User Interface Software & Technology*. 489–500.

[25] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 1307–1323. https://doi.org/10.18653/v1/2020.findings-emnlp.117

[26] Google Cloud. 2025. Run AutoSxS Pipeline to Perform Pairwise Model-based Evaluation. https://cloud.google.com/vertex-ai/generative-ai/docs/models/side-by-side-eval. Google Cloud Documentation (accessed 2025-09-08).

[27] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A Survey on LLM-as-a-Judge. *arXiv preprint arXiv:2411.15594* (2024).

[28] Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. LLM-Rubric: A Multidimensional, Calibrated Approach to Automated Evaluation of Natural Language Texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 13806–13834. https://doi.org/10.18653/v1/2024.acl-long.745

[29] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. 2019. Improving fairness in machine learning systems: What do industry practitioners need?. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–16.

[30] David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty Years of Confusion in Human Evaluation: NLG Needs Evaluation Sheets and Standardised Definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, Brian Davis, Yvette Graham, John Kelleher, and Yaji Sripada (Eds.). Association for Computational Linguistics, Dublin, Ireland, 169–182. https://doi.org/10.18653/v1/2020.inlg-1.23

[31] Md Rafiqul Islam, Shanjita Akter, Linta Islam, Imran Razzak, Xianzhi Wang, and Guandong Xu. 2024. Strategies for Evaluating Visual Analytics Systems: A Systematic Review and New Perspectives. *Information Visualization* 23, 1 (2024), 84–101. https://doi.org/10.1177/14738716231212568 arXiv:https://doi.org/10.1177/14738716231212568

[32] Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA.

[33] Minsuk Kahng, Ian Tenney, Megha Pushkarna, et al. 2024. LLM Comparator: Visual Analytics for Side-by-Side Evaluation of Large Language Models. In *CHI EA*. https://doi.org/10.1145/3613905.3650755

[34] Minjeong Kahng, Ian Tenney, Megha Pushkarna, Michael X. Liu, James Wexler, Emily Reif, others, and Lucas Dixon. 2024. LLM Comparator: Interactive Analysis of Side-by-Side Evaluation of Large Language Models. *IEEE Transactions on Visualization and Computer Graphics* (2024).

[35] Youn-ah Kang, Carsten Gorg, and John Stasko. 2009. Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study. In *2009 IEEE Symposium on Visual Analytics Science and Technology*. IEEE, 139–146.

[36] Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking Benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 4110–4124. https://doi.org/10.18653/v1/2021.naacl-main.324

[37] Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An Open Source Language Model Specialized in Evaluating other Language Models. *arXiv preprint arXiv:2405.01535* (2024).

[38] Tae Soo Kim, Younghoon Lee, Joohee Shin, Yu-Hsiang Kim, and Juho Kim. 2024. EvalLM: Interactive Evaluation of Large Language Model Prompts on User-Defined Criteria. In *CHI*. https://doi.org/10.1145/3613904.3642216

[39] Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. 2011. Empirical Studies in Information Visualization: Seven Scenarios. *IEEE Transactions on Visualization and Computer Graphics* 18, 9 (2011), 1520–1536.

[40] Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, et al. 2024. Spider 2.0: Evaluating Language Models on Real-world Enterprise Text-to-Sql Workflows. *arXiv preprint arXiv:2411.07763* (2024).

[41] Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023. Generative Judge for Evaluating Alignment. *arXiv preprint arXiv:2310.05470* (2023).

[42] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, E. Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yüksekgönül, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, O. Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic Evaluation of Language Models. *ArXiv* abs/2211.09110 (2022). https://api.semanticscholar.org/CorpusID:263423935

[43] Innar Liiv. 2010. Towards Information-Theoretic Visualization Evaluation Measure: A Practical Example for Bertin's Matrices. In *Proceedings of the 3rd BELIV'10 Workshop: BEyond Time and Errors: Novel EvaLuation Methods for Information Visualization* (Atlanta, Georgia) *(BELIV '10)*. Association for Computing Machinery, New York, NY, USA, 24–28. https://doi.org/10.1145/2110192.2110196

[44] Chin-Yew Lin. 2004. Rouge: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. 74–81.

