InterViz: A Large-Scale Conversational Benchmark Towards Interactive Natural Language Interfaces to Data Visualization

Anonymous ACL submission

Abstract

NL2Vis, the task of translating natural language (NL) queries into corresponding data visualizations (Vis), has gained significant attention in recent years due to its potential to enable non-experts to create visualizations easily. However, we have identified three limitations in existing NL2Vis datasets: (1) lack of contextual dependencies, (2) narrow range of programming code types, and (3) limited diversity of visualization charts. These limitations hinder the development and evaluation of parsing models for this task. To address these issues, we present InterViz, a large-scale, pragmatic dataset designed to interactively generate data visualization code from natural language queries within a conversational context. The dataset comprises 6,656 coherent question sequences, a total of 20,661 NL2Vis pairs, from 160 databases across 138 domains, covering numerous data visualization and programming languages. To collect a more diverse range of natural queries for visualization, we leverage large language models (LLMs) to simulate human behaviors in creating natural questions, ensuring the coherence and cohesion of conversations with in-context learning. We evaluated several strong baselines, including fine-tuning-based models and in-context-learning-based models, on this benchmark. Our findings show that InterViz presents substantial challenges for future research in NL2Vis.

1 Introduction

Natural Language Interfaces to Data Visualization (NLIDV) are critical in making complex data analytics and visualizations accessible to a broad range of users, including those without a technical background or programming skills (Shen et al., 2023; Qin et al., 2019; Tang et al., 2019). Building an efficient and robust

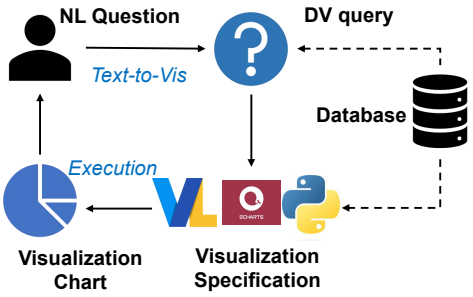

Figure 1: Workflow of the Natural Language to Data Visualization Framework

NLIDV system introduces the Text-to-Vis problem, a complex challenge that involves accurately interpreting the user's natural language queries and generating the corresponding data visualization code (Luo et al., 2018, 2021b). Figure 1 depicts the workflow of the natural language to data visualization framework. The process starts with the user's natural language query, where the parsing model converts into functional representations referred to as Data Visualization queries (DV queries). These DV queries, often in Visualization Query Language (VQL) format, contain instructions for data retrieval and visualization. After interacting with the database and retrieving the required data, the DV query is translated into visualization specifications (e.g., Vega-Lite or Matplotlib), guiding the creation of informative charts. The Text-to-Vis problem (Cui et al., 2020) encompasses a variety of issues, such as handling diverse visualization types, interpreting complex queries, and understanding contextual dependencies within a conversational interface. The development of a reliable Text-to-Vis system requires not only advanced Natural Language Processing techniques but also high-quality, context-dependent, and diverse datasets to train and evaluate these models.

Several datasets and benchmarks have been

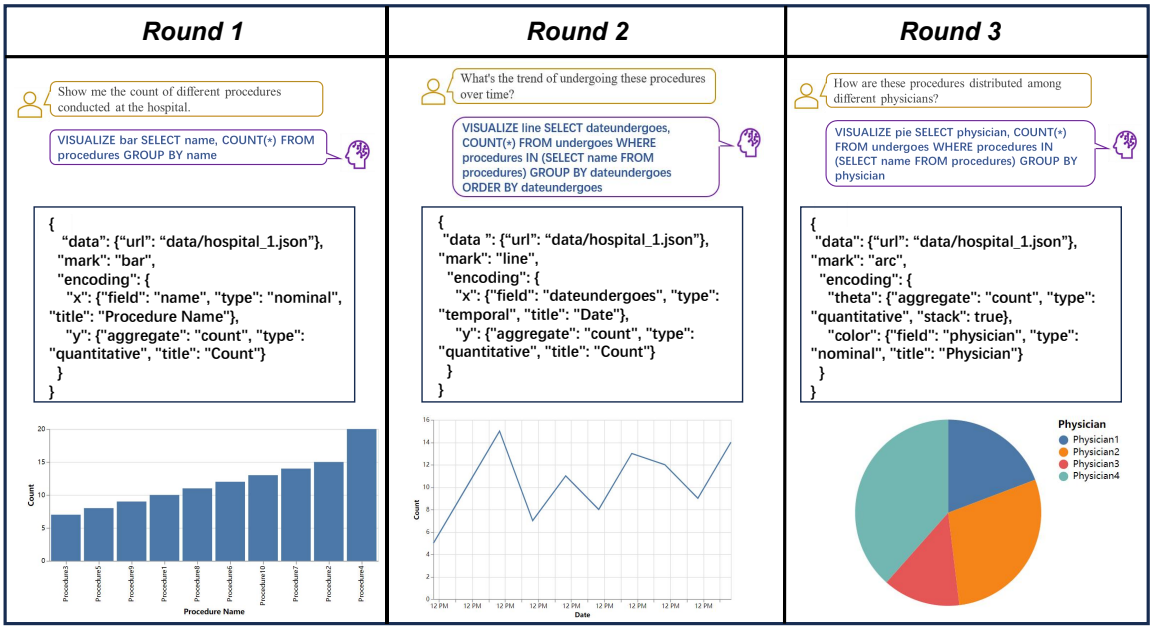

Figure 2: An Example of Multi-Round Conversational Interaction in InterViz Dataset

