Evaluating LLMs for Visualization Tasks

Saadiq Rauf Khan, Vinit Chandak and Sougata Mukherjea
Indian Institute of Technology, Delhi, India

Keywords: Large Language Models, Visualization Generation, Visualization Understanding.

Abstract: Information Visualization has been utilized to gain insights from complex data. In recent times, Large Language Models (LLMs) have performed very well in many tasks. In this paper, we showcase the capabilities of different popular LLMs to generate code for visualization based on simple prompts. We also analyze the power of LLMs to understand some common visualizations by answering simple questions. Our study shows that LLMs could generate code for some visualizations as well as answer questions about them. However, LLMs also have several limitations. We believe that our insights can be used to improve both LLMs and Information Visualization systems.

1 INTRODUCTION

With the amount and complexity of information increasing at staggering rates, Information Visualization is being utilized to enable people understand and analyze information. Over the years many techniques have been developed for creating information visualizations of different types of data. Information visualization can be created using various tools[1], libraries in many programming languages[2] as well as scripts[3]. However, the complexity of these tools, libraries and scripts can pose a barrier, especially for individuals without a strong background in data science or programming. To address this, automation of visualization creation using artificial intelligence techniques has also been explored (Wu et al., 2022).

Natural language interfaces allow users to generate visualizations using simple and intuitive commands. The integration of natural language processing in data visualization tools enhances the efficiency of data analysis. Analysts can now focus more on interpreting the data rather than the technicalities of creating visualizations. This advancement democratizes data analysis, making it more accessible to a broader audience.

Large Language Models (LLMs) like GPT-3 (Brown et al., 2020) are capable of completing text inputs to produce human-like results. They have revolutionized Natural Language Processing by achieving state-of-the-art results on various tasks. Similarly, deep learning models that are trained on a large amount of existing code and can generate new code given some forms of specifications such as natural language descriptions or incomplete code (Chen et al., 2021).

Another important task is the machine understanding of the visualizations. It accelerates data analysis by allowing machines to process and interpret large volumes of visual data quickly, reducing the time needed for manual interpretation. Moreover, it improves accuracy by providing consistent extraction of information from visualizations.

In this paper, we explore whether visualizations can be created or understood by prompting Large Language Models in natural language. Given the enormous potential of LLMs our aim was to explore whether LLMs are ready for Visualization tasks. Firstly, we evaluated whether popular LLMs like OpenAI's GPT-4[4], Google's Gemini[5] and Anthropic's Claude[6] could generate code for visualizations based on some simple prompts. Secondly, we investigated whether the LLMs could understand simple visualizations and answer questions about them. Our analysis shows that for some tasks LLMs performed very well; for example, most LLMs could produce code to generate simple visualizations. However, our study has also exposed several limitations of

[1] for example, Tableau: https://www.tableau.com

[2] for example, matplotlib: https://matplotlib.org/

[3] for example, VegaLite: https://vega.github.io/vega-lite/

[4] gpt4: https://https://openai.com/index/gpt-4

[5] Gemini: https://gemini.google.com

[6] Claude: https://www.anthropic.com/claude

Khan, S. R., Chandak, V. and Mukherjea, S.
Evaluating LLMs for Visualization Tasks.
DOI: 10.5220/0013079600003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 1: GRAPP, HUCAPP and IVAPP*, pages 791-798
ISBN: 978-989-758-728-3; ISSN: 2184-4321

the LLMs - they were incorrect in several tasks - both in generation and understanding.

The two main contributions of the paper are as follows:

1. We have done an analysis of the capabilities of some of the popular LLMs to generate Python code and Vega-lite scripts for visualizations based on prompts.
2. We explored the power of LLMs to understand simple visualizations and answer questions about them.

The remainder of the paper is organized as follows. Section 2 cites related work. Section 3 analyzes the LLMs for visualization generation, while Section 4 analyzes the LLMs for Visualization Understanding. Finally, Section 5 concludes the paper.

2 RELATED WORK

2.1 Large Language Models

Large Language Models like GPT-3 (Brown et al., 2020) have shown impressive results in various natural language understanding tasks. Given a suite of appropriate prompts[7] a single LLM can be used to solve a great number of tasks. Various prompt engineering techniques have been developed to find the most appropriate prompts to allow a LLM to solve the task at hand (Liu et al., 2023). On the other hand, Codex which is trained on 54 million software repositories on GitHub, has demonstrated stunning code generation capability — solving over 70% of 164 Python programming tasks with 100 samples (Chen et al., 2021).