[45] Can Liu, Chunlin Da, Xiaoxiao Long, Yuxiao Yang, Yu Zhang, and Yong Wang. 2025. SimVecVis: A Dataset for Enhancing MLLMs in Visualization Understanding. arXiv:2506.21319 [cs.HC] https://arxiv.org/abs/2506.21319

[46] Riccardo Lunardi, Vincenzo Della Mea, Stefano Mizzaro, and Kevin Roitero. 2025. On Robustness and Reliability of Benchmark-Based Evaluation of LLMs. arXiv:2509.04013 [cs.CL] https://arxiv.org/abs/2509.04013

[47] Tianqi Luo, Chuhan Huang, Leixian Shen, Boyan Li, Shuyu Shen, Wei Zeng, Nan Tang, and Yuyu Luo. 2025. nvBench 2.0: A Benchmark for Natural Language to Visualization under Ambiguity. *arXiv preprint arXiv:2503.12880* (March 2025). https://arxiv.org/abs/2503.12880 arXiv:2503.12880.

[48] Yuyu Luo, Jiawei Tang, and Guoliang Li. 2021. nvBench: A Large-scale Synthesized Dataset for Cross-domain Natural Language to Visualization Task. *arXiv preprint arXiv:2112.12926* (2021).

[49] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show Me: Automatic Presentation for Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1137–1144. https://doi.org/10.1109/TVCG.2007.70594

[50] Paula Maddigan and Teo Susnjak. 2023. Chat2vis: Generating Data Visualizations via Natural Language using ChatGPT, Codex and GPT-3 Large Language Models. *IEEE Access* 11 (2023), 45181–45193.

[51] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 2263–2279. https://doi.org/10.18653/v1/2022.findings-acl.177

[52] Microsoft. 2025. Copilot: Your AI Companion. https://copilot.microsoft.com/. Official website, accessed September 8, 2025.

[53] Microsoft. 2025. Q&A for Power BI Business Users. https://learn.microsoft.com/en-us/power-bi/consumer/end-user-q-and-a. Microsoft Learn article; Accessed: 2025-09-05.

[54] Xuanfan Ni and Piji Li. 2024. A Systematic Evaluation of Large Language Models for Natural Language Generation Tasks. *arXiv preprint arXiv:2405.10251* (2024).

[55] Chaim Noy. 2008. Sampling Knowledge: The Hermeneutics of Snowball Sampling in Qualitative Research. *International Journal of Social Research Methodology* 11, 4 (2008), 327–344. https://doi.org/10.1080/13645570701401305

[56] OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[57] Pallets Project. 2025. Flask: A Python Microframework. https://flask.palletsprojects.com/ Accessed: 2025-09-10.

[58] Bo Pan, Yixiao Fu, Ke Wang, Junyu Lu, Lunke Pan, Ziyang Qian, Yuhan Chen, Guoliang Wang, Yitao Zhou, Li Zheng, Yinghao Tang, Zhen Wen, Yuchen Wu, Junhua Lu, Biao Zhu, Minfeng Zhu, Bo Zhang, and Wei Chen. 2025. VIS-Shepherd: Constructing Critic for LLM-based Data Visualization Generation. (06 2025). https://doi.org/10.48550/arXiv.2506.13326

[59] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[60] Aditya Pathak, Rachit Gandhi, Vaibhav Uttam, Arnav Ramamoorthy, Pratyush Ghosh, Aaryan Raj Jindal, Shreyash Verma, Aditya Mittal, Aashna Ased, Chirag Khatri, Yashwanth Nakka, Devansh, Jagat Sesh Challa, and Dhruv Kumar. 2025. Rubric Is All You Need: Improving LLM-Based Code Evaluation With Question-Specific Rubrics. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V.1 (ICER '25)*. Association for Computing Machinery, New York, NY, USA, 181–195. https://doi.org/10.1145/3702652.3744220

[61] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction Tuning with GPT-4. https://doi.org/10.48550/arXiv.2304.03277

[62] Peter Pirolli and Stuart Card. 2005. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proc. Int. Conf. on Intelligence Analysis*.