proposed for the Text-to-Vis problem, facilitating the development and evaluation of Natural Language Interfaces for Data Visualization (Srinivasan et al., 2021; Song et al., 2023). Despite their value, the existing resources exhibit three significant limitations that hinder the development of comprehensive, robust, and versatile Text-to-Vis systems (Luo et al., 2021a). First, many of these datasets lack contextual dependencies, focusing on isolated queries rather than sequences of related queries within a conversational context. This lack of context fails to reflect the interactive nature of real-world data visualization tasks, where users typically explore their data through a series of related queries (Guo et al., 2021). Second, these datasets often exhibit a narrow range of programming code types. The dominance of a single code type in these datasets makes it challenging to develop and test models capable of generating diverse visualization codes. Finally, existing resources typically offer a limited diversity of visualization charts. The variety of chart types in real-world data visualization tasks far exceeds those covered in current datasets, limiting the ability of trained models to handle complex, real-world Text-to-Vis tasks. Addressing these limitations is crucial to developing fully functional, robust, and user-friendly NLIDV systems.

To address these issues, we present InterViz, a large-scale, pragmatic dataset designed to interactively generate data visualization code from natural language inputs within a conversational context for the Text-to-Vis task. InterViz offers a diverse set of natural language queries and corresponding visualization codes, encapsulating a broad array of visualization types and code formats. The dataset comprises 6,656 coherent question sequences, resulting in 20,661 NL2Vis pairs drawn from 160 databases. These sequences are structured to emulate real-world interactive data visualization tasks, typically comprising 2-4 turns in a conversation, enabling the investigation and modeling of contextual dependencies. Figure 2 shows an example from the InterViz dataset, where the context and the previous question are essential to parse the next turn. Additionally, InterViz incorporates various programming languages for visualization, including Vega-Lite, ECharts, and Matplotlib, as well as the inter-media form named DV query. This variety promotes the development of models capable of generating an extensive array of visualization codes. Further enhancing its distinctiveness, InterViz covers a multitude of visualization chart types, surpassing the typical selection found in existing datasets (Luo et al., 2021a).

InterViz is constructed through a combination of methods, resulting in two distinct yet complementary subsets: InterViz-S and

InterViz-D. These subsets are crafted using a multi-faceted approach that includes expert knowledge, crowdsourcing, and large language models (LLMs). For InterViz-S, we use a transformational methodology inspired by nvbench (Luo et al., 2021a) to convert SQL interactions from the Text-to-SQL dataset SparC (Yu et al., 2019) into visualization code. This process involves sequential steps of deletions and insertions on SQL trees, producing a candidate set of visualizations that are further filtered to obtain the most contextually fitting ones. Meanwhile, InterViz-D is built from scratch using only database schema information due to the limited visualization diversity in the SparC dataset. This subset creation involves LLMs generating interactions based on the database schema, incorporating context-dependent techniques such as coreference and ellipsis. We further employ in-context learning and Chain-of-Thought methods to generate relevant, coherent, and diverse interactions. These techniques, combined with a robust review process involving experts and crowdsourcers, ensure that InterViz offers a diverse and high-quality platform for developing and evaluating NL2Vis models.

To gain a deeper understanding of the characteristics of InterViz and to establish initial benchmarks, we conducted comprehensive data analysis and experimental evaluations using state-of-the-art approaches. We utilized both fine-tuning-based approaches, such as the T5-family models, and in-context-learning-based approaches, like ChatGPT, with the Chain-of-Thought technique. The best approach achieves an exact match accuracy of 54.2% overall questions and 42.3% overall question sequences, indicating that InterViz presents significant challenges for future research. Additionally, we observe that in-context-learning-based approaches significantly outperform fine-tuning-based methods in Interaction Match accuracy, demonstrating their potential in handling complex, context-dependent NLIDV tasks.

In summary, this paper makes the following main contributions:

- We present InterViz, a large-scale, context-dependent dataset for NL2Vis tasks, with totaling 6,656 coherent question sequences and 20,661 NL2Vis pairs. Developed through an innovative collaboration between experts, crowdsourcers, and AI, InterViz introduces a novel conversational paradigm that captures the contextual dependencies present in real-world NL2Vis interactions, and significantly broadens the diversity of visualization codes and chart types beyond existing benchmarks.
- We provide a comprehensive understanding of InterViz and establish baseline performance metrics for NL2Vis tasks.
- Experimental results on InterViz present limited ability of SOTA fine-tuning-based and in-context-learning-based approaches in NL2Vis tasks.