2.2 Visualization Generation

With the popularity of information visualization, many techniques have been developed to create visualizations for different types of data. Information visualization can be created using various tools, libraries in many languages, as well as scripts based on Visualization Grammars.

AI techniques have also been explored to automate the creation of visualizations, for example, using decision trees (Wang et al., 2020) and sequence-to-sequence recurrent neural networks (Dibia and Demiralp, 2019). ChartSpark (Xiao et al., 2024) is a pictorial visualization authoring tool conditioned on both semantic contexts conveyed in textual inputs and data information embedded in plain charts.

One significant direction of research is automating the creation of data visualizations based on users' natural language queries. Many systems for using natural language to generate visualizations (NL2VIS) are based on libraries of natural language processing. For example, NL4DV (Narechania et al., 2021) uses CoreNLP (Manning et al., 2014). These systems either have constraints on user input or cannot understand complex natural language queries (Shen et al., 2023). Researchers have also trained neural networks using deep learning-based approaches (Luo et al., 2022) to process complex natural languages. However, a single approach based on deep learning cannot perform well on various tasks.

With the popularity of LLMs, there is significant interest in their application across various fields, including data visualization. (Vázquez, 2024) investigates the capabilities of ChatGPT in generating visualizations. This study systematically evaluates whether LLMs can correctly generate a wide variety of charts, effectively use different visualization libraries, and configure individual charts to specific requirements. The study concludes that while ChatGPT show promising capabilities in generating visualizations, there are still areas needing improvement.

Similarly, (Li et al., 2024) explores the capabilities of GPT-3.5, to generate visualizations in Vega-Lite from natural language descriptions using various prompting strategies. The key findings reveal that GPT-3.5 significantly outperforms previous state-of-the-art methods in the NL2VIS task. It demonstrates high accuracy in generating correct visualizations for simpler and more common chart types. However, the model struggles with more complex visualizations and tasks that require a deeper understanding of the data structures.

LLMs have been integrated into NL2VIS systems, such as Chat2Vis (Maddigan and Susnjak, 2023) and LIDA (Dibia, 2023), which generate Python code to construct data visualizations. However, there remains a need for a systematic evaluation of how well these LLMs can generate visualizations using different prompt strategies.

2.3 Visualization Understanding

In recent times various Multi-modal Large Language models (MMLLMs) have been proposed for understanding of charts. Examples include ChartAssistant (Meng et al., 2024) and UReader (Ye et al., 2023). Many datasets and benchmarks have also been introduced to test the capabilities of LLMs and MMM-

[7]A system prompt is a set of fixed instructions created by the developers to constrain the LLM's response

LLMs for chart understanding. Examples include ChartQA (Masry et al., 2022) and HallusionBench (Guan et al., 2024). Research has also been done to evaluate the Large Language models in different aspects of visualization understanding. For example, (Bendeck and Stasko, 2025) evaluates GPT-4 for various visualization literacy tasks, including answering questions and identifying deceptive visualizations. The assessment finds that GPT-4 can perform some tasks very efficiently, but struggles with some other tasks.

3 ANALYZING LLMs FOR VISUALIZATION GENERATION

3.1 Process

To evaluate the capabilities of LLMs in generating information visualizations, we followed a similar process as (Vázquez, 2024). We prompt the LLM to create a visualization based on a given specification and examine the code generated by the LLM. We chose Python for generating the visualization code due to its wide array of visualization libraries like *matplotlib*. We also examine the ability of the LLMs to generate *Vega-lite* scripts.