[63] Catherine Plaisant. 2004. The Challenge of Information Visualization Evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*. 109–116.

[64] Luca Podo, Muhammad Ishmal, and Marco Angelini. 2024. Vi(E)va LLM! A Conceptual Stack for Evaluating and Interpreting Generative AI-based Visualizations. arXiv:2402.02167 [cs.HC] https://arxiv.org/abs/2402.02167

[65] React Team. 2025. React: A JavaScript Library for Building User Interfaces. https://react.dev/ Accessed: 2025-09-10.

[66] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 4902–4912. https://doi.org/10.18653/v1/2020.acl-main.442

[67] RQ Contributors. 2025. RQ: Simple Job Queues for Python. https://python-rq.org/ Accessed: 2025-09-10.

[68] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. *arXiv preprint arXiv:2004.04696* (2020).

[69] Vidya Setlur, Sarah E Battersby, Melanie Tory, Rich Gossweiler, and Angel X Chang. 2016. Eviza: A Natural Language Interface for Visual Analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 365–377.

[70] Vidya Setlur and Melanie Tory. 2022. How Do You Converse with an Analytical Chatbot? Revisiting Gricean Maxims for Designing Analytical Conversational Behavior. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–17.

[71] Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai, and Jianmin Wang. 2022. Towards Natural Language Interfaces for Data Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics* 29, 6 (2022), 3121–3144.

[72] Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the Judges: A Systematic Study of Position Bias in LLM-as-a-Judge. *arXiv preprint arXiv:2406.07791* (2024).

[73] Tableau Software. 2025. Tableau Agent. https://www.tableau.com/products/tableau-agent. Accessed: 2025-09-05.

[74] Yuanfeng Song, Xuefang Zhao, and Raymond Chi-Wing Wong. 2024. Marrying Dialogue Systems with Data Visualization: Interactive Data Visualization Generation from Natural Language Conversations. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 2733–2744. https://doi.org/10.1145/3637528.3671935

[75] Arjun Srinivasan, Nikhila Nyapathy, Bongshin Lee, Steven M Drucker, and John Stasko. 2021. Collecting and Characterizing Natural Language Utterances for Specifying Data Visualizations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–10.

[76] Arjun Srinivasan and Vidya Setlur. 2021. Snowy: Recommending Utterances for Conversational Visual Analysis. In *The 34th Annual ACM Symposium on user interface software and technology*. 864–880.

[77] Arjun Srinivasan and John Stasko. 2017. Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2017), 511–521.

[78] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adri Garriga-Alonso, et al. 2023. Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models. *Transactions on machine learning research* (2023).

[79] Hendrik Strobelt, Brent Hoover, Arvind Satyanarayan, and Sebastian Gehrmann. 2021. LMdiff: A Visual Diff Tool to Compare Language Models. In *EMNLP System Demonstrations*. https://doi.org/10.18653/v1/2021.emnlp-demo.12

[80] Hendrik Strobelt, Alex Warstadt Webson, Victor Sanh, Brent Hoover, John Beyer, Hanspeter Pfister, and Alexander M. Rush. 2023. Interactive and Visual Prompt Engineering for Ad-hoc Task Adaptation with Large Language Models. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2023), 1146–1156. https://doi.org/10.1109/TVCG.2022.3209479

[81] Yiwen Sun, Jason Leigh, Andrew Johnson, and Sangyoon Lee. 2010. Articulate: A Semi-automated Model for Translating Natural Language Queries into Meaningful Visualizations. In *International Symposium on Smart Graphics*. Springer, 184–195.

[82] Tableau Software. 2019. Sample – Superstore Dataset. https://www.tableau.com/learn/sample-data. Accessed: 2025-09-08.

[83] Ian Tenney, Rishi Mullins, Brian Du, et al. 2024. Interactive Prompt Debugging with Sequence Salience. *arXiv preprint arXiv:2404.07498* (2024).

