

# Comparing the Performance of LLM-based Data Visualization Methods in Chat2VIS

Dewei Tan

*Master of Science in Applied Computer Science  
Fairleigh Dickinson University  
Vancouver, Canada  
d.tan@student.fdu.edu  
ID: 2093688*

Zihuan Zhu

*Master of Science in Applied Computer Science  
Fairleigh Dickinson University  
Vancouver, Canada  
z.zhu2@student.fdu.edu  
ID: 2086903*

Jing Li

*Master of Science in Applied Computer Science  
Fairleigh Dickinson University  
Vancouver, Canada  
j.li8@student.fdu.edu  
ID: 2091325*

Xiangjun Li

*Master of Science in Applied Computer Science  
Fairleigh Dickinson University  
Vancouver, Canada  
x.li3@student.fdu.edu  
ID: 2092016*

Lina Gu

*Master of Science in Applied Computer Science  
Fairleigh Dickinson University  
Vancouver, Canada  
l.gu@student.fdu.edu  
ID: 2088991*

**Abstract**—Visualizations from vague human language has long been challenges for natural language processing (NLP). During Natural Language Text-Generated Data Visualization (NL2VIS) system development, there are growing needs for non-experts to interpret and visualize data accurately and informatively. Our project aims to offer solutions by extending Chat2VIS study. We would incorporate a broader range of LLMs such as ChatGPT (4o Mini, 4o, o1), YoloPandas, offering a detailed comparison with prior NL4DV and ncNet systems. Through an increased number of datasets, comprehensive case studies and ambiguous queries with no specified chart types, we examine and compare the performance differences of these models, contrasting their accuracy and efficiency, paving the way for further studies.

**Index Terms**—large language models, visualizations from natural language, ChatGPT, natural language interfaces.

## I. INTRODUCTION

### A. Problem Statement

Natural language processing (NLP) has long been dealing with the complexities of human language, particularly in the domains of natural language understanding (NLU) [1]. One important application of NLP is Natural Language to Data Visualization (NL2VIS), where natural language queries are transformed into meaningful data visualizations. Data visualization researches are putting great efforts to find new approaches to realize easy and direct conversion, but there are gaps in this domain and ongoing challenges.

### B. Research Importance

NL2VIS researchers image the scenario where users ask question like "compare prices of the product in different business" and obtain a bar chart displaying the difference. Such implement is particularly beneficial for users who lack expertise in graphical data analysis but need quick and accurate visual representations of complex datasets. Workable Natural Language Interfaces (NLIs) not only reach a wider range of users but also allow beginners to interact with complex data in a natural and user-friendly way.

### C. Research Challenges

The ability of machines to accurately comprehend and generate language poses significant challenges, owing to the inherent ambiguity, variability, and context-dependence of human communication. Machines are also required to know user's intent even with mistakenly written questions. Most current NL2VIS systems rely on statistical parsers, limiting them to handling only simple queries. Although advanced deep learning models have their place in other NLP tasks, they are not widely applied to NL2VIS, primarily due to the lack of large-scale labeled data [2] and computational costs that are essential for supporting these systems. This gap creates great difficulties in developing more flexible and powerful NL2VIS systems.

#### D. New solution

The advent of large language models like ChatGPT by OpenAI has marked a significant milestone in contextual visualization. These models demonstrated remarkable capabilities in NLP tasks including conversational agents, text generation, and summarization [3]. They excel in terms of automation, flexibility, and contextual understanding. The introduction of LLMs into NL2VIS enhances complex natural language queries processing and expands the flexibility of visual outputs.

#### E. Project Contribution

On the basis of the Chat2VIS platform presented by Maddigan et al. [4], our project adopts ChatGPT (4o mini, 4o, o1), NL4DV, ncNet, and YoloPandas as NL2VIS system candidates. We evaluate and compares the performance differences between these models using an expanded collection of datasets and complex queries not confined to any chart types, providing insights for more sophisticated visualizations systems.

### II. RELATED WORK

#### A. Existing NL2VIS Solutions

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 [5], 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 open-source NL4DV [6], although using symbolic NLP methods, relies on semantic parsers like Stanford CoreNLP [7] 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 [8], trained with the nvBench [2] 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 New LLMs Solution

In recent years, researchers have begun to analyze the performance of ChatGPT-3 and ChatGPT-4 in generating charts [9]. 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 Improvement

The authors in Chat2VIS study [4] examined GPT models including GPT-3, CodeX, ChatGPT. Our project would compare the capabilities of a wider range of LLMs, including GPT-4o mini, GPT-4o and GPT-o1, YoloPandas [10] for NL2VIS tasks and contrasts the performances with prior NL4DV and ncNet studies in terms of accuracy and efficiency.

The authors' research included only six case studies for visualizations. Our project would include more datasets to examine the LLM-based NL2VIS system. We would analyze these datasets with more sophisticated queries, and go beyond NVBench's built-in seven chart types, achieving for a detailed comparison with previous studies.

### III. DESIGN AND IMPLEMENTATION

#### A. Solution Strategy

To use LLMs to visualize data by natural language query, the whole project is divided into several steps. For individual LLM, data processing methods may vary.

This project would realize visualizations in two ways. The first one is LLMs such as ChatGPT, which are not specifically developed for NL2VIS. Hence, extra prompt engineering should be done to restrict their output.