The methodology for the analysis involved several key steps:

1. Selection of Visualization Techniques: We selected 24 visualization techniques for tabular data. These include common charts like bar graphs and pie charts as well as charts that may not be that popular like Violin Plots and Locator Maps. We exclude visualization techniques for hierarchical and network representations.
2. Creation or Acquisition of Suitable Datasets: We created or sourced data sets that were appropriate for the chosen visualization techniques, ensuring that they provided a robust basis for testing. These data sets cover a wide range of data types, including categorical, quantitative, temporal, and geographical data. This enables a comprehensive evaluation of the LLMs' ability to generate accurate and varied visualizations.
3. Selection of LLMs to Analyze: We utilized 4 LLMs - OpenAI's GPT-3.5 and GPT-4o as well as Google's Gemini-1.5-pro and Anthropic's Claude 3 Opus for our analysis to provide a broad perspective on the capabilities of current popular models generalizable across different LLM designs.
4. Design and Fine-tuning of Prompts: We used zero-shot prompting[8] for this task. We carefully designed and refined the prompts to maximize the effectiveness and accuracy of the LLMs in generating the desired visualizations. An example prompt is: *Can you write a Python script that generates a Bubble chart using columns mpg (quantitative), disp (quantitative), and hp (quantitative) from the CSV file cars.csv?*
5. Testing: We conducted a thorough test to evaluate the performance of LLMs, examining the variety of charts they could generate.

3.1.1 Experimental Procedure

The assessment of the LLMs focused on the accuracy, efficiency, and versatility of the models in producing effective visual representations of data. For each experiment, we followed the following process to ensure consistency and accuracy:

1. Initialize a New Session: Begin each experiment by creating a fresh session. Given that LLM chat sessions utilize previous prompts as context, it was crucial to start with a new session for each experiment. This approach ensured that each test was conducted independently, preventing any carry-over effects from previous prompts. For example, if multiple prompts requested charts in Vega-lite, subsequent prompts without a specified library or language might default to Vega-lite.
2. Consistent Prompt Input: Enter all prompts within the same session and on the same day to maintain uniform conditions.
3. Execute and Analyze: Utilize the LLM output (either Python code or Vega-lite scripts) to create a visualization and analyze it.

3.2 Chart Generation Using Python

In the first analysis, each of the 4 LLMs was prompted to generate Python code for all the 24 distinct chart types. The performance of the LLMs are shown in Table 1. Each tick mark (✓) represents a correct generation and each cross mark (✗) represents an incorrect generation. GPT-4o came out to be the best performer with the ability to produce around 95% of the charts followed by GPT-3.5 being able to produce 79% of the charts. The performance of Gemini and Claude was similar to that of GPT 3.5.

[8]Zero-shot prompting is a machine learning technique that involves giving an AI model a task or question without providing any specific training or examples (Liu et al., 2023)

Table 1: Performance Comparison of LLMs in Chart Generation using Python.

Chart Type	GPT-3.5	GPT-4o	Gemini	Claude
Area Chart	✓	✓	✓	✓
Bar Chart	✓	✓	✓	✓
Box Plot	✓	✓	✓	✓
Bubble Chart	✓	✓	✓	✓
Bullet Chart	✗	✓	✗	✗
Choropleth	✓	✓	✓	✗
Column Chart	✓	✓	✓	✓
Donut Chart	✓	✓	✓	✓
Dot Plot	✗	✓	✓	✓
Graduated Symbol Map	✗	✓	✗	✗
Grouped Bar Chart	✓	✓	✓	✓
Grouped Column Chart	✓	✓	✓	✓
Line Chart	✓	✓	✓	✓
Locator Map	✓	✓	✓	✓
Pictogram Chart	✗	✗	✗	✗
Pie Chart	✓	✓	✓	✓
Pyramid Chart	✗	✓	✗	✗
Radar Chart	✓	✓	✗	✗
Range Plot	✓	✓	✗	✗
Scatter Plot	✓	✓	✓	✓
Stacked Bar Chart	✓	✓	✓	✓
Stacked Column Chart	✓	✓	✓	✓
Violin Plot	✓	✓	✓	✓
XY Heatmap Chart	✓	✓	✓	✓
Total	19(79%)	23(95%)	18(75%)	17(70%)

Note that correct generation means that the LLM could produce correct code for the visualization based on the requirement specified by the prompt. Since LLMs are known to produce inconsistent results, we tuned the LLM parameters so that randomness in the output is minimized. During the experiments each prompt is repeated three times and we accept the outputs only if they remain the same.

Most of the errors were due to the lack of knowledge of some LLMs on certain types of visualization, especially uncommon ones. For example, only GPT-4o was able to produce the correct bullet charts. GPT 3.5 produced a Pyramid chart instead, whereas Gemini's and Claude's outputs were erroneous. The comparison is shown in Figure 1.