[84] James J. Thomas and Kristin A. Cook. 2006. A Visual Analytics Agenda. *IEEE Comput. Graph. Appl.* 26, 1 (Jan. 2006), 10–13. https://doi.org/10.1109/MCG.2006.5

[85] Melanie Tory and Vidya Setlur. 2019. Do What I Mean, Not What I Say! Design Considerations for Supporting Intent and Context in Analytical Conversation. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 93–103.
[86] TypeScript Team. 2025. TypeScript: Typed JavaScript at Any Scale. https://www.typescriptlang.org/ Accessed: 2025-09-10.
[87] Vega Project. [n.d.]. Vega Specification. https://vega.github.io/vega/docs/specification/. Accessed: 2025-09-09.
[88] Vijay Venugopal and Greg Michnikov. 2024. Chat with your Business Data – Conversational Analytics comes to Gemini in Looker. https://cloud.google.com/blog/products/business-intelligence/conversational-analytics-in-looker-is-now-in-preview. Google Cloud Blog; Accessed: 2025-09-05.
[89] Henrik Voigt, Özge Alaçam, Monique Meuschke, Kai Lawonn, and Sina Zarrieß. 2022. The Why and the How: A Survey on Natural Language Interaction in Visualization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 348–374.
[90] Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. Self-preference Bias in LLM-as-a-Judge. *arXiv preprint arXiv:2410.21819* (2024).
[91] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-examining whether, why, and how human-AI interaction is uniquely difficult to design. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.
[92] Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. 2024. Justice or Prejudice? Quantifying Biases in LLM-as-a-Judge. *arXiv preprint arXiv:2410.02736* (2024).
[93] Bowen Yu and Cláudio T Silva. 2019. FlowSense: A Natural Language Interface for Visual Data Exploration within a Dataflow System. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2019), 1–11.
[94] Tao Yu, Rui Zhang, He Yang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, et al. 2019. Cosql: A Conversational Text-to-Sql Challenge towards Cross-domain Natural Language Interfaces to Databases. *arXiv preprint arXiv:1909.05378* (2019).
[95] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A Large-scale Human-labeled Dataset for Complex and Cross-domain Semantic Parsing and Text-to-Sql Task. *arXiv preprint arXiv:1809.08887* (2018).
[96] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating Text Generation with BERT. *arXiv preprint arXiv:1904.09675* (2019).
[97] Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting BLEU/NIST Scores: How Much Improvement do We Need to Have a Better System?. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva (Eds.). European Language Resources Association (ELRA), Lisbon, Portugal. https://aclanthology.org/L04-1489/
[98] Lianmin Zheng, Wai-Ching Lam Chiang, et al. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *NeurIPS Datasets and Benchmarks*.
[99] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less Is More for Alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=KBMOKmX2he
[100] Denny Zhou, Nathanael Scharli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed H. Chi. 2022. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. *ArXiv* abs/2205.10625 (2022). https://api.semanticscholar.org/CorpusID:248986239
[101] Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. 2024. PromptBench: A Unified Library for Evaluation of Large Language Models. arXiv:2312.07910 [cs.AI] https://arxiv.org/abs/2312.07910
[102] Roy Ziv and Shyamal Anadkat. 2024. Getting Started with OpenAI Evals. https://cookbook.openai.com/examples/evaluation/getting_started_with_openai_evals. OpenAI Cookbook (accessed 2025-09-08).
[103] Torre Zuk, Lothar Schlesier, Petra Neumann, Mark S. Hancock, and Sheelagh Carpendale. 2006. Heuristics for Information Visualization Evaluation. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization* (Venice, Italy) *(BELIV '06)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/1168149.1168162

