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

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

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

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

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

***Abstract*—This is the abstract.**

***Index Terms*—component, formatting, style, styling, insert.**

I. Introduction

A. Problem Statement

B. Importance

C. Challenges

D. Solution

E. Contribution

II. Related Work

A. Existing Solutions

B. Comparison

III. Design and Implementation

A. Solution Strategy

In order to use large language models to translate natural language into data visualization charts, the whole project is divided into several steps. To each LLM, Data processing methods may vary.

This project utilized two kinds of LLM tools. One is general-purpose LLMs such as ChatGPT. This type of LLMs do not have a specific output format; so extra prompt engineering steps should be taken to restrict their output format.

The other kind is LLM based data visualization tools like NL4DV, nvBench(based on ncNET) and YoloPandas. Some data processing modules may have been integrated inside these tools. [2]–[4] A more feasible way is to use their built-in functions to deal with data. In this way,a more real performance differences will be shown.

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

1) Choose or upload a dataset: 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 CSV example datasets such as Movies, Housing, Cars and so on. User can simply choose them by clicking corresponding radio buttons. These datasets come from the original paper.

When using customized dataset, the .CSV file have to uploaded to the project by clicking the ”upload” button. After fully uploaded, users can see their data on the webpage.

2) Data set 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 format prompts: Some formatting and drawing instructions will be added. In this step, Basic information such as 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) Get user's needs: 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) Link LLM: According to users' choice, link LLMs by the API key they inputted. Send prompts to each LLM and get the answer code.

6) Draw the charts: 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 all functions. The web app 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 specific algorithms in this project. All the algorithms are used in LLMs, which act as a black box to this project.

D. Implementation

1) Modules:

a) Front end: A simple 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) Propmt engineering: Many string manipulation works will be done in this part. We'll use python as coding language. As 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 familiarization and task division:

- Read related paper and library document, get familiar with the logic.
- Divide modules developing tasks to each team members.

b) Week 4: Preparation and Test:

- Test LLM apis
- Test ralated libararies.

c) Week 5: Front-end and CSV reading:

- Finish webpage front-end coding work.
- Finish the CSV reading process.

d) Week 6: Prompt engineering:

- Determine the prompt engineering rules.
- Complete the process of prompt splicing.
- obtaining the drawing code from LLM.

e) Week 7: Drawing adjustment:

- Adjust prompt format.
- Adjust drawing format.

f) Week 8: Use case test:

- Test with different use case.
- fix bugs.

g) Week 9: Use case test:

- Test with more use case.

h) Week 10: Presentation:

- Organize data and documents.
- Prepare presentation.

E. Methodology

- Languages: Python
- Libraries: Pandas, Streamlit, Matplotlib
- LLMs: ChatGPT 4o mini, 4o, o1; NL4DV; ncNET and YOLOPandas
- APIs: LLM APIs for natural language processing

References

[1] P. Maddigan and T. Susnjak, ”Chat2VIS: Generating Data Visualizations via Natural Language Using ChatGPT, Codex and GPT-3 Large Language Models,” in IEEE Access, vol. 11, pp. 45181-45193, 2023, doi: 10.1109/ACCESS.2023.3274199.

[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] YoloPandas. Python Package Index (PyPI), 2023, [online] Available: https://pypi.org/project/yolopandas/.

[4] Luo, Yuyu, Jiawei Tang, and Guoliang Li. ”nvBench: A large-scale synthesized dataset for cross-domain natural language to visualization task.” arXiv preprint arXiv:2112.12926 (2021).