The second approach uses data visualization tools like NL4DV, nvBench(based on ncNet). Certain data processing modules may have been integrated inside these tools [2], [6]. A more feasible way to deal with data is to use built-in functions.

Generally, the project's workflow is divided into the following steps:

1) *Dataset selection*: A dataset is a basic premise in the program. In this project, users can either use example dataset or upload their own datasets to visualize data.

By default, the project contains some default example datasets including Movies, Housing, Cars and more. User can simply choose them by clicking corresponding radio buttons. These datasets come from the original paper.

When using customized dataset, CSV files could be uploaded to the project by clicking the "upload" button. After fully uploaded, users can see their data on the webpage.

2) *Dataset preprocessing*: Pandas is imported to deal with CSV files. At this step, every column's title and its data format is been recorded. This information will become part of the prompt sent to LLM to get the right code.

3) *Chart formatting*: Some formatting and drawing instructions will be added. In this step, required information including drawing library and python version; chart style information such as chart title and X, Y coordinate style will be added to the prompt. Wo that LLM can master more details and output the correct code.

4) *User needs elicitation*: Read users' input of what LLM they want to use, and what query they want to make. Integrated these information into prompts to different LLM.

5) *LLM integration*: According to users' choice, link LLMs by the API key they inputted. Send prompts to each LLM and get the answer code.

6) *Charts creation*: Draw charts based on the answer code from different LLM. Arrange and display these charts on the webpage.

## B. Product Description

In this program, a web based app will be made to carry these functions. The web application contains the following parts:

1) *Dataset viewer and selection*: Users can use this module to choose different datasets, preview the data rows. Moreover, users can use this module to upload their own datasets.

2) *LLM setting and selection*: Users use this module to control LLM configuration, such as turn on or off some LLMs, and manage API keys.

3) *Query input*: Users utilize this module to communicate with LLMs using natural language to describe the chart they want.

4) *Chart viewer*: This module shows the data visualization charts output from different LLMs.

## C. Algorithms

No new algorithms are proposed in this project. Opaque algorithms work in black-box LLMs.

## D. Implementation

### 1) Modules:

a) *Front end*: A webpage will be made using streamlit framework. Most functions will be arranged in a single page to keep the minimalist user experience.

On the left, there's a sidebar to hold dataset selection and LLM setting part. Data rows in dataset will shown in a new tab based on users' needs.

On the right, the upper is the query input part. And the lower part is used for show charts.

b) *Prompt engineering*: String manipulation works will be done for better visualization. For data preprocessing, Pandas is used to extract data from CSV files.

c) *Data visualization*: Matplotlib is used to create charts. The script that generates the image comes from the output of the LLMs.

### 2) Weekly Deliverables:

#### a) Week 3: Project discussion and task division:

- Read related paper and library document, understand article logics.
- Assign modules and tasks to each team members.

#### b) Week 4: Preparation and Test:

- Test LLM APIs.
- Test related libraries.

#### c) Week 5: Front-end and CSV codes:

- Apply LLM APIs ready.
- Finish webpage front-end codes.
- Achieve CSV reading function.

#### d) Week 6: Prompt engineering:

- Determine the prompt engineering rules.
- Test prompts for different LLMs.
- Examine the chart code from LLMs.

#### e) Week 7: Drawing adjustment:

- Adjust drawing format.
- Adjust prompt format accordingly.

#### f) Week 8: Use case test:

- Test different use case.
- Fix bugs.

#### g) Week 9: Data replication:

- Reproduce the article results.

#### h) Week 10: Presentation:

- Organize data and documents.
- Prepare presentation.

## E. Methodology

- Languages: Python
- Libraries: pandas, numpy, streamlit, matplotlib
- LLMs: ChatGPT (4o mini, 4o, o1); NL4DV; ncNet and YoloPandas
- APIs: LLM APIs for natural language processing

## REFERENCES

- [1] G. Yenduri, M. Ramalingam, G. C. Selvi, Y. Supriya, G. Srivastava, P. K. R. Maddikunta, G. D. Raj, R. H. Jhaveri, B. Prabadevi, W. Wang, A. V. Vasilakos, and T. R. Gadekallu, "Gpt (generative pre-trained transformer)— a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions," *IEEE Access*, vol. 12, pp. 54608–54649, 2024.
- [2] Y. Luo, J. Tang, and G. Li, "nvbench: A large-scale synthesized dataset for cross-domain natural language to visualization task," 2021.
- [3] P. P. Ray, "Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 121–154, 2023.
- [4] P. Maddigan and T. Susnjak, "Chat2vis: Generating data visualizations via natural language using chatgpt, codex and gpt-3 large language models," *IEEE Access*, vol. 11, pp. 45181–45193, 2023.
- [5] Y. Luo, X. Qin, N. Tang, and G. Li, "Deepeye: Towards automatic data visualization," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 101–112, 2018.
- [6] A. Narechania, A. Srinivasan, and J. Stasko, "Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 369–379, 2021.
- [7] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (K. Bontcheva and J. Zhu, eds.), (Baltimore, Maryland), pp. 55–60, Association for Computational Linguistics, 2014.
- [8] 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*, vol. 28, no. 1, pp. 217–226, 2022.
- [9] P.-P. Vázquez, "Are llms ready for visualization?," in *2024 IEEE 17th Pacific Visualization Conference (PacificVis)*, pp. 343–352, 2024.
- [10] Y. Developers, "Yolopandas: Interact with pandas objects via llms and langchain."