Appendix

Formative Study Apparatus

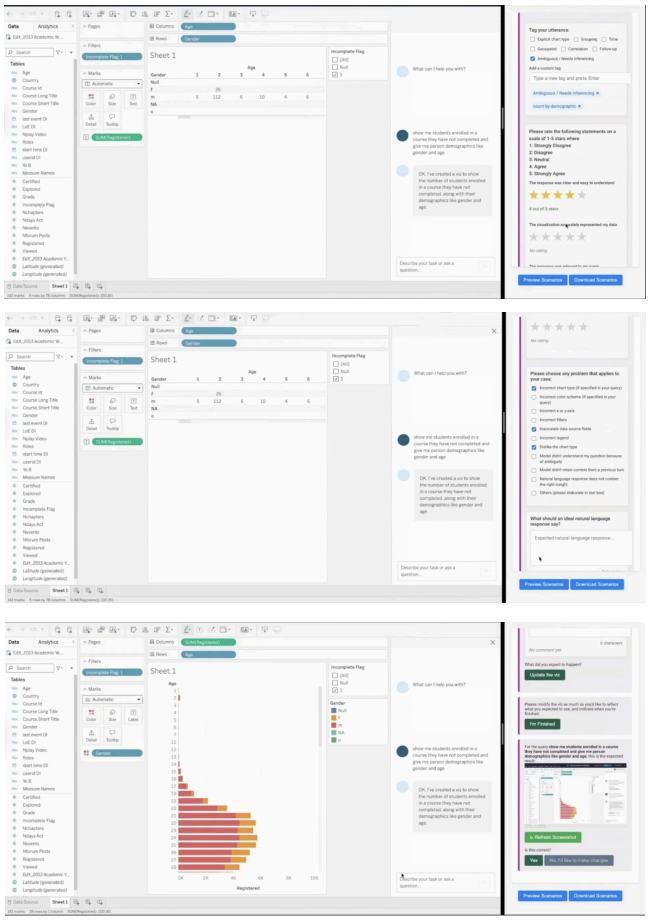


Figure 5: A browser plugin recorded participants' interaction with a popular CVA tool, capturing their utterances, model responses, in-the-loop evaluations via Likert-type scales, and corrected expected outputs.

Datasource	Domain	Details	Source
Superstore	Business & Finance	Contains information about products, sales, and profits that can be used to identify key areas of improvement within this fictitious company.	Tableau
The 2014 Inc. 5000	Business & Finance	Inc. Magazine's annual list of the 5000 fastest growing private companies in the U.S., compiled by measuring each company's percentage revenue growth over a four-year period.	Inc. Magazine
Global Sport Finances	Business & Finance	The top-paying pro sports teams and the top paid athletes.	ESPN
American University Data (IPEDS)	Education	Primary source for data on colleges, universities, and technical/vocational postsecondary institutions in the U.S.	National Center for Education Statistics
edX/HarvardX (AY 2012–2013)	Education	De-identified data from the first year (Fall 2012, Spring 2013, Summer 2013) of MITx and HarvardX courses on the edX platform.	Harvard Dataverse
Life Expectancy WHO	Healthcare	Historical and current life expectancy by country, often paired with other health indicators.	World Health Organization
Global Vaccination Coverage for COVID-19	Healthcare	Tracks immunization coverage for COVID-19 vaccines across different countries over multiple years.	World Health Organization

Table 3: Overview of datasources across business & finance, education, and healthcare domains, each linked to its original source.

