A Concept for Integrating an LLM-Based Natural Language Interface for Visualizations Grammars

Adrian Jobst
adrian.jobst@hpi.de
Hasso Plattner Institute
Digital Engineering Faculty
University of Potsdam
Potsdam, Brandenburg, Germany

Daniel Atzberger
daniel.atzberger@hpi.de
Hasso Plattner Institute
Digital Engineering Faculty
University of Potsdam
Potsdam, Brandenburg, Germany

Mariia Tytarenko
m.Tytarenko@cgv.tugraz.at
Graz University of Technology
Graz, Styria, Austria

Willy Scheibel
willy.scheibel@hpi.de
Hasso Plattner Institute
Digital Engineering Faculty
University of Potsdam
Potsdam, Brandenburg, Germany

Jürgen Döllner
juergen.doellner@hpi.de
Hasso Plattner Institute
Digital Engineering Faculty
University of Potsdam
Potsdam, Brandenburg, Germany

Tobias Schreck
tobias.schreck@cgv.tugraz.at
Graz University of Technology
Graz, Styria, Austria

Abstract

In this paper, we propose a natural language interface visualization framework leveraging visualization grammar to balance the flexibility and stability of generated visualizations. Our system employs a JSON schema for visualization specification and an instruction prompt with semantically distinct sections for task context, visualizations, datasets, and control mechanisms. This design enables robust state management, live prompt adjustments, ensures clarity, consistency, and reusability in visualization generation.

CCS Concepts

• **Human-centered computing → Natural language interfaces**; **Information visualization**; *Visualization techniques*; Visualization toolkits.

Keywords

Natural-Language Interfaces, Visualization Generation, LLMs

ACM Reference Format:

Adrian Jobst, Daniel Atzberger, Mariia Tytarenko, Willy Scheibel, Jürgen Döllner, and Tobias Schreck. 2025. A Concept for Integrating an LLM-Based Natural Language Interface for Visualizations Grammars. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3701716.3717812

1 Context & Challenges

Large Language Models (LLMs) are being integrated increasingly into *Visual Analytics* (VA) applications and visualization systems [5]. *Natural Language Interfaces* (NLIs), i.e., systems that "interpret a user's natural language queries as input and output appropriate visualizations" that are backed by an LLM represent a distinct class of these applications [10]. In particular, NLIs provide a complementary interaction mode, allowing easier access to visualizations, especially for people with physical disabilities (e.g., color blindness or impaired vision) or limited prior visualization experience [3, 4].

Existing NLIs often generate source code for visualizations based on natural language input, leveraging dedicated libraries such as *D3.js* or *matplotlib* [9, 11]. While these approaches offer flexibility, the resulting visualizations can deviate from established design principles, produce invalid code, or suffer from low readability due to underspecification [2, 7, 7, 9, 12]. Complimentary approaches restrict LLMs to modifying only specific aspects of a visualization, thereby minimizing flexibility to ensure stability [1, 6, 8].

We propose a concept for an NLI that takes an in-between approach by leveraging visualization grammar, such as the *Grammar of Graphics*, to generate visualizations based on user specifications. Our methodology reduces flexibility by limiting the language model to predefined mappings within the grammar, while still allowing enough adaptability to accommodate a range of visualization needs.

2 Proposed Concept

Our system consists of two key components and is inspired by the concept presented by Jobst et al. [6]: (1) a JSON schema derived from a visualization grammar to specify visualizations, and (2) an instruction prompt that provides context and guidance to the LLM. The instruction prompt design is inspired by Choe et al. [3] but emphasizes semantic separation of concerns where feasible. The prompt includes several sections: (1) an **Overall Task Section** outlining the application's purpose and the LLM's role; (2) descriptions of visualizations used in the application within a **Visualizations Section**; (3) details about available datasets, such as attribute names from a dataset and additional metadata not directly inferable from the source within a **Dataset Section**; and (4) a **Control Section** specifying communication between the LLM and the user interface, including state information in textual form and a JSON schema.

The Visualizations and Datasets sections exemplify where separating concerns enhances clarity. For instance, the Visualizations section focuses solely on available visualizations, without detailing dataset-to-visual mapping. This separation, combined with our use

WWW Companion '25, April 28-May 2, 2025, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/2025/04
https://doi.org/10.1145/3701716.3717812

```
...

Control Section
To alter application behavior, e.g. visualization you can include control
information in json format at the end of your responses. Control information
should be included in the textual response you give in the Output section.
Include it in markdown style, so prepend "\`\`\`json". Here is a json schema
of your control options and possible input:
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "geom": {
      "type": "string",
      "description": "The used chart type",
      "default": "point" },
    "x": {
      "type": "string",
      "description": "Data variable mapped to horizontal axis" },
    "y": {
      "type": "string",
      "description": "Data variable mapped to vertical axis" },
    "selected": {
      "type": "array",
      "description": "Indices of highlighted observations",
      "items": { "type": "number" },
      "default": [] },
    "VLAT-score": {
      "type": "number",
      "description": "VLAT score of the user",
      "default": 4 }
  },
  "required": ["geom", "x", "y"],
  "additionalProperties": false
}
```

Figure 1: Control section of an instruction prompt. The JSON schema describes the LLM-to-application interaction. The chart configuration syntax is inspired by ggplot2.

