Creating a Real-world Benchmark for Text-to-Vis

Hy Nguyen[1*], **Andrew Reeson**[2], **Cécile Paris**[2], **Josiah Poon**[1],
and **Jonathan K. Kummerfeld**[1]
The University of Sydney[1] CSIRO's Data61[2]
nngu0448@uni.sydney.edu.au*

Abstract

Generating visualisations from natural language requests (Text-to-Vis) is highly valuable as it allows users to create data plots quickly and easily, even if they lack programming skills. While recent LLMs can perform this task, there is a lack of high-quality benchmarks that evaluate these models in real-world uses. This paper introduces a new benchmark dataset drawn from real-world scenarios and a novel annotation framework that empowers novices to handle annotation tasks typically reserved for programming and data analysis experts. We believe this benchmark could be a good challenge for LLMs and would drive the development of Text-to-Vis to better meet practical needs.

1 Problem

Text-to-Vis refers to the task of taking a human language request for visualisation along with input data and generating code to produce the corresponding visualisation, as described in Figure 1. Such a system would streamline and enhance data analysis so that non-expert users can easily express their visualisation requirements and gain valuable insights from their data.

Despite its potential, the Text-to-Vis field remains in its early stages and faces several challenges, especially the scarcity of high-quality benchmarks. Although there are several, including nvBench (Luo et al., 2021), nlvUtterances (Srinivasan et al., 2021), PlotCoder (Chen et al., 2021), and ChartDialogs (Shao and Nakashole, 2020), most of them are either synthesised, automatically extracted, or influenced by a limited set of annotators. Our prior work identified a significant gap between these benchmarks and real-world visualisations, as evaluations do not test the same distribution of chart types, attributes, and the number of actions (Nguyen et al., 2024).

In this work, we aim to create a new benchmark for Text-to-Vis that accurately reflects real-world practices. To achieve this, we plan to collect annotation data from various sources, including data statistics websites, visualisation forums, and academic papers [1]. As illustrated with corresponding colors in Figure 1, the collected data include (1) plotting data, (2) natural language requests, (3) code snippets, and (4) visualisation images. We will also create a new annotation framework for novices to create code. Note that this task is often reserved for data visualisation and programming experts.

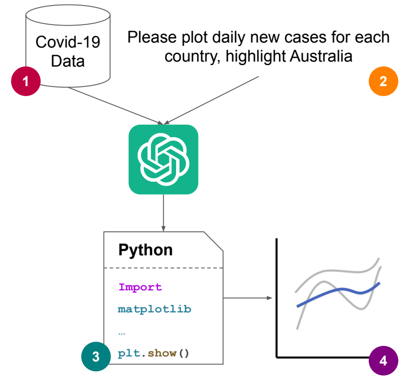
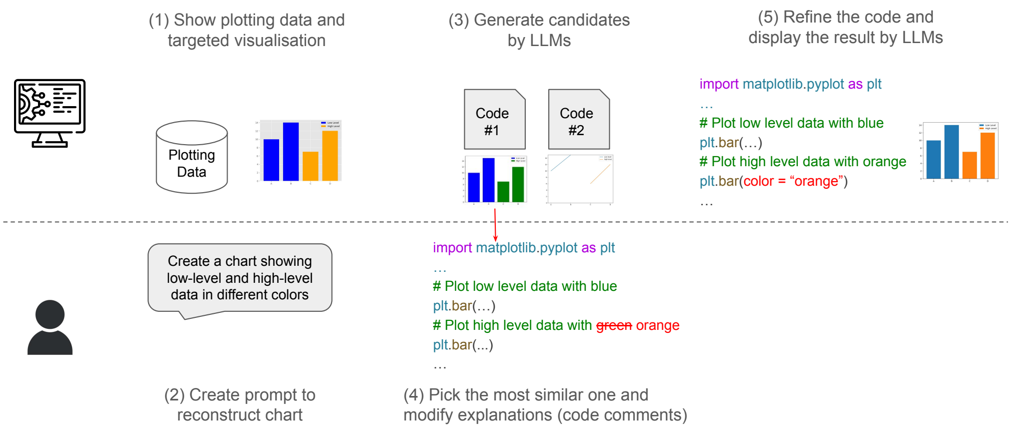

Covid-19 Data
Please plot daily new cases for each country, highlight Australia
1
2
Python
Import
matplotlib
...
plt.show()
3
4