2 Related Work

2.1 Natural Language Interface for Data Visualization

Significant progress has been made in Natural Language Interfaces to Data Visualization (NLIDV) over the past few years (Shen et al., 2023). NLIDV bridges the gap between natural language processing and data visualization, aiming to create intuitive and efficient interfaces for data analysis. Early efforts in this field (Gao et al., 2015) focus on facilitating natural language interaction with static datasets. These systems use template-based or rule-based approaches to convert natural language queries into visualizations. More recent work, including Eviza (Setlur et al., 2016) and NL4DV (Narechania et al., 2021), have moved toward interactive visualization environments that support free-form dialogue and iterative refinement of visualizations. These systems employ machine learning techniques to interpret natural language and generate the corresponding visualizations. Recently, the advent of Large Language Models has further enriched this field, inspiring new interfaces based on LLMs (Chen et al., 2022).

2.2 NL2Vis benchmarks

The development and advancement of Text-to-Vis tasks have been significantly aided by introducing several benchmark datasets. Early creations such as Vega-Lite (Satyanarayan et al., 2017) and Echarts (Li et al., 2018) offer repositories of visualization examples that could be

used for reference and exploration. However, they lack the structured pairing of natural language queries with corresponding visualization codes necessary for model training and evaluation. In response, benchmarks like the Text-to-Vis Challenge and nvbench have been developed (Kumar et al., 2016; Luo et al., 2021a; Song et al., 2023). These resources provide pairs of natural language descriptions with corresponding visualization codes, enabling the training and evaluation of models for Text-to-Vis tasks. Despite these advancements, current benchmarks are limited in their coverage of context-dependent interactions, various code types, and diversity of visualization charts.

2.3 NL2Vis approaches

The evolution of Text-to-Vis approaches mirrors the broader progress in natural language processing and understanding. Initial systems are predominantly rule-based, relying on a predefined set of rules or templates to translate natural language queries into visualization code (Gao et al., 2015). The advent of deep learning models brings about a generation of approaches based on Seq2seq and Transformer models. They are trained on pairs of natural language queries and visualization code, learning to generate the code directly from the query (Dibia and Demiralp, 2019; Luo et al., 2022; Song et al., 2022). More recently, the emergence of large language models (LLMs) like GPT-3 (Brown et al., 2020), T5 (Raffel et al., 2020), and ChatGPT, has presented new opportunities for Text-to-Vis tasks (Mitra et al., 2022; Maddigan and Susnjak, 2023). These models, trained on vast amounts of text data, have shown remarkable abilities in various language tasks, including Text-to-Vis. In particular, methods such as Chain-of-Thought (Wei et al., 2022) and in-context-learning (Dong et al., 2023), have been utilized to prompt these models, leveraging their ability to maintain and use context over extended interactions. This has been shown to enhance their performance on context-dependent tasks.

3 Problem Definition

The Text-to-Vis task aims to translate a user's natural language question into a data visualization query (Song et al., 2022). Formally, the input is $x = (q, s)$, where q represents natural language questions for the user's visualization need, and s is the corresponding database schema. The schema s consists of a set of tables $T_x = \{t_i\}_{i=1}^{n_t}$ and for each table $t_i \in T_x$, a collection of columns $C_x = \{c_{i,j}\}_{j=1}^{L_i}$, where n_t is the number of tables in the schema and L_i is the count of columns in table t_i. The task of Text-to-Vis aims to synthesize the appropriate visualization query y. Typically, these systems convert natural language into a Data Visualization Query (DV Query). This DV Query is subsequently transformed into visualization specification code compatible with various libraries, such as ECharts, Vega-Lite, and Matplotlib.

4 Dataset Construction

We create InterViz through a synergy of methods, generating two complementary subsets: InterViz-S and InterViz-D. Each subset offers unique insights and challenges, contributing to a comprehensive understanding of the NL2Vis task. For InterViz-S, our approach is inspired by the methodology employed by nvbench (Luo et al., 2021a) to synthesize NL2Vis dataset from NL2SQL benchmarks. In this case, we transform all interactions from the text-to-SQL dataset SparC into visualization code. InterViz-D is created via a trilateral framework involving experts, crowdsourcers, and large language models (LLMs). In this framework, LLMs initially generate interactions based on the Spider database schema. Subsequently, these generated interactions are reviewed and refined by experts and crowdsourcers, guaranteeing high accuracy and relevance.

4.1 Construction of InterViz-S

For constructing InterViz-S, we build on the techniques used in nvbench, incorporating their SQL tree transformation approach.