3.3 Chart Generation via Vega-Lite Scripts

We also wanted to test and compare the performance of the aforementioned LLMs to generate Vega-lite scripts. Here we prompted the LLMs to generate Vega-lite scripts for all the 24 selected charts. For this evaluation, we used GPT-4o and Gemini for experimentation.

For the Vega-lite scripts, the results are shown in Table 2. Vega-Lite proved to be difficult for LLMs. Gemini was only able to create roughly 40% of the total charts. The performance of GPT-4o also reduced significantly when switching from Python to Vega-lite. For example, both GPT-4o and Gemini could not produce Violin charts as shown in Figure 2.

4 ANALYZING LLMs FOR VISUALIZATION UNDERSTANDING

4.1 Data Set

For analyzing the capabilities of LLMs for understanding visualizations, we have used the FigureQA dataset (Kahou et al., 2018). FigureQA consists of common charts accompanied by questions and answers concerning them. The corpus is synthetically generated on a large scale: its training set contains 100,000 images with 1.3 million questions. The corpus has five common visualizations for tabular data, namely, horizontal and vertical bar graphs, continu-

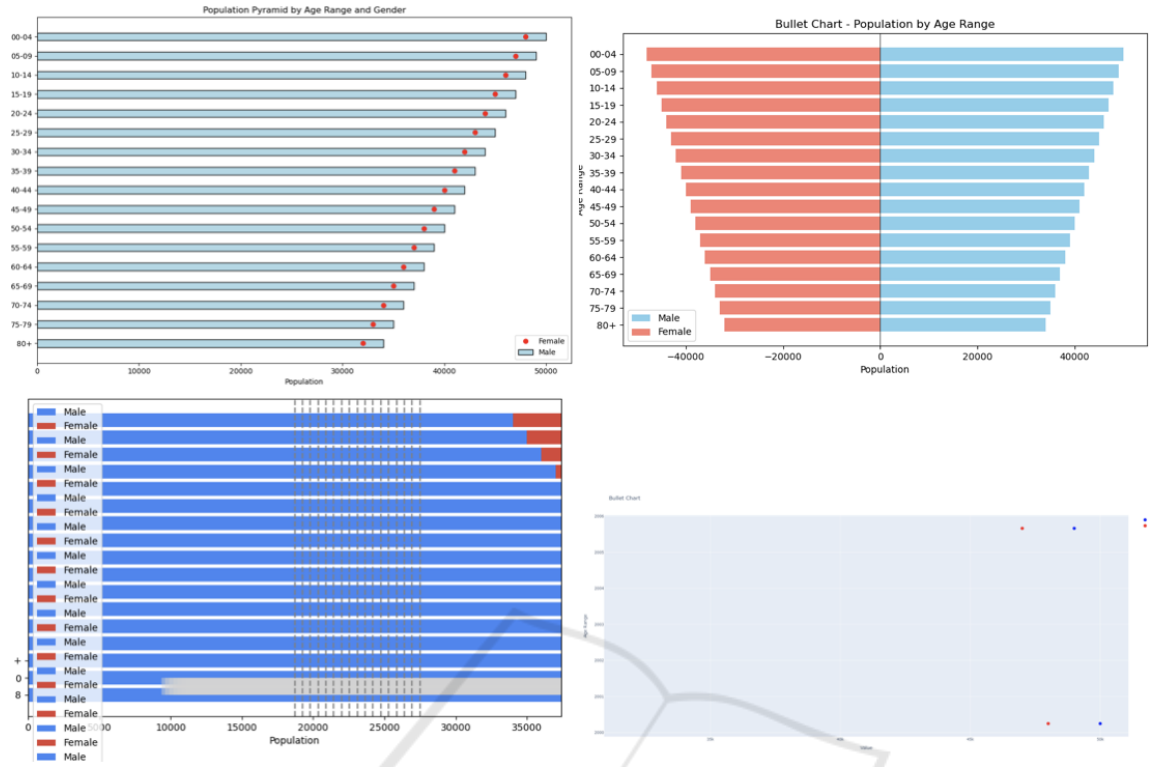

Figure 1: Comparison of Bullet charts. Only GPT-4o (top left) was able to produce the correct chart. GPT 3.5 produced a pyramid chart instead (top right). Gemini's and Claude's outputs were erroneous (bottom).