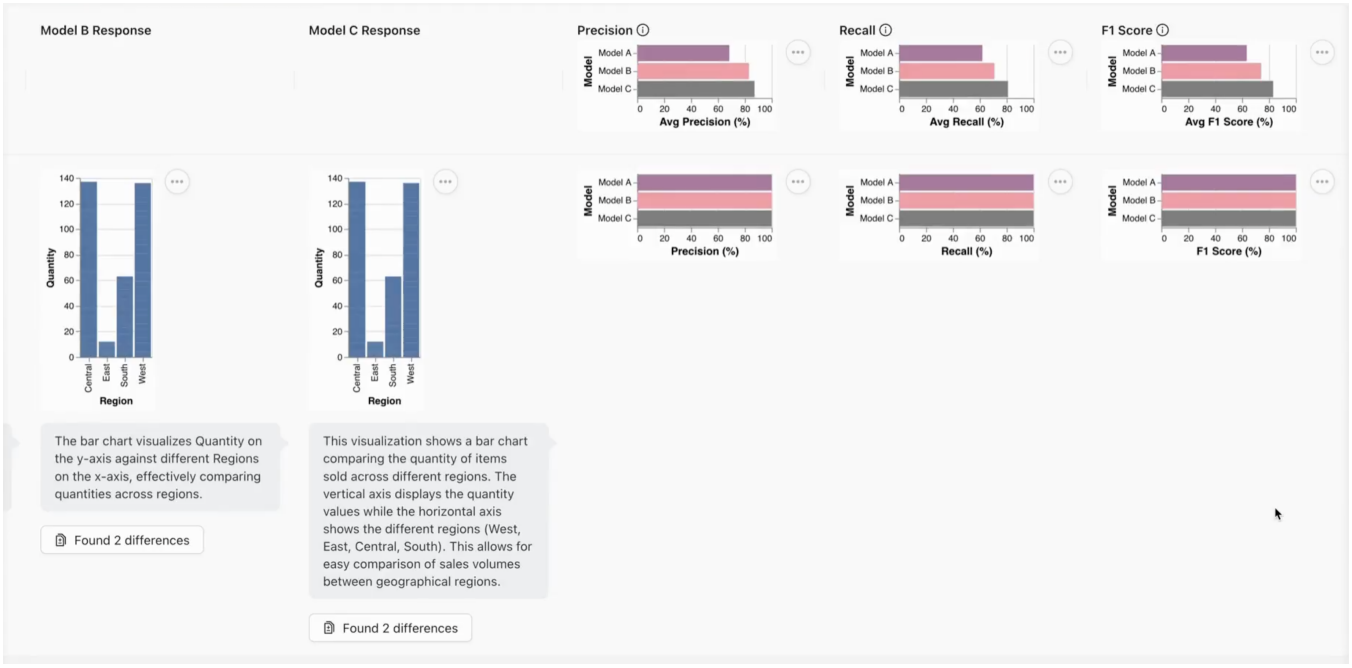
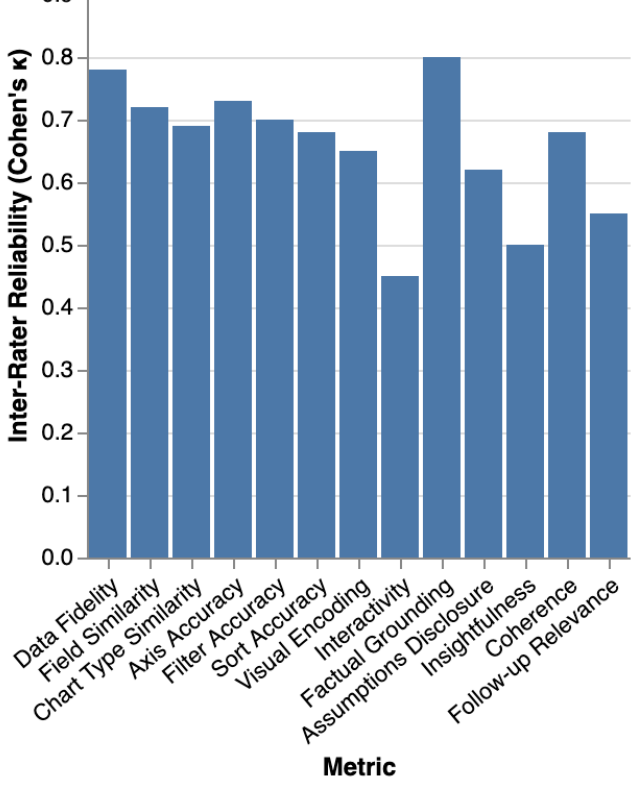
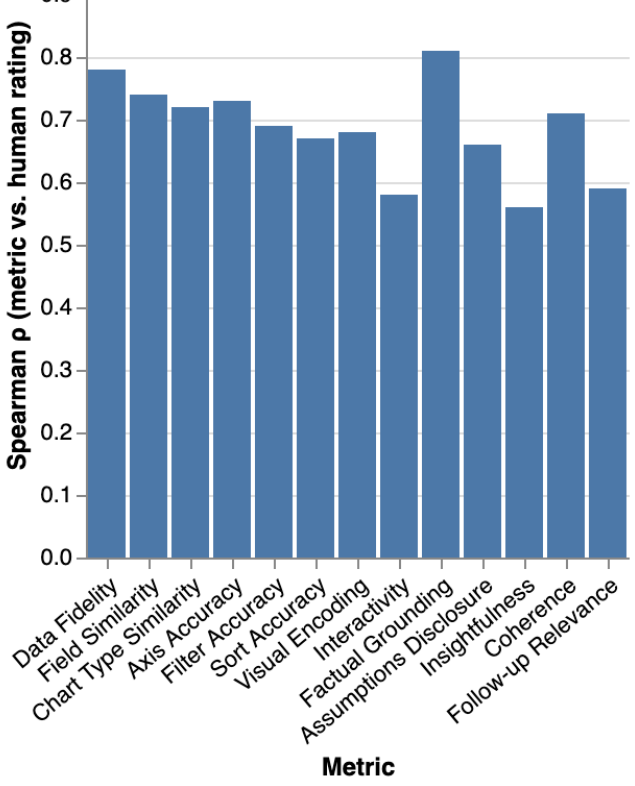