In the first stage, we perform deletions and insertions to edit the original SQL queries. Deletions are applied to the *Select* and *Order* features of the SQL trees. The Select deletions consider the number of attributes typically needed for different visualizations, while Order deletions address that some visualizations, such as pie charts, do not require order. The insertion process, on the other hand, in-

Dataset	Language	# DB	# Seq.	# Query	# Avg. Turn	Contextual Dependency
Spider	SQL	200	10,181	10,181	1.0	×
SparC	SQL	200	4,298	12,726	3.0	✓
CoSQL	SQL	200	3,007	15,598	5.2	✓
nvbench	VQL/Vega-Lite	153	7,219	7,219	1.0	×
InterViz	VQL/Vega-Lite	160	6,656	20,661	3.1	✓
InterViz-S	/ECharts	160	3,456	10,235	3.0	✓
InterViz-D	/Matplotlib	160	3,200	10,426	3.3	✓

Table 1: An overview comparison between InterViz and other Text-to-SQL/Vis benchmarks.

volves creating new *Group* and *Visualize* features, potentially modifying Order operations, and ensuring the resulting visualization types (e.g., bar, line) are valid and follow data visualization conventions. This process generates a set of candidate visualization queries.

In the second stage, we perform a filtering process on the candidate visualizations. For each NL2SQL interaction in the SparC dataset, we pick the most corresponding visualization from the candidate set according to the context and semantics of the interaction. This careful selection process results in a diverse and representative set of NL2Vis interactions for InterViz-S, ensuring high relevance and accuracy in the final dataset.

The original SParC dataset consists of 4,298 question sequences and 160 databases, but only 3,456 and 160 are publicly available for training and development. Hence, we could only transform those to construct InterViz-S.

4.2 Construction of InterViz-D

Unlike InterViz-S, which starts with a pre-built application, InterViz-D is built from scratch using only the database schema in the Spider dataset as its foundation. This approach is necessary since the SparC dataset, an NL2SQL dataset, focuses more on retrieving necessary data than its visualization, resulting in a limited range of visualization types. To ensure a more diversified and contextually nuanced dataset, we engineer several context dependencies, such as coreference and ellipsis, to guide the LLMs in the generation process. The measurement of contextual dependency in InterViz is shown in Table 2

The construction of InterViz-D heavily leverages the capabilities of Large Language Models (LLMs). Utilizing techniques such as in-context learning and Chain-of-Thought, we can harness the power of LLMs to generate natural language queries that are contextually accurate, coherent, and cohesive. In-context learning enabled LLMs to understand and maintain the conversational context, ensuring the generated queries were contextually relevant. On the other hand, the Chain-of-Thought methodology ensures that instructions given to the LLMs during the generation process are precise and contextually aware, promoting the creation of diverse visualization types.

Furthermore, we design a prompting strategy encouraging LLMs to explore a wide spectrum of visualization types, thereby enhancing the diversity of the final dataset. This construction process involves a collaborative framework of experts, crowdsourcers, and AI, wherein interactions generated by the LLMs are refined and validated. This collaboration ensures that the resulting InterViz-D dataset is relevant and accurate and covers a diverse array of data visualizations.

Dataset	Context Independent	Context Dependent		
		Overall	Coreference	Ellipsis
SParC	47.5%	52.5%	36.6%	20.9%
CoSQL	68.2%	31.8%	18.1%	4.9%
InterViz	54.8%	45.2%	30.7%	20.6%
InterViz-D	61.5%	38.5%	25.3%	20.4%
InterViz-S	47.5%	52.5%	36.6%	20.9%

Table 2: Measurement of Contextual dependency

4.3 Data Review and Post-process

Due to the dataset's complexity and significance, a thorough data review process has

been implemented to ensure its quality. Initially, the generated data interactions are reviewed by an expert panel of individuals with a deep understanding of natural language processing and data visualization. The expert panel scrutinizes each interaction, evaluating its accuracy, relevance, and adherence to the contextual rules established in the generation process.

Following the expert review, the interactions undergo a crowdsourced verification process. Leveraging the power of the crowd, this stage ensures an additional layer of validation by engaging a diverse group of reviewers to assess the interactions in terms of readability, clarity, and overall coherence.

The data are subjected to a post-processing stage, where any inaccuracies or inconsistencies identified during the review process are rectified. The data are further fine-tuned to enhance its usability and comprehensibility, ensuring that the InterViz dataset's final version is high in quality and fit for application in developing and testing next-generation text-to-Vis systems.

5 Data Statistics

The InterViz dataset is comprehensive in its scope, consisting of 6,656 coherent question sequences, amounting to 20,661 NL2Vis pairs, as shown in Fig. 1. These sequences are drawn from a wide array of sources, covering 160 distinct databases, thereby encapsulating a broad range of topics and ensuring the generalizability of the results derived from this dataset.

Regarding the specific coding languages, the InterViz dataset includes three popular languages for data visualization: Vega-Lite (Satyanarayan et al., 2017), ECharts (Li et al., 2018), and Matplotlib, as well as VQL, a SQL-like pseudo syntax for querying a database and simultaneously specifying a visualization. This multi-language characteristic provides a richer set of potential visualizations and further contributes to the diversity and comprehensiveness of the dataset.

The dataset has been divided into three distinct subsets to facilitate training, validation, and testing processes. Specifically, 80% of the data has been allocated to the training set, 10% to the development set, and the remaining 10% to the test set, as illustrated in Fig. 3. This standard split ensures sufficient data for model training while providing independent subsets for model tuning and performance evaluation.

Dataset	Split	# DB	# Seq.	# Query
InterViz	Train	128	5,328	16,516
	Dev	16	644	2,058
	Test	16	644	2,087
InterViz-S	Train	140	3,034	9,032
	Dev	20	422	1,203
	Test	-	-	-
InterViz-D	Train	128	2,560	8,346
	Dev	16	320	1,044
	Test	16	320	1,036

Table 3: Dataset split statistics

6 Evaluation Metrics

6.1 Exact Matching

The primary metric used for evaluating the performance of models on the InterViz dataset is Exact Match (EM) accuracy. This metric is often employed in tasks requiring the model's output to match the target output perfectly, such as code generation and translation tasks. In the context of the Text-to-Vis task, the output visualization code generated by the model is compared with the ground truth visualization code. If the two codes are identical, the model is said to have achieved an exact match. This is a highly stringent metric because it requires the model to produce the exact code without any deviation. Even trivial differences or discrepancies in order or formatting can cause an otherwise correct response to be marked as incorrect. As such, a high EM score indicates that a model can produce precise and accurate visualization code that exactly corresponds to the natural language query and context provided.

6.2 Component Matching

Complementary to Exact Match accuracy, we apply Component Accuracy to provide a more fine-grained assessment of the models' performance on the InterViz dataset. This metric is divided into three categories: Vis Type Accuracy, Data Transformation Accuracy, and Axis Accuracy. Each of these measures the model's ability to correctly generate specific

components of the data visualization code.

6.3 Valid Accuracy

We also employ the Valid Score to evaluate models on the InterViz dataset. The Valid Score is a straightforward yet crucial measure that captures the extent to which the visualization code generated by a model is executable without errors. It is computed as the ratio of valid codes to the total number of generated codes. Mathematically, it is defined as Valid Score $= N_{\text{valid}}/N$, where N_{valid} is the number of visualization codes that can be run without errors and N is the total number of generated codes.

6.4 Interaction Match

To measure model performance in multi-round scenarios, we must distinguish two metrics: Question Match and Interaction Match. Question Match quantifies the extent to which the model's predictions match the ground truth at the question level within an interaction. The score is computed as the total number of questions with exact matches divided by the total number of questions. Interaction Match goes one step further, considering the entire interaction as a single unit for evaluation. In this case, a score of 1 is assigned to an interaction if and only if there is an exact set match for every question within the interaction. The score is then calculated as the total number of interactions with exact matches divided by the total number of interactions.

7 Experiments

7.1 Baselines

In order to gain insights into how state-of-the-art (SOTA) approaches perform on the InterViz dataset, we conduct comprehensive experiments with various methods. The chosen approaches span two major categories: fine-tuned (FT)-based approaches and in-context learning (ICL)-based approaches.

7.1.1 FT-based Models

The FT based models include the T5 (Text-to-Text Transfer Transformer) family (Raffel et al., 2020), pre-trained transformers designed to solve various NLP tasks. Here, we employ T5-base, T5-large, and T5-3B. For each of these models, we fine-tune and evaluate them on the InterViz dataset. All models are trained on an NVIDIA A100 GPU at a learning rate of 1e-4 and pretrained model weights are taken from the huggingface hub.

7.1.2 ICL-based Models

In addition to the finetune-based models, we explore applying an in-context learning-based model, specifically, the ChatGPT model, known as gpt-3.5-turbo. To adapt ChatGPT to the Text-to-Vis task, we implement the Chain-of-Thought (CoT) technique, which guides the model's generative process by structuring the input prompt into several incremental steps, complemented with the lead sentence, 'Let's think step by step.' However, we observe that without explicit guidance, the output of ChatGPT may be unpredictable, occasionally generating unexpected formats and explanations. To mitigate this issue, we leverage in-context learning, providing ChatGPT with a one-shot example that illustrates the correct procedure of thought and output format. All experiments were performed through the OpenAI API.

7.2 Result Analysis

Table 4-5 report the performance of the state-of-the-art models on the InterViz dataset. The results show that: (1) Both FT-based models and LLMs exhibit potential but also encounter substantial challenges. The highest-performing model only achieves an Exact Match Accuracy of 54.2% overall questions and 42.3% overall interaction sequences. This highlights the complex nature of the Text-to-Vis task and demonstrates that there is still considerable room for improvement. (2) In-context-learning-based approaches substantially surpass fine-tuning-based methods in Interaction Match accuracy. This indicates their aptitude in managing complex, context-dependent NLIDV tasks. (3) LLMs can comprehend the complex grammar of visualization specifications when provided with suitable context and prompts. However, variations in accuracy occur between parts of the vis specifications, which implies that future studies may benefit from a stronger focus on prompt strategies.

7.3 Case Study

In this section, we present a detailed case study further to illustrate the capabilities and limita-

Model	Development Data			Test Data		
	QM	**IM**	**VA**	**QM**	**IM**	**VA**
Finetune − based − model						
T5-base	42.5	16.4	75.4	43.0	16.9	77.2
T5-large	48.8	18.9	79.6	49.7	17.8	78.5
T5-3B	53.5	21.5	83.2	54.2	21.8	84.0
ICL − based − model						
gpt-3.5-turbo zero shot	32.5	32.2	86.8	32.1	30.9	87.3
gpt-3.5-turbo one shot	46.6	41.7	97.2	47.1	42.3	98.3
gpt-3.5-turbo + COT	48.2	40.8	98.1	45.5	38.7	97.9

Table 4: Evaluation results on InterViz, as measured by Question Match (QM) accuracy, Interaction Match (IM) accuracy and valid score (VA)(%)

Model	Development Data			Test Data		
	Vis	**Axis**	**Data**	**Vis**	**Axis**	**Data**
gpt-3.5-turbo zero shot	95.4	71.4	81.6	96.1	72.1	82.3
gpt-3.5-turbo one shot	97.0	73.2	82.5	98.2	73.6	83.0
gpt-3.5-turbo + COT	96.8	74.1	80.8	98.1	75.3	81.9

Table 5: Evaluation results on InterViz, as measured by Vis component accuracy (%)

tions of the evaluated models. For this purpose, we consider a typical interaction sequence from the InterViz dataset. The sequence includes an initial natural language query requesting a specific visualization and several related questions that build upon the previous interaction.

Let's consider a query: show me the number of different medications we have. For this query, the T5-based models can generate the appropriate visualization command but face difficulties when the interaction sequence introduces more complexity. For example, if the next question in the sequence is 'What's the distribution of those medications being prescribed?', the models show difficulty in maintaining context and correctly adjusting the visualization.

On the other hand, when we evaluate ChatGPT with the Chain-Of-Thought technique, we realize that it performs more effectively in handling the context of the interaction sequence. It successfully generates the appropriate commands to adjust the visualization according to the new question. However, when the interaction sequence becomes longer, or when the questions become more ambiguous, even ChatGPT struggles to generate the correct commands.

This case study highlights the strengths and weaknesses of the current state-of-the-art models in handling the Text-to-Vis task. While we observe promising results, the models' difficulty maintaining context over long interaction sequences and handling ambiguous queries indicates the need for continued advancements in these areas.

8 Conclusion

In this work, we present InterViz, a comprehensive multi-turn dialogue dataset specifically designed for the challenging task of natural language interface for data visualization. Through a fusion of curated and AI-generated interactions, InterViz provides new opportunities for investigating the capabilities of current language models in translating natural language queries into data visualization code. Through our experiments and case study, we have highlighted the strengths and weaknesses of state-of-the-art models like T5 and ChatGPT, highlighting areas ripe fo further exploration and improvement. We hope this InterViz will encourage the development of more intelligent, context-aware models capable of handling complex, multi-turn interactions in a broader array of applications.

Limitations

Despite the innovative nature and contributions of the InterViz dataset, we acknowledge that it presents certain limitations. First, while the collaborative approach for creating InterViz-D guarantees a high level of quality and relevance, the reliance on language models may also introduce biases or blind spots of the models into the dataset. Besides, the current evaluation metrics, although comprehensive, might not entirely reflect the nuances of users' requirements for data visualization and can produce false-negative cases. For example, multiple correct visualization codes may exist for a given natural language query, and our exact match accuracy might penalize slight variations of incorrect visualizations. In the future, we aim to refine these metrics better to capture the complexity and diversity of the text-to-vis task.

Ethics Statement

In conducting this research and constructing the InterViz dataset, we have made a concerted effort to adhere to ethical guidelines and considerations. The use of large language models and crowdworkers to construct the InterViz-D dataset is performed with care to ensure that neither models nor crowdworkers are exposed to inappropriate or sensitive content. In our collaborative framework, crowdworkers' contributions are anonymized, estimated, and remunerated fairly. Furthermore, the resulting dataset has been carefully curated to exclude potentially sensitive or personal information. The InterViz dataset and our research methods aim to advance the field of data visualization and natural language processing in an inclusive, accessible, and ethically considerate manner. Nevertheless, we acknowledge that our work, like any, is not free from potential misuse or unintended consequences. We encourage the research community to use InterViz responsibly and consider the ethical implications when designing and deploying Text-to-Vis systems.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Yiru Chen, Ryan Li, Austin Mac, Tianbao Xie, Tao Yu, and Eugene Wu. 2022. NL2INTERFACE: interactive visualization interface generation from natural language queries. *CoRR*, abs/2209.08834.