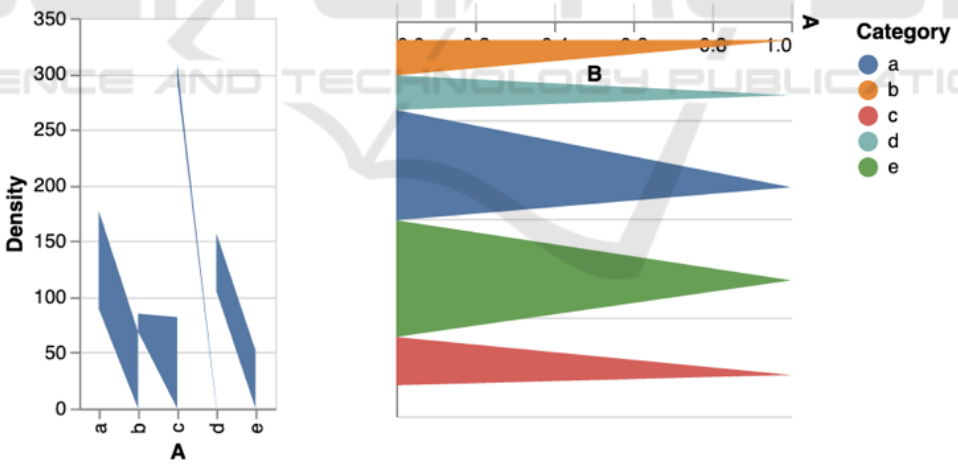

Figure 2: Both GPT-4o and Gemini could not produce Violin charts via Vega-lite scripts.

ous and discontinuous line charts, and pie charts.

There are 15 types of questions that compare the quantitative attributes of two plot elements or one plot element with all the others. In particular, the questions examine properties such as the maximum, minimum, median, roughness, and greater than/less than relationships. All are posed as a binary choice between yes and no.

4.2 Automated Analysis on FigureQA

To evaluate the ability of LLMs to understand and answer questions of information visualization we randomly chose 100 images from the data set and the corresponding 1,342 questions. Our random choice of the images will lead to variations in the chart types. We evaluated 3 LLMs - Google's Gemini-1.5-pro, OpenAI's GPT-4o and Anthropic's Claude 3 Opus.

Table 2: Performance Comparison of LLMs in Chart Generation using Vega-lite scripts.

Chart Type	GPT-4o	Gemini
Area Chart	✓	✓
Bar Chart	✓	✓
Box Plot	✓	✓
Bubble Chart	✓	✗
Bullet Chart	✗	✗
Choropleth	✓	✗
Column Chart	✓	✓
Donut Chart	✓	✓
Dot Plot	✓	✗
Graduated Symbol Map	✓	✗
Grouped Bar Chart	✗	✗
Grouped Column Chart	✗	✗
Line Chart	✓	✗
Locator Map	✓	✗
Pictogram Chart	✗	✗
Pie Chart	✓	✓
Pyramid Chart	✓	✗
Radar Chart	✗	✗
Range Plot	✗	✗
Scatter Plot	✓	✓
Stacked Bar Chart	✓	✓
Stacked Column Chart	✓	✓
Violin Plot	✗	✗
XY Heatmap Chart	✓	✓
Total	17(70%)	10(41%)

The results of the evaluation are shown in Table 3. As we can see, GPT-4o is the best performer, followed by Gemini-1.5-pro which is slightly behind and then Claude 3 Opus, which is much worse when compared to the other two models.

4.3 Need for Manual Analysis

While this initial automated test with the FigureQA dataset provided quantitative metrics for evaluating the performance of the selected LLMs, we know that relying solely on binary questions does not offer a comprehensive assessment of the model's true comprehension abilities. The binary nature of the FigureQA questions introduces a significant limitation: the susceptibility to random guessing. Models can achieve approximately 50% accuracy by making random choices without genuinely understanding the content of the figure/chart.