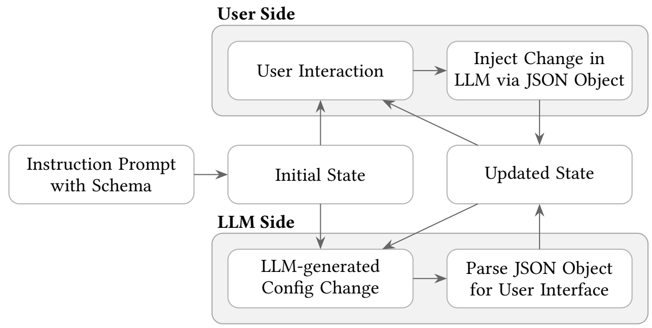

Figure 2: Possible state changes through user and LLM.

of grammar, ensures that even small specification changes can produce different visualizations. The Control Section serves as the core of the NLI, defining how the LLM interacts with the user interface while also establishing a feedback channel from the interface to the LLM. As illustrated in Figure 1, it includes a textual explanation and, crucially, a JSON schema. Using a schema instead of natural language or JSON examples offers significant advantages: it enables validation of generated responses while providing field descriptions and default values, ensuring consistency and clarity.

Figure 2 shows how the application state evolves. Initially, the LLM relies on the instruction prompt, making schema defaults and UI settings crucial. The LLM updates state by including a JSON object in its response, such as adding observation IDs to *selected* for highlighting. Likewise, the UI communicates updates by embedding changes in the same JSON format.

3 Outlook

By integrating visualization grammar and a semantically structured instruction prompt, our approach offers a promising solution to improve the clarity, consistency, and accessibility of LLM-assisted visualization systems. Using well-established JSON schemas and textual descriptions, the framework reduces the technical expertise required by developers to create reliable natural language interfaces, lowering the barrier to adoption. Furthermore, the modular design simplifies the evaluation and benchmarking of V-NLI approaches, which we will focus on in the future, along with testing the reusability of individual prompt sections.

Acknowledgements

This work is part of the "SPUR-OPT" project (grant 16KN113523) funded by the Federal Ministry for Economic Affairs and Climate Action of Germany. The work of Mariia Tytarenko was funded by the FWF as part of the project 'Human-Centered Interactive Adaptive Visual Approaches in High-Quality Health Information' (A$^+$CHIS; Grant No. FG 11-B).

References

[1] Alexander Bendeck and John Stasko. 2024. An Empirical Evaluation of the GPT-4 Multimodal Language Model on Visualization Literacy Tasks. *IEEE TVCG* (2024). https://doi.org/10.1109/TVCG.2024.3456155

[2] 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 TVCG* (2024), 1–11. https://doi.org/10.1109/TVCG.2024.3456320

[3] Kiroong Choe, Chaerin Lee, Soohyun Lee, Jiwon Song, Aeri Cho, Nam Wook Kim, and Jinwook Seo. 2024. Enhancing Data Literacy On-demand: LLMs as Guides for Novices in Chart Interpretation. *IEEE TVCG* (2024), 1–17. https://doi.org/10.1109/TVCG.2024.3413195

[4] Enamul Hoque. 2023. NLP4Vis: Natural Language Processing for Information Visualization – Half-day tutorial at IEEE VIS Conference 2023. URL: https://nlp4vis.github.io/IEEEVis-2023/index.html.

[5] Maeve Hutchinson, Radu Jianu, Aidan Slingsby, and Pranava Madhyastha. 2024. LLM-Assisted Visual Analytics: Opportunities and Challenges. In *Proc. CGVC '24*. EG. https://doi.org/10.2312/cgvc.20241237

[6] Adrian Jobst, Daniel Atzberger, Willy Scheibel, Jürgen Döllner, and Tobias Schreck. 2025. Delphi: A Natural Language Interface for 2.5D Treemap Visualization of Source Code. In *Proc. IVAPP*. SciTePress. in press.

[7] Tae Soo Kim, Yoonjoo Lee, Jamin Shin, Young-Ho Kim, and Juho Kim. 2024. EvalLM: Interactive Evaluation of Large Language Model Prompts on User-Defined Criteria. In *Proc. CHI*. ACM, Article 306, 21 pages. https://doi.org/10.1145/3613904.3642216

[8] Michael Xieyang Liu, Frederick Liu, Alexander J Fiannaca, Terry Koo, Lucas Dixon, Michael Terry, and Carrie J Cai. 2024. " We Need Structured Output": Towards User-centered Constraints on Large Language Model Output. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–9.

[9] 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. https://doi.org/10.1109/ACCESS.2023.3274199

[10] Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai, and Jianmin Wang. 2023. Towards Natural Language Interfaces for Data Visualization: A Survey. *IEEE TVCG* 29, 6 (2023), 3121–3144. https://doi.org/10.1109/TVCG.2022.3148007

[11] Pere-Pau Vázquez. 2024. Are LLMs ready for Visualization?. In *Proc. PacificVis*. IEEE, 343–352. https://doi.org/10.1109/PacificVis60374.2024.00049

[12] Huichen Will Wang, Mitchell Gordon, Leilani Battle, and Jeffrey Heer. 2024. DracoGPT: Extracting Visualization Design Preferences from Large Language Models. *IEEE TVCG* (2024), 1–11. https://doi.org/10.1109/TVCG.2024.3456350