Natural Language Dataset Generation Framework for Visualizations Powered by Large Language Models

Hyung-Kwon Ko
hyungkwonko@gmail.com
KAIST
Republic of Korea

Hyeon Jeon
hj@hcil.snu.ac.kr
Seoul National University
Republic of Korea

Gwanmo Park
gmpark@hcil.snu.ac.kr
Seoul National University
Republic of Korea

Dae Hyun Kim
dhkim16@cs.stanford.edu
KAIST
Republic of Korea

Nam Wook Kim
nam.wook.kim@bc.edu
Boston College
USA

Juho Kim
juhokim@kaist.ac.kr
KAIST
Republic of Korea

Jinwook Seo*
jseo@snu.ac.kr
Seoul National University
Republic of Korea

ABSTRACT

We introduce VL2NL, a Large Language Model (LLM) framework that generates rich and diverse NL datasets using Vega-Lite specifications as input, thereby streamlining the development of Natural Language Interfaces (NLIs) for data visualization. To synthesize relevant chart semantics accurately and enhance syntactic diversity in each NL dataset, we leverage 1) a guided discovery incorporated into prompting so that LLMs can steer themselves to create faithful NL datasets in a self-directed manner; 2) a score-based paraphrasing to augment NL syntax along with four language axes. We also present a new collection of 1,981 real-world Vega-Lite specifications that have increased diversity and complexity than existing chart collections. When tested on our chart collection, VL2NL extracted chart semantics and generated L1/L2 captions with 89.4% and 76.0% accuracy, respectively. It also demonstrated generating and paraphrasing utterances and questions with greater diversity compared to the benchmarks. Last, we discuss how our NL datasets and framework can be utilized in real-world scenarios. The codes and chart collection are available at https://github.com/hyungkwonko/chart-llm.

CCS CONCEPTS

• **Human-centered computing → Visualization**; **Natural language interfaces**.

KEYWORDS

Vega-Lite, natural language datasets, large language models, framework, natural language interfaces, data visualization

ACM Reference Format:
Hyung-Kwon Ko, Hyeon Jeon, Gwanmo Park, Dae Hyun Kim, Nam Wook Kim, Juho Kim, and Jinwook Seo. 2024. Natural Language Dataset Generation Framework for Visualizations Powered by Large Language Models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3613904.3642943

1 INTRODUCTION

Recent advancements in Natural Language Processing (NLP) techniques empowered individuals with limited data analysis and visualization expertise to engage in text-based interaction and execute data visualization tasks [75, 83]. Many studies have incorporated Natural Language Interfaces (NLIs) into their systems to augment more natural and user-friendly interactions [73]. For example, Voder [78] enables querying key data insights within charts using NL sentences, significantly decreasing the reliance on manual programming for data retrieval. Furthermore, users can provide text to receive automatic recommendations for the most appropriate chart types [17, 60], rather than selecting effective representations manually based on graphical language criteria.

While the presence of suitable datasets modeling human behaviors is crucial in developing effective NLIs or tools for visualizations, prior work has repeatedly pointed to the scarcity of sizable pairs of high-quality datasets (chart, NL) [11, 14, 27, 47, 74, 79]. In detail, existing chart collections are occasionally synthetic [47, 98], limited in diversity (e.g., chart type) [11, 51], or are limited to simpler charts (e.g., basic bar charts, univariate line charts) [18]. Making things worse, only a fraction of these collections (17 out of 56) is publicly accessible [11]. Furthermore, prior work builds the NL datasets that goes with the visualizations through crowdsourcing [79]. However, the process can be costly and time-consuming as it requires recruiting specific sets of target users of the system, some of whom

[*] corresponding author

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642943

must meet notably stringent qualification criteria. Moreover, it is challenging to capture the language variations that arise from a diverse spectrum of user expertise, usage scenarios, and personal preferences, although this is essential for addressing the syntactic variations among the target users of the systems in the real-world [20, 73, 97]. What exacerbates the situation is there are multiple types of NL tasks (e.g., captioning, chart generation & modification, and chart question-answering) where each one necessitates a new dataset tailored to the specific task or transfer knowledge.