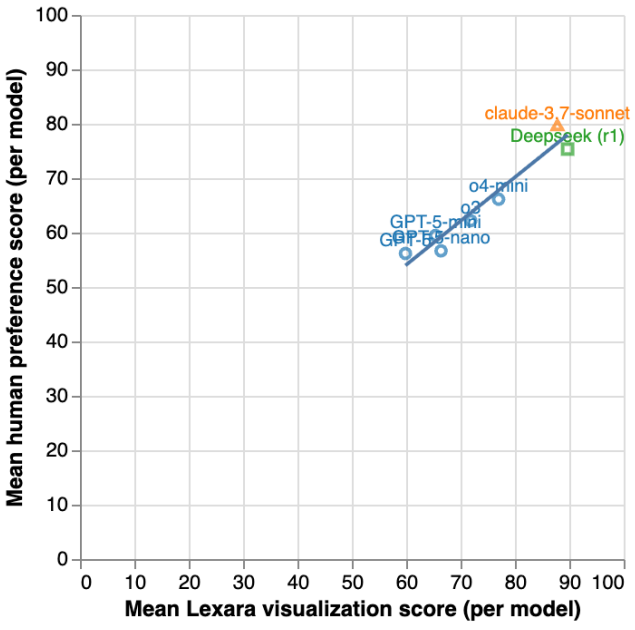
Figure 6: In the formative study, participants reviewed three anonymized model outputs presented in a static table, alongside traditional NLG metrics including F1, Precision, Recall for each response.



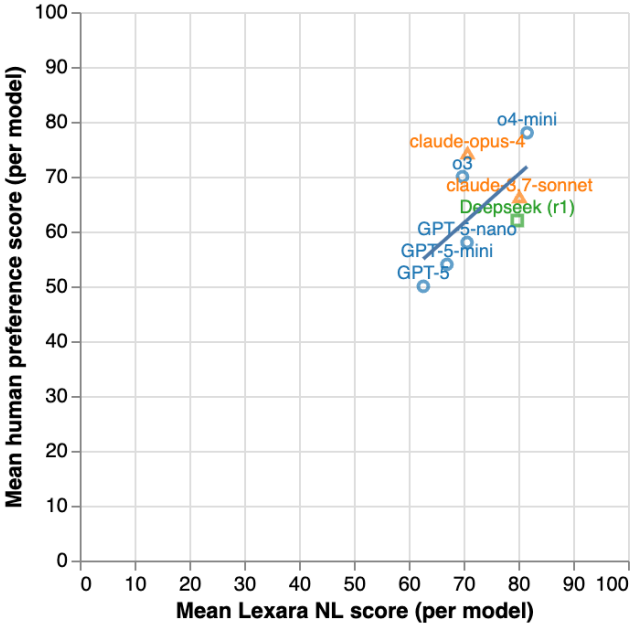
(a) Inter-rater reliability (Cohen's κ) for all metrics.



(b) Metric–human alignment (Spearman ρ) for all metrics.



(c) Model-level alignment for visualization scores.



(d) Model-level alignment for natural language/conversation scores.

Figure 7: Lexara metrics are both reliable and aligned with human judgments: (a–b) most metrics show κ and ρ above 0.6; (c–d) models with higher Lexara visualization and natural language scores are also preferred by humans.