4.4 Data for Manual Analysis

To address this limitation, we moved beyond automated binary questioning and incorporated manual analysis as a crucial step in our methodology. This involved developing custom, non-binary questions aimed at probing deeper into the visual reasoning abilities of the models. For the manual analysis, we have selected 20 random charts for each of chart type in the FigureQA dataset. We have introduced new non-binary questions for each chart type that are useful to evaluate the level of understanding a model has of a chart. Examples of these questions are:

- **Vertical/Horizontal Bar Chart:** How many bars are there?
- **Line Chart:** How many dotted/non-dotted lines are there?
- **Pie Chart:** Which color pie has the largest area?

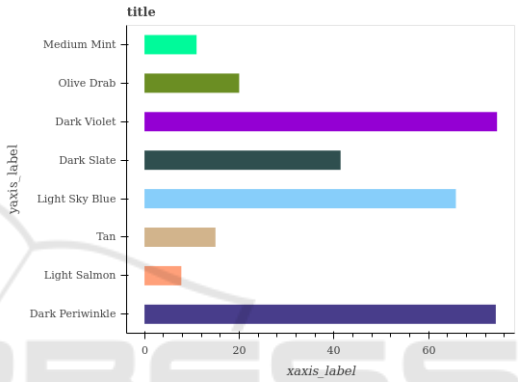

Figure 3: All three models could not determine the color of the longest bar since all of the models struggle in determining the larger/smaller bars.

4.5 Manual Analysis Results

For each LLM, for each of the chart, we uploaded the image and then asked it all the questions. All the answers were checked against the correct answers for the same questions. This, along with the inter-LLM comparison, will inherently compare the LLMs performance with a human baseline. We have also compared the models performance with and without a simple system prompt: *Analyze the following chart carefully and answer the following questions correctly.*

Table 4 shows the results. GPT-4o and Gemini performance were almost identical and better than Claude's performance. From the analysis we gained several key insights as follows:

- **Performance Across Different Chart Types**
  - The performance of LLMs varied significantly among different types of charts. For example, GPT-4o had 85% accuracy on pie charts and a mere 20% accuracy on line charts. This suggests that certain visualization formats may be easier for machines to interpret than others.

Table 3: Comparison of Performance Metrics between LLMs to answer Yes-No (binary) questions.

Metric	Gemini-1.5-pro	GPT-4o	Claude 3 Opus
Total Questions	1,342	1,342	1,342
Total Correct Answers	863	886	733
Total Wrong Answers	479	456	609
Accuracy (%)	64.31%	66.02%	54.61%

Table 4: Comparison of Performance Metrics between LLMs to answer non Yes-No (non-binary) questions.

	Gemini 1.5 Pro	**GPT-4o**	**Claude 3 Opus**
Images for which all qs answered correctly without prompt.	53.8%	51.3%	33.8%
Images for which all qs answered correctly with prompt.	63.8%	57.5%	38.8%
Qs answered without prompt.	84.8%	87%.8	70.5%
Qs answered correctly with prompt.	89.2%	89.4%	73.4%

  - Most of the models performed much better on pie charts as compared to other chart types.
  - All of the models performed very poorly on the line charts. This might be because of the presence of dotted lines, which might be treated as some kind of noise by the models. Sometimes Gemini-1.5-Pro did not recognize the dotted lines at all - especially when there is a mixture of dotted and non-dotted lines,
  - All the models struggled with identifying relationships between close boundaries and lengths of shapes. When the bar lengths are close on a bar graph, all of the models struggled in comparing them; An example is shown in Figure 3).
- **Impact of System Prompts**
  - In all the cases, the use of system prompts improved the performance of models. The improvement varied with the models and the chart types. Gemini-1.5-Pro improved significantly with the use of system prompts.
  - This shows the importance of context and guidance in improving the performance of LLMs.

5 CONCLUSION

In this paper, we explore the capabilities of LLMs in generating visualizations from natural language commands. We evaluated the performance of various prominent LLMs in creating different types of chart using Python and Vega-Lite scripts. In addition, we analyze the abilities of LLMs in understanding and answering questions about some charts. The paper extends the prior art to explore the capabilities of LLMs for visualization generation and understanding. The findings of our research provide valuable insight into the current state of LLMs in the field of data visualization. Our study shows that LLMs are very efficient in some tasks, but fail in some more complex tasks. The results of this paper can be used to address the limitations of LLMs and improve them in the future. Some areas of future work include the following.