We present a new collection of 1,981 Vega-Lite specifications (Figure 2). This is the largest set of human-generated charts obtained from GitHub to date. It covers varying levels of complexity from a simple line chart without any interaction (i.e., simple charts) to a chart with four plots where data points are linked with selection interactions (i.e., extra complex charts) (see the charts highlighted with red stroke in Figure 2). As we focus on amassing a richer set of charts in terms of complexity, more than 86% of them are in complex and extra complex levels. Compared to the benchmarks, our dataset shows the highest average pairwise edit distance between specifications, which proves that the charts are highly diverse from one another. Moreover, it contains the largest number of charts with composite views, interactions (e.g., tooltips, panning & zooming, and linking), and diverse chart types (e.g., map, grid & matrix, diagram, etc.) (Table 2).

We also introduce VL2NL, a 3-stage NL generation framework that can be generalized to various NL tasks on visualizations (Figure 3). First, the framework preprocesses the underlying datasets and minifies Vega-Lite specification for efficient and effective usage by an LLM. Next, the framework leverages guided discovery [7] so that LLMs can steer themselves to create varying NL datasets in a self-directed manner. Here, it analyzes and integrates chart semantics (e.g., mark, encoding) with our scaffolding in accordance with the characteristics of each NL dataset. Also, by answering on key questions, it autonomously concentrates on the chart's key features or propose high-level decisions. Finally, the framework applies a score-based paraphrasing (Table 5) with an LLM to simulate and include syntactic variations of human language in NL datasets.

To test VL2NL, we generated L1 captions that simply describe how the chart encodes data, L2 captions that describe the statistical properties of the data in a chart [45], utterances for chart generation [79], and questions for chart question answering [24, 32]. Our experiments showed that the accuracy of the analyzed chart semantics and generated L1/L2 captions is 89.4% and 76.0%, respectively. Moreover, the generated and paraphrased NL datasets showed greater syntactic diversity in terms of 4.75 out of 6 within-distribution metrics on average. Last, we demonstrate the application of our NL datasets in finetuning experiments, and the use of VL2NL in both fully-automatic and mixed-initiative modes within an interactive system for real-world scenarios.

The main contributions of our work are summarized as follows:

- We collect 1,981 real-world Vega-Lite specifications that are diverse and go beyond simple charts;
- We present 3-stage NL dataset generation framework for visualizations powered by LLMs that employs guided discovery and score-based paraphrasing;
- We perform quantitative and qualitative analysis on the NL datasets generated by our framework.

2 BACKGROUND AND RELATED WORK

In this section, we explain Vega-Lite specification and existing chart collections. Next, we present the types of NL datasets that are of particular interest in the context of this work. Last, we explain the use of LLMs in synthesizing NL datasets.

2.1 Chart Datasets

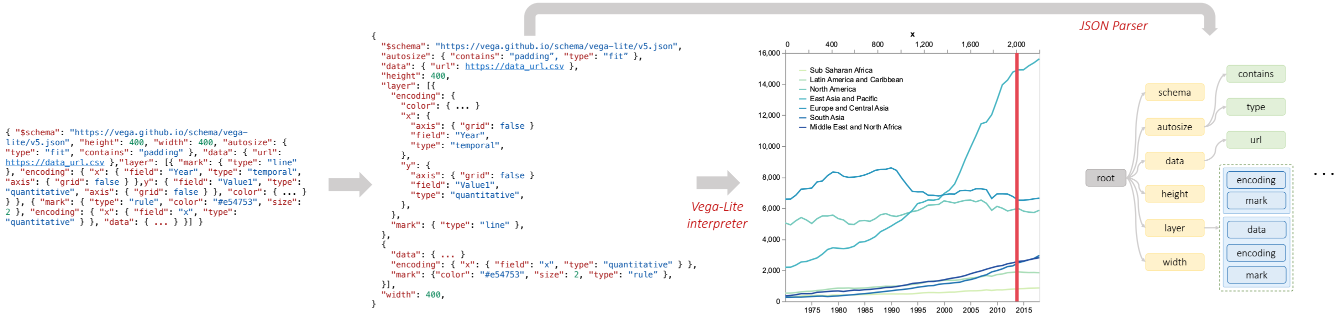
According to Chen et al.'s recent survey [11], chart datasets are typically collected in three formats: bitmap graphics (e.g., `.png`), vector graphics (e.g., `.svg`), and programs (e.g., Vega-Lite specifications [68]). Among the surveyed datasets, the majority (48 out of 56) consisted of bitmap graphics, followed by vector graphics (10 out of 56), while programs were less prevalent, comprising only five instances (some works included multiple formats).