Figure 1: A visual description of Text-to-Vis

2 Proposed Solution

Data Sources Plotting data and visualisations from statistics websites and visualisation forums can be gathered with clear copyright information. However, obtaining these from the academic domain can be challenging because authors often include pixel-based visualisations (PNG, JPG) in their publications rather than providing raw plotting data and code snippets. To overcome this, we plan to email authors to request the necessary data for our annotation. Although this method requires significant resources and ethics approval, it offers a

[1] https://ourworldindata.org, https://www.statista.com, https://chart-studio.plotly.com, https://arxiv.org/


(1) Show plotting data and targeted visualisation
Plotting Data
(3) Generate candidates by LLMs
Code #1
Code #2
(5) Refine the code and display the result by LLMs
(2) Create prompt to reconstruct chart
(4) Pick the most similar one and modify explanations (code comments)

```
import matplotlib.pyplot as plt
...
# Plot low level data with blue
plt.bar(...)
# Plot high level data with orange
plt.bar(color = "orange")
...
```

Create a chart showing low-level and high-level data in different colors

```
import matplotlib.pyplot as plt
...
# Plot low level data with blue
plt.bar(...)
# Plot high level data with ~~green~~ orange
plt.bar(...)
...
```

Figure 2: Annotation framework

direct approach to obtaining authentic data for our study.

Annotation Framework Drawing inspiration from Tian et al. (2023) in Text-to-SQL, we will develop a framework for indirect annotation in Text-to-Vis, explicitly designed for novice annotators. The framework contains five main steps, as shown in Figure 2. First, plotting data and a desired visualisation image are displayed on the labelling tool's interface. Second, the annotator then writes a natural language request to re-create the visualisation for given plotting data. Third, the system uses LLMs to generate several candidate pairs of code snippets and their resulting visualisations (produced by executing the code). Fourth, the annotator selects the most accurate pair and refines the associated code comments. Fifth, the LLMs modify the selected program based on the annotator's input, producing refined code and corresponding visualisation. This process repeats through the last four steps, iterating until the recreated visualisation closely matches the original. In cases where non-expert workers cannot complete the task, those instances will be escalated to experts for resolution.

Experiments and Analysis We will establish baselines for the new dataset using state-of-the-art LLMs like GPT-4 and Gemini. We will evaluate metrics such as plot type accuracy, data matching accuracy, and code-based similarity (CodeBLEU, CodeBERTScore). To assess the dataset's quality, we will manually review a sample to verify whether the recreated visualisations closely match the originals. We will compare task completion rates between novice and expert annotators to evaluate the effectiveness of the annotation framework.

3 Conclusion

This paper outlines several limitations in current Text-to-Vis benchmarks and introduces the development of a new dataset. The process involves gathering real-world visualisation data from diverse sources and creating an annotation framework that enables non-experts to handle tasks typically performed by experts. We believe this new benchmark will advance the development of Text-to-Vis to meet users' real-world visualisation needs.

References

Xinyun Chen, Linyuan Gong, Alvin Cheung, and Dawn Song. 2021. PlotCoder: Hierarchical decoding for synthesizing visualization code in programmatic context. In *Proceedings of ACL-IJCNLP*.

Yuyu Luo, Nan Tang, Guoliang Li, Chengliang Chai, Wenbo Li, and Xuedi Qin. 2021. Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks. In *Proceedings of SIGMOD/PODS*.

Hy Nguyen, Xuefei He, Andrew Reeson, Cecile Paris, Josiah Poon, and Jonathan K. Kummerfeld. 2024. Do text-to-vis benchmarks test real use of visualisations? In *Proceedings of EMNLP*.

Yutong Shao and Ndapandula Nakashole. 2020. Chartdialogs: Plotting from natural language instructions. In *Proceedings of ACL*.

Arjun Srinivasan, Nikhila Nyapathy, Bongshin Lee, Steven M Drucker, and John Stasko. 2021. Collecting and characterizing natural language utterances for specifying data visualizations. In *Proceedings of CHI*.

Yuan Tian, Zheng Zhang, Zheng Ning, Toby Li, Jonathan K. Kummerfeld, and Tianyi Zhang. 2023. Interactive text-to-SQL generation via editable step-by-step explanations. In *Proceedings of EMNLP*.