- We want to explore whether more advanced prompting techniques like Chain-of-Thought (Wei et al., 2022) can improve the results.
- We need to expand the analysis to other types of information visualization such as graphs and trees.
- Combining the capabilities of LLMs and visualization tools to generate interactive visualizations is another promising research direction.

REFERENCES

Bendeck, A. and Stasko, J. (2025). An Empirical Evaluation of the GPT-4 Multimodal Language Model on Visualization Literacy Tasks. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1105–1115.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., et al. (2020). Language Models are Few-shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*, pages 1877–1901.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Ponde, H., et al.

(2021). Evaluating Large Language Models Trained on Code. *ArXiv*.

Dibia, V. (2023). LIDA: A Tool for Automatic Generation of Grammar agnostic Visualizations and Infographics using Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), 2023*, pages 113 – 126.

Dibia, V. and Demiralp, C. (2019). Data2Vis: Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks. *IEEE Computer Graphics and Applications*, 39(5):33–46.

Guan, T., Liu, F., Wu, X., Xian, R., Li, Z., et al. (2024). HallusionBench: An Advanced Diagnostic Suite for Entangled Language Hallucination and Visual Illusion in Large Vision-Language Models. In *Computer Vision and Pattern Recognition (CVPR 2024)*.

Kahou, S. E., Michalski, V., Atkinson, A., Kadar, A., Trischler, A., and Bengio, Y. (2018). FigureQA: An Annotated Figure Dataset for Visual Reasoning. *ArXiv*.

Li, G., Wang, X., Aodeng, G., Zheng, S., Zhang, Y., Ou, C., Wang, S., and Liu, C. H. (2024). Visualization Generation with Large Language Models: An Evaluation. *ArXiv*.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, 55(9):1–35.

Luo, Y., Tang, N., Li, G., Tang, J., Chai, C., and Qin, X. (2022). Natural Language to Visualization by Neural Machine Translation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):217–226.

Maddigan, P. and Susnjak, T. (2023). Chat2Vis: Generating Data visualizations via Natural Language using Chatgpt, Codex and GPT-3 Large Language Models. *IEEE Access*, 11.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (2014): System Demonstrations*, pages 55–60.

Masry, A., Do, X. L., Tan, J. Q., Joty, S., and Hoque, E. (2022). ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland.

Meng, F., Shao, W., Lu, Q., Gao, P., Zhang, K., Qiao, Y., and Luo, P. (2024). ChartAssisstant: A Universal Chart Multimodal Language Model via Chart-to-Table Pre-training and Multitask Instruction Tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Narechania, A., Srinivasan, A., and Stasko, J. T. (2021). NL4DV: A Toolkit for Generating Analytic Specifications for Data Visualization from Natural Language Queries. *IEEE Transactions on Visualization and Computer Graphics*, 27(2).

Shen, L., Shen, E., Luo, Y., Yang, X., Hu, X., Zhang, X., Tai, Z., and Wang, J. (2023). Towards Natural Language Interfaces for Data Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 29(6):3121–3144.

Vázquez, P.-P. (2024). Are LLMs ready for Visualization? In *IEEE PacificVis 2024 Workshop - Vis Meets AI*, pages 343–352.

Wang, Y., Sun, Z., Zhang, H., Cui, W., Xu, K., Ma, X., and Zhang, D. (2020). DataShot: Automatic Generation of Fact Sheets from Tabular Data. *IEEE Transactions on Visualization & Computer Graphics*, 26(1):895–905.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., hsin Chi, E. H., Xia, F., Le, Q., and Zhou, D. (2022). Chain of Thought Prompting Elicits Reasoning in Large Language Models. *Neural Information Processing Systems*.

Wu, A., Wang, Y., Shu, X., Moritz, D., Cui, W., Zhang, H., Zhang, D., and Qu, H. (2022). AI4VIS: Survey on Artificial Intelligence Approaches for Data Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):5049–5070.

Xiao, S., Huang, S., Lin, Y., Ye, Y., and Zeng, W. (2024). Let the Chart Spark: Embedding Semantic Context into Chart with Text-to-Image Generative Model. *IEEE Transactions on Visualization & Computer Graphics*, 30(1):284–294.

Ye, J., Hu, A., Xu, H., Ye, Q., Yan, M., et al. (2023). UReader: Universal OCR-free Visually-situated Language Understanding with Multimodal Large Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.