Among many program formats, we are especially interested in Vega-Lite (Figure 1), which is an abstract specification that enables the creation of interactive visualizations using a high-level grammar. It is represented as a nested JSON object, consisting of numerous key-value pairs, which can be also seen as a tree structure [47, 98]. Each key defined in the specification is referred to as a property [89], serving a distinct role in generating charts. For example, `mark` property is used to map data to graphical elements (e.g., points, lines).

Vega-Lite provides additional advantages beyond those offered by SVG formats, since it is easy to modify and reuse for creating diverse chart variations [22]. It provides interactive features like zooming, panning, and brushing, as well as concatenating or faceting multiple plots/views. Furthermore, it support data-driven manipulation, allowing users to dynamically update the data and reflect changes in real time. It can be seamlessly converted to other formats like bitmaps and SVG [69], while converting from those formats to program specifications typically requires manual effort or complex external algorithms [63].

There are two types of Vega-Lite benchmarks: synthetic and real-world datasets. A critical limitation of synthetic datasets lies is their reliance on pre-defined templates and rules, which leads to a high degree of repetition and a limited range of chart types and functionalities (see Table 1). On the other hand, the real-world dataset reveals significant variation from one spec to another, ensuring a high level of diversity in realistic scenarios. However, they are generally much smaller in size compared to synthetic datasets [11].

We found three synthetic Vega-Lite datasets. In detail, Poco et al. generated 4,318 Vega specifications [71] using the Compass recommendation engine [89]. They randomly selected values for a few variables (e.g., fonts, font size, legend positions, etc.) from a curated set of options. These specifications were later converted to Vega-Lite specifications in Data2Vis [18]. Zhao et al. [98] followed a similar approach to generate the Chartseer dataset, consisting of 9,925 specifications based on Data2Vis, although it is specifically designed for training a deep learning model and may not readily render into charts, making it less suitable for broader research adoption. The nvBench dataset [47] presented 7,274 specifications,



```
{ "$schema": "https://vega.github.io/schema/vega-
lite/v5.json", "height": 400, "width": 400, "autosize": {
"type": "fit", "contains": "padding" }, "data": { "url":
https://data_url.csv },"layer": [{ "mark": { "type": "line"
}, "encoding": { "x": { "field": "Year", "type": "temporal",
"axis": { "grid": false } },y": { "field": "Value1", "type":
"quantitative", "axis": { "grid": false } }, "color": { ... }
} }, { "mark": { "type": "rule", "color": "#e54753", "size":
2 }, "encoding": { "x": { "field": "x", "type":
"quantitative" } }, "data": { ... } }] }
```

(a) Raw Vega-Lite Code Data

```
{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "autosize": { "contains": "padding", "type": "fit" },
  "data": { "url": https://data_url.csv },
  "height": 400,
  "layer": [{
    "encoding": {
      "color": { ... }
      "x": {
        "axis": { "grid": false }
        "field": "Year",
        "type": "temporal",
      },
      "y": {
        "axis": { "grid": false }
        "field": "Value1",
        "type": "quantitative",
      },
    },
    "mark": { "type": "line" },
  },
  {
    "data": { ... }
    "encoding": { "x": { "field": "x", "type": "quantitative" } },
    "mark": {"color": "#e54753", "size": 2, "type": "rule" },
  }],
  "width": 400,
}
```

(b) After sorting keys and setting indentation

(c) Generated Vega-Lite chart

(d) Tree Structure

Figure 1: Example of Vega-Lite Specification. As previously noted in several works [47, 98], Vega-Lite specification can be regarded to follow a tree structure, with its keys (i.e., properties) connected in a nested structure.

representing SQL queries as tree structures and mapping them into Vega-Lite specifications.

There are two real-world datasets that consist of human-generated specifications. For instance, Kim et al. [32] curated 52 charts from various web sources, encompassing two chart types (bar chart and line chart). Additionally, the Vega-Lite gallery example dataset [70], the largest publicly available human-generated collection of Vega-Lite data, provides 716 high-quality examples with diverse chart types and interactions. However, due to the challenges associated with data collection, these datasets have a limited quantity of specifications compared to synthesized datasets. As a result, researchers often face difficulties in finding a comprehensive set of specifications for their own research purposes.