Weiwei Cui, Xiaoyu Zhang, Yun Wang, He Huang, Bei Chen, Lei Fang, Haidong Zhang, Jian-Guang Lou, and Dongmei Zhang. 2020. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE Trans. Vis. Comput. Graph.*, 26(1):906–916.

Victor Dibia and Çagatay Demiralp. 2019. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE Computer Graphics and Applications*, 39(5):33–46.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G. Karahalios. 2015. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST 2015, Charlotte, NC, USA, November 8-11, 2015*, pages 489–500. ACM.

Jiaqi Guo, Ziliang Si, Yu Wang, Qian Liu, Ming Fan, Jian-Guang Lou, Zijiang Yang, and Ting Liu. 2021. Chase: A large-scale and pragmatic chinese dataset for cross-database context-dependent text-to-sql. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2316–2331. Association for Computational Linguistics.

Abhinav Kumar, Jillian Aurisano, Barbara Di Eugenio, Andrew E. Johnson, Alberto Gonzalez, and Jason Leigh. 2016. Towards a dialogue system that supports rich visualizations of data. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest*

Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA, pages 304–309. The Association for Computer Linguistics.

Deqing Li, Honghui Mei, Yi Shen, Shuang Su, Wenli Zhang, Junting Wang, Ming Zu, and Wei Chen. 2018. Echarts: A declarative framework for rapid construction of web-based visualization. *Vis. Informatics*, 2(2):136–146.

Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. Deepeye: Towards automatic data visualization. In *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*, pages 101–112. IEEE Computer Society.

Yuyu Luo, Jiawei Tang, and Guoliang Li. 2021a. nvbench: A large-scale synthesized dataset for cross-domain natural language to visualization task. *CoRR*, abs/2112.12926.

Yuyu Luo, Nan Tang, Guoliang Li, Chengliang Chai, Wenbo Li, and Xuedi Qin. 2021b. Synthesizing natural language to visualization (NL2VIS) benchmarks from NL2SQL benchmarks. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*, pages 1235–1247. ACM.

Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. 2022. Natural language to visualization by neural machine translation. *IEEE Trans. Vis. Comput. Graph.*, 28(1):217–226.

Paula Maddigan and Teo Susnjak. 2023. Chat2vis: Fine-tuning data visualisations using multilingual natural language text and pre-trained large language models.

Rishab Mitra, Arpit Narechania, Alex Endert, and John T. Stasko. 2022. Facilitating conversational interaction in natural language interfaces for visualization. *CoRR*, abs/2207.00189.

Arpit Narechania, Arjun Srinivasan, and John T. Stasko. 2021. NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Trans. Vis. Comput. Graph.*, 27(2):369–379.

Xuedi Qin, Yuyu Luo, Nan Tang, and Guoliang Li. 2019. Making data visualization more efficient and effective: a survey. *The VLDB Journal*, 29:93 – 117.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-lite: A grammar of interactive graphics. *IEEE Trans. Vis. Comput. Graph.*, 23(1):341–350.

Vidya Setlur, Sarah E. Battersby, Melanie Tory, Rich Gossweiler, and Angel X. Chang. 2016. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST 2016, Tokyo, Japan, October 16-19, 2016*, pages 365–377. ACM.

Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai, and Jianmin Wang. 2023. Towards natural language interfaces for data visualization: A survey. *IEEE Trans. Vis. Comput. Graph.*, 29(6):3121–3144.

Yuanfeng Song, Xuefang Zhao, and Raymond Chi-Wing Wong. 2023. Marrying dialogue systems with data visualization: Interactive data visualization generation from natural language conversations. *CoRR*, abs/2307.16013.

Yuanfeng Song, Xuefang Zhao, Raymond Chi-Wing Wong, and Di Jiang. 2022. Rgvisnet: A hybrid retrieval-generation neural framework towards automatic data visualization generation. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 1646–1655. ACM.

Arjun Srinivasan, Nikhila Nyapathy, Bongshin Lee, Steven Mark Drucker, and John T. Stasko. 2021. Collecting and characterizing natural language utterances for specifying data visualizations. *CoRR*, abs/2110.00680.

Nan Tang, Eugene Wu, and Guoliang Li. 2019. Towards democratizing relational data visualization. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, pages 2025–2030. ACM.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir R. Radev. 2019. Sparc: Cross-domain semantic parsing in context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4511–4523. Association for Computational Linguistics.

#	Question & Data Visualization Query	Contextual Dependency
Question Sequence $\mathcal{Q}^1$		
q_1^1	Which airlines operate flights between New York and Los Angeles?	Independent
y_1^1	*Visualize BAR SELECT Airline FROM Flights WHERE SourceAirport = 'New York' AND DestAirport = 'Los Angeles'* *("mark": "bar", "encoding": "x": "field": "Airline", "type": "nominal", "y": "aggregate": "count", "type": "quantitative")*	
q_2^1	Can you also show the flight numbers for these flights?	Dependent (Coreference)
y_2^1	*Visualize BAR SELECT Airline, FlightNo FROM Flights WHERE SourceAirport = 'New York' AND DestAirport = 'Los Angeles'* *("mark": "bar", "encoding": "x": "field": "Airline", "type": "nominal", "y": "aggregate": "count", "type": "quantitative", "color": "field": "FlightNo", "type": "nominal")*	
q_3^1	Can you change the chart to a pie chart instead?	Dependent (Ellipsis)
y_3^1	*Visualize PIE SELECT Airline, COUNT(*) FROM Flights WHERE SourceAirport = 'New York' AND DestAirport = 'Los Angeles' WITH LEGEND, COLORS* *("mark": "pie", "encoding": "color": "field": "Airline", "type": "nominal", "theta": "field": "COUNT(*)", "type": "quantitative", "radius": "type": "quantitative")*	
Question Sequence $\mathcal{Q}^2$		
q_1^2	Which devices are available at the shop "Apple"?	Independent
y_1^2	*Visualize BAR SELECT Device FROM Stock WHERE ShopID = (SELECT shop_id FROM shop WHERE shop_name = 'Apple')* *("mark": "bar", "encoding": "x": "field": "Device", "type": "nominal", "y": "aggregate": "count", "type": "quantitative")*	
q_2^2	Can you show the quantity of each device in stock?	Dependent (Coreference)
y_2^2	*Visualize BAR SELECT Device, Quantity FROM Stock WHERE ShopID = (SELECT shop_id FROM shop WHERE shop_name = 'Apple')* *("mark": "bar", "encoding": "x": "field": "Device", "type": "nominal", "y": "field": "Quantity", "type": "quantitative")*	

Table 6: Question sequence examples in InterViz dataset.

NL-round 1	**Can you show me the number of different medications we have?**
Gold	`VISUALIZE bar SELECT COUNT(DISTINCT name) FROM medication`
T5	`VISUALIZE bar SELECT COUNT(DISTINCT name) FROM medication`
ChatGPT	`VISUALIZE bar SELECT COUNT(DISTINCT name) FROM medication`
NL-round 2	What's the distribution of those medications being prescribed?
Gold	`VISUALIZE bar SELECT medication, COUNT(*) FROM prescribes GROUP BY medication`
T5	`VISUALIZE bar SELECT medication_code, COUNT(*) FROM prescribes GROUP BY medication_code`
ChatGPT	`VISUALIZE bar SELECT medication, COUNT(*) FROM prescribes GROUP BY medication`
NL-round 3	And which doctors prescribe the most?
Gold	`VISUALIZE bar SELECT physician, COUNT(*) FROM prescribes GROUP BY physician ORDER BY COUNT(*) DESC`
T5	`VISUALIZE bar SELECT name, COUNT(*) FROM prescribes GROUP BY physician ORDER BY COUNT(*) DESC`
ChatGPT	`VISUALIZE bar SELECT doctor, COUNT(*) FROM prescribes GROUP BY doctor ORDER BY COUNT(*) DESC`

Table 7: Case Study

Prompt Step	Prompt Contents
Task Definition	You are a data synthesizer tasked with generating data from natural language to visualization code based on the provided database schema. Please follow the instruction steps.
Conversation Creation	Create a conversation between a user and a bot, where the user asks natural language queries and the bot responds with corresponding visualization code. Conduct 2-4 rounds of interactions to create a logically coherent conversation.
Context Dependency	Try to establish connections between queries in each round with three types of contextual Dependency types. Independent: The step doesn't depend on the context established in the previous step. Coreference: The user refers back to a previously mentioned aspect. Ellipsis: The user omits redundant information that can be inferred from the context.
Chart Diversity	Aim to include diverse visualization types and achieve balance among different types. Optional visualization types includes bar charts, line charts, pie chart, scatter charts, etc.
Output Format	First output the visualization query in VQL format, which is a SQL-like pseudo syntax for data querying and visualization. Then provide the visualization specification in format of Vega-lite language.
Demonstrations	Here is an example demonstrating the desired output format based on the provided database schema.
Input	The provided database schema is serialized as below.

Table 8: Prompt examples in construction of InterViz-D

Prompt type	Prompt structure
Zero-shot	instruction: "You are a semantic parser to translate natural language question to data visualization query. The corresponding database schema will be provided in the format of CREATE TABLE commands", [question] + [schema]
One-shot	instruction: "You are a semantic parser to translate natural language question to data visualization query. The corresponding database schema will be provided in the format of CREATE TABLE commands. Here is an example showing the format of input and output" + [example] [question] + [schema]
Chain-of-Thought	instruction: "You are a semantic parser to translate natural language question to data visualization query. The corresponding database schema will be provided in the format of CREATE TABLE commands. Let's think step by step" + [example] + [COT steps] [question] + [schema]

Table 9: Prompt examples of different prompt types in inference.