A. Two phases of NL2VIS models

NL2VIS is built on NL modeling solutions, which can be divided into two kinds: traditional symbolic NL processing models and modern deep learning models.

Traditional NL mapping systems, such as Deepeye [1], are built upon symbolic methods. These systems use pre-defined rules and probabilistic grammar-based methods to parse NL. But these hand-crafted models often lack accuracy and flexibility, require more resources, and can be hard to use.

Recent studies, such as NL4DV [2], an open-source Python tool, although using symbolic NLP methods, relies on semantic parsers like Stanford CoreNLP [3] to improve accuracy. This toolkit allows users without prior experience to utilize NL2VIS.

The second type of NL2VS models is deep-learning end-to-end model. These models integrate language understanding, reasoning, and chart creation into one system.

ncNet [4], trained with the nvBench [5] dataset, uses transformer-based models. It processes natural language query and accepts an optional chart template to restrict the results. This method has shown good accuracy. However, these advanced systems rely on users explicitly deciding which types of charts to create.

B. Advantages of LLMs solution

Compared to previous methods, LLMs for visualizations offer a simpler and more solution through effective prompt engineering and code encapsulation. They keep accurate even when queries are underspecified or highly vague. Additionally, as LLM accuracy continues to evolve, their visualisation inference abilities will also grow.

The LLM solution provides capabilities for automated chart selection, good at choosing the correct charts and rendering them appropriately, better than earlier methods.

Considering LLM prompts can be designed to obtain only limited information from datasets, data security concerns are greatly fixed and privacy is maintained.

C. Study Limitations and Our Work

The author examined GPT models including GPT-3, CodeX, ChatGPT. Our study would compare the capabilities of a wider range of LLMs, including GPT-4o and GPT-4o mini and GPT-o1, YoloPandas [6] for NL2VIS tasks and contrasts the performances with prior NL4DV and ncNet studies in terms of accuracy and efficiency.

The author’s research included only six case studies for visualizations. Our study will include more datasets to examine the LLM-based NL2VIS system. We will use these datasets with a wider range of queries, and go beyond NVBench’s built-in seven chart types, achieving for a detailed comparison with previous studies.

References

1. Y. Luo, X. Qin, N. Tang, and G. Li. Deepeye: Towards automatic data visualization. In ICDE, pp. 101–112, 2018.

2. A. Narechania, A. Srinivasan, and J. Stasko, ‘‘NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries,’’ IEEE Trans. Vis. Comput. Graphics, vol. 27, no. 2, pp. 369–379,Feb. 2021.

3. C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, ‘‘The Stanford CoreNLP natural language processing toolkit,’’ in Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations, 2014, pp. 55–60.

4. Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin. Natural language to visualization by neural machine translation. IEEE Transactions on Visualization and Computer Graphics, pp. 1–1, 2021.

5. Y. Luo, J. Tang, and G. Li, ‘‘NvBench: A large-scale synthesized dataset for cross-domain natural language to visualization task,’’ 2021, arXiv:2112.12926.

6. YoloPandas Developers. (2023). YoloPandas. Python Package Index (PyPI). [Online]. Available: <https://pypi.org/project/yolopandas/>