2.2 NLIs for Data Visualization

NLIs for data visualization have garnered significant attention due to their user-friendly nature [74, 80, 86]. These interfaces allow users to focus on their tasks rather than learning how to interact with systems [13]. A recent survey paper [74] suggested six high-level topics (e.g., visualization recommendation) to cluster tasks. They also presented a pipeline with seven stages by extending the classical information visualization pipeline [8].

To address diverse NLI tasks, we considered three types of NL datasets: captions, utterances, and questions. This choice was made based on the analysis of each topic, the number of representative works, and the relevance of NL datasets to their respective tasks.

The first NL dataset is chart caption. The captions can help people communicate and grasp insights in the charts easily, also improving the accessibility for readers of the blind and low vision people [45]. A lot of research delved into this problem leveraging from templates [56] to deep learning models [61, 64, 77].

Lundgard and Satyanarayan [45] proposed a four-level classification of captions where each level contains different semantic content of the same chart: L1 provides elemental and encoded attributes, including chart type and encoding channel; L2 encompasses statistical and relational attributes such as descriptive statistics and correlation; L3 addresses perceptual and cognitive attributes, covering complex trends and patterns; L4 contains contextual and domain-specific knowledge. Recently, VisText [84] generated L2/L3 captions by training ByT5 transformer model [93] with crowdsourced dataset. Our work shares similarities with VisText in generating captions with varying levels. However, it differs in that we do not rely on crowdsourcing NL datasets or training machine learning models. Instead, our approach solely depends on Vega-Lite specification input and vanilla LLMs. It is worth noting that previous studies in caption generation have predominantly focused on basic chart types, as highlighted in [74]. In contrast, our work offers a generalizable solution capable of generating captions for complex and diverse charts.

The second NL dataset is utterance for chart generation. For many decades, automatically representing graphical information has been one of the important topics in information visualization [49]. Many NLIs were introduced and adopted to solve multiple stages that are entangled one another for the automatic representation. The most relevant stages are 1) utterance interpretation [19, 25, 35, 42, 48, 60, 81, 97] and 2) mapping utterances to visual elements [26, 30, 46, 50, 57, 82, 85, 88, 90], and 3) human interaction for clarifying ambiguity or suggesting commands [20, 25, 59, 60, 73].

Srinivasan et al. [79] analyzed the characteristics and semantics of NL utterances employed in chart generation. According to their research, NL utterances can be classified into three types based on their structures: commands, which are instructions or systematic requests; queries, which are concise lists of keywords similar to web search queries; and questions, which are data-driven inquiries that users wish to visualize. In our work, we generate all three types of utterances, incorporating heightened syntactic diversity for a comprehensive evaluation.

The last NL dataset is question. Chart Question Answering is a popular task in both machine learning [10, 43, 44, 52] and human-computer interaction [32] communities. This popularity stems from its effectiveness in eliciting insights and aiding in decision-making processes [24].

Kim et al. [32] investigated the semantics used in the questions by collecting 629 crowd-sourced questions and provided two orthogonal dimensions. First axis is lookup or compositional, which is whether to retrieve a single value or using multiple mathematical operations. Second axis is visual or non-visual, which is whether to reference visual chart features or not. These question types are all focusing on retrieving factual short answers. In our work, we

Table 1: Summary of the Vega-Lite dataset construction process. First we collect all possible cases of URLs including Vega-Lite specifications (a). Next, we have filtered unique URLs that are allowed to redistribute for academic purpose (b, c). Finally we iteratively inspect each specification manually to check whether it is valid and unique, since we want to collect charts with a high level of diversity (d).

	# of URLs / Vega-Lite specs
(a) URLs crawled	67,789
(b) URLs w/o duplicate	18,420
(c) URLs w/ license	7,408
(d) Specs after manual inspection	1,981

target five different types of questions, including the aforementioned types as well as the open-ended question type [24], which encourages deeper reflection on the underlying reasons or causes behind specific events or patterns.

2.3 LLMs and NL Datasets

Many past research typically have used crowdsourcing to collect varying types of NL datasets (e.g., captions, utterances, questions, etc.) by asking crowd workers to come up with generation queries using available chart datasets [32, 47, 79]. However, this approach is frequently time-consuming and costly [16, 91], which can adversely affect the scalability of datasets. It is prone to issues such as participant laziness and the collection of subpar queries [5]. To ensure a consistent performance among workers, it is essential to simplify the tasks and making them easy to follow, thereby preventing workers from feeling overwhelmed or fatigued during the study, as recommended by Kittur et al. [36]. With all these efforts, such crowdsourced NL datasets are often fragmented, posing challenges for researchers seeking to apply them to their own tasks. The characteristics of NL queries designed for each task can vary significantly, making a single NL dataset unsuitable for other tasks. This motivates the need for a unified and adaptable framework that can generate NL datasets tailored to any specific NLIs for data visualization research.

As LLMs are known to simulate human behavior [62] and have become more prevalent due to their powerful performance, researchers are increasingly using generated NL datasets to train smaller-sized language models for specific tasks [55, 72, 95]. This training strategy is known as 'teaching via data' [41]. Here, LLMs, acting as teacher models, generate synthetic datasets which are then used to train smaller-sized models, referred to as students, designed for specific tasks. This method is adopted to increase the performance of different tasks like knowledge-based question answering [41], symbolic language generation (e.g., SQL query) [96], and semantic parsing [67]. Our work aligns with this trend, aiming to assist researchers in developing NLIs for data visualization by generating the necessary NL datasets using LLMs.

3 VEGA-LITE DATASET

We have collected a new set of 1,981 real-world Vega-Lite specifications. In this section, we present the details of our data collection process.

3.1 Dataset Construction

3.1.1 Search Queries. We utilize the GitHub API[1] to create our Vega-Lite dataset. Due to the API's limitation of providing up to 1,000 results per search query, we employ various techniques, as we elaborate below, to crawl Vega-Lite specifications in a mutually exclusive and exhaustive manner to the best of our abilities.

When building search queries, we use the keyword `https://vega.github.io/schema/vega-lite/[version]` to indicate the version of the specification that Vega-Lite uses for rendering purposes. We collect versions from v2 to v5: there are no v1 data to be found. To partition the query into a more fine-grained manner, we use keywords such as `.csv` and `.json` to gather specifications with external links. Similarly, we employ keywords like `values` and `datasets` to identify ones with internally embedded data. We also leverage additional keywords using the main properties defined in the version 5 Vega-Lite specification[2]. These properties encompass essential elements for creating a single plot, including `data`, `transform`, `mark`, and `encoding`, while there are properties like `layer`, `facet`, `concat`, and `repeat`, which are specifically relevant to visualizing *composite* views [68] (e.g., layered plots, trellis plots, or multiple views). A comprehensive list of the properties we use can be found on the official documentation page[3].

3.1.2 Inclusion and Exclusion Criteria. We target files with extension `.json`, `vg.json`, `.vl.json`, `.vl`, and `.vg` which denotes Vega-Lite specifications. We also examine HTML and JavaScript files containing Vega-Lite specifications manually to get additional specifications. Throughout the process, we exclude forked repositories to prevent redundancy. We also filter out any data from the benchmark datasets, such as Vega-Lite gallery [70].

3.1.3 Post-processing. To obtain a large number of unique sets of Vega-Lite specifications, we follow a step-by-step approach. During the initial stage, a total of 67,789 URLs are collected. Despite efforts to ensure a mutually exclusive and comprehensive set of specifications, duplicate URLs are identified and removed, resulting in 18,420 unique URLs. Each URL is scrutinized to verify the license of the corresponding repository, ensuring compliance with copyright regulations for academic redistribution. This process yields 7,408 URLs. Lastly, we verify their validity using the Vega-Lite editor [69]. This involves identifying the URLs of the datasets used by each specification and making necessary modifications, ranging from minor adjustments such as closing unclosed brackets to more significant ones like debugging the entire code, in order to achieve successful rendering. An overview of the post-processing and the number of URLs and specifications obtained at each stage can be found in Table 1. Our chart collection is publicly accessible via the following link: https://hyungkwonko.info/chart-llm-data.

3.2 Quantitative Analysis

3.2.1 Benchmarks. We compare three synthetic and two real-world Vega-Lite datasets [18, 32, 47, 70, 98] described in Section 2. To ensure a fair comparison, we implement a process to remove exact

[1] https://docs.github.com/en/rest
[2] https://github.com/vega/schema
[3] https://vega.github.io/vega-lite/docs

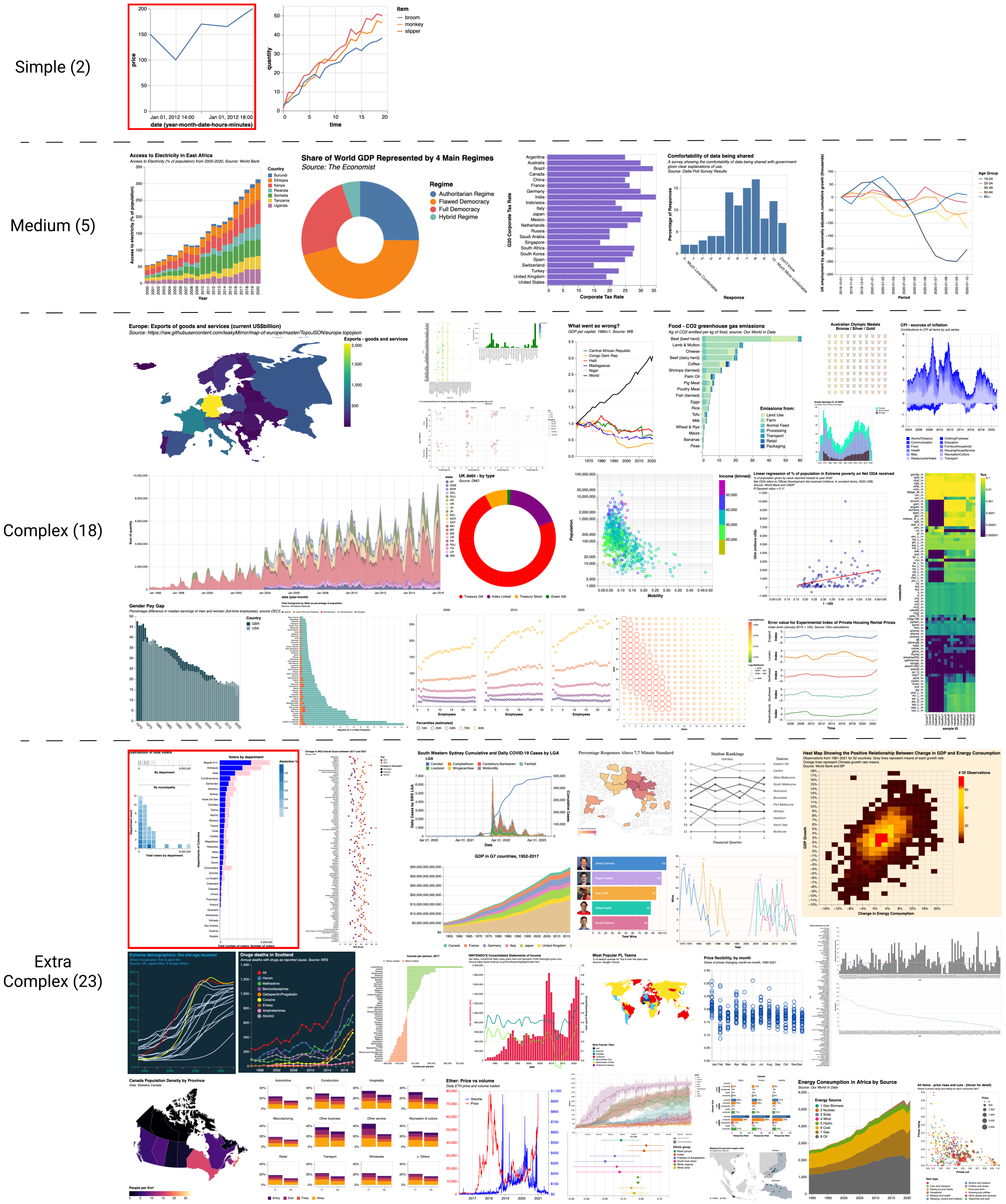


Figure 2: Vega-Lite dataset divided by their complexity levels: simple, medium, complex, extra complex. These 48 charts were selected via stratified sampling and used in our evaluation (Section 5). The level is divided based on the number of keys each specification contains. The number of keys, which are the criteria for dividing the levels, are set based on the quartiles (Q1, Q2, Q3) of Vega-Lite example gallery dataset [70].

code duplication within each benchmark. In detail, each specification is sorted in alphabetical order by the keys and edited to maintain consistent indentation. Next, we convert each file into a hash where files with identical hashes are subsequently removed from the dataset. Following this procedure, the number of specifications in Chartseer dataset decrease from 9,917 to 9,897, nvBench decrease from 7,241 to 6,680, and the Vega-Lite gallery example dataset decrease from 716 to 709.

Table 2: Summary statistics of our dataset and benchmark datasets that are publicly available. Two types of datasets are presented: synthetic and real-world datasets. The best statistics within each type are highlighted in bold, while the best statistics across all datasets are also underscored.

Type	Evaluation Metric / Criteria	Synthetic data (machine-generated)			Real-world data (human-generated)		
		Data2Vis [18]	Chartseer [98]	nvBench [47]	Kim et al. [32]	Gallery [70]	Ours
Quantity	# of specs	4,318	**9,897**	6,680	52	709	**1,981**
Complexity	Total # of keys across specs	101,881	**147,676**	98,074	769	26,469	**107,802**
	Average # of keys in a spec	**24**	15	15	15	37	**54**
	Simple (key ≤ 16)	0	6,164	**6,354**	41	**186**	73
	Medium (key ≤ 24)	**4,318**	3,733	326	10	170	**199**
	Complex (key ≤ 41)	0	0	0	1	179	**733**
	Extra complex (key > 41)	0	0	0	0	174	**976**
	Average depth of JSON	**4.00**	3.00	3.48	3.13	5.01	**5.19**
	Average branching factor	1.22	**1.44**	1.18	1.17	**1.41**	1.38
Diversity	Total # of unique keys	**24**	12	18	31	275	**362**
	Average pairwise edit distance	**122.62**	75.90	48.18	129.51	1,096.11	**1,549.48**
	Composite views	0	0	0	0	136	**746**
	Interaction (e.g., zoom, pan)	0	0	0	0	188	**1,010**
	# of chart types	**6**	**6**	4	2	**10**	**10**

3.2.2 Quality Metrics. To comprehensively assess the Vega-Lite datasets, we consider three different aspects: quantity, complexity, and diversity. Initially, we count the number of collected specifications to determine the overall quantity of Vega-Lite specifications, as previously done by Luo et al. [47]. However, we argue that additional metrics are necessary to gauge the quality of the Vega-Lite dataset. This is because some specifications include only mandatory properties to construct a single plot without any interaction (e.g., `data`, `encoding`, `mark` for a simple bar chart), while others contain multiple plots or views linked by varying interactions. Therefore, the number of keys in a specification can highly differ depending on whether it includes properties for data pre-processing (e.g., `aggregate`, `calculate`, etc.), interactivity (e.g., `bind`, `select`, etc.), or composite views (e.g., `concat`, `repeat`, etc.). We can expect the Vega-Lite specification becomes more complex as the number of defined properties increases. Therefore, we propose a new standard to understand the complexity of a Vega-Lite dataset by counting the total number of keys present across all specifications and the average number of keys in a singe specification. To ensure a fair comparison, we only consider keys defined in the version 5 specification. We also ignore keys associated with internally embedded datasets, such as `values` and `datasets,` along with their corresponding keys. In addition to this, we also measure the average depth and branching factor of the JSON structure as they are commonly adopted to evaluate the complexity of a JSON file.

We found no metrics to quantify the diversity of chart dataset [11]. Therefore, we also propose metrics for gaining insights into the diversity of dataset in terms of both the range of properties within the entire dataset and the variance between individual specifications. Specifically, we count the number of unique keys employed across the entire dataset and calculate the average pairwise edit distance among all possible pairs of specifications. The number of unique keys indicates how many distinct properties that can be defined in a Vega-Lite specification are used across the specifications. For example, if a handful of unique keys are used within the dataset, this indicates a restricted recurrence of only a few properties. In turn, it likely signifies a low level of diversity. The average pairwise edit distance provides an overview of the dissimilarity between each pair at the code level. To perform this analysis, we sort the keys alphabetically, replace their corresponding values with empty values, and exclude keys associated with embedded datasets, as mentioned earlier.

3.2.3 Complexity Levels. We observe that the existing criteria used to establish the complexity levels of charts are somewhat subjective and may not possess broad applicability [31, 45, 47]. Instead, we suggest using the number of keys as a criterion for categorizing the complexity levels of charts, particularly in the context of Vega-Lite specifications. This is because, as explained above, the number of properties increases proportionately to the number of keys in a specification. To establish the standard number of keys, we refer to the Vega-Lite example gallery dataset [70] and calculate the quartiles (Q1, Q2, Q3) based on the distribution of the number of keys. These quartiles, specifically 16, 24, and 41, are utilized as reference points to divide the specifications' level of complexity. For instance, a specification with a total number of keys less than or equal to 16 is classified as 'simple' complexity. Likewise, a specification with a total number of keys greater than 16 and less than or equal to 24 is classified as 'medium' complexity (Figure 2).

3.2.4 Composite View, Interactivity, and Chart Type Distribution. We choose three additional factors by referring to previous works [6, 40] to further assess the quality of the datasets. First, we examine

Table 3: Prompting techniques to generate each NL dataset. Each prompt is designed by choosing the most appropriate techniques considering their different characteristics.

Target	Technique	L1 caption	L2 caption	Utterance	Question
Semantic	($\mathcal{S}$) Scaffolding	O	-	O	-
	($\mathcal{K}$) Key question	-	O	O	O
Syntactic	Paraphrasing	-	-	O	O

the presence of composite views, which offer diverse perspectives on the same data simultaneously [12]. Secondly, considering the benefits of collecting Vega-Lite specifications over static bitmap images, we count the number of charts that incorporate interactive techniques such as tooltips, zooming, and brushing. Lastly, we evaluate the number of charts types based on the taxonomy proposed by Borkin et al. [6].

3.2.5 Results. We present the results in terms of quantity, complexity, and diversity, highlighting the superiority of our dataset compared to the benchmarks. Regarding quantity, all three synthetic datasets demonstrate a higher number of specifications compared to the other three real-world datasets. Among all datasets, Chartseer shows the highest number of specifications (i.e., 9,897), while our dataset has 1,981 specifications which outnumbers the other real-world datasets in terms of quantity.

In terms of complexity, our dataset ranks the first in average number of keys in a single specification (i.e., 54) and the second in total number of keys across specifications (i.e., 107,802), which is 1.4 and 4.0 times larger than the largest previous real-world Vega-Lite dataset, respectively. Chartseer presents the highest total number of keys across specifications (i.e., 147,676) with the smallest average number of keys per specification (i.e., 15) among all datasets. Our dataset includes the highest number of specifications classified as complex (i.e., 733) and extra complex (i.e., 976), while all synthetic datasets do not contain any specifications in the complex and extra complex level. Data2Vis and nvBench demonstrate the largest number of specifications classified as medium (i.e., 4,318) and easy (i.e., 6,354), respectively. Our dataset also exhibits the highest average depth of JSON structure (i.e., 5.19), while Chartseer showcases the highest average branching factor (i.e., 1.44).

Lastly, with respect to diversity, our dataset demonstrates the largest total number of unique keys and the highest average pairwise edit distance among all datasets. Furthermore, our dataset includes the largest number of specifications featuring composite views (i.e., 1,010) and interactions (i.e., 746), exceeding the Vega-Lite gallery dataset by 1.8 and 5.3 times, respectively. None of the synthetic datasets or Kim et al.'s dataset include specifications with composite views and interactions. Both our dataset and the Vega-Lite gallery dataset cover the widest variety of chart types, encompassing ten types: Area, Bar, Circle, Diagram, Distribution, Grid & Matrix, Line, Map, Point, and Trees & Networks. Please refer to Table 2 for detailed results.

4 VL2NL: NL GENERATION FRAMEWORK

The goal of our framework is to generate high-quality NL datasets using Vega-Lite specifications and prompt engineering. VL2NL consists of three stages (Figure 3). First it preprocesses underlying datasets (e.g., `csv`) and minifies the Vega-Lite specifications. Next, it identifies relevant and accurate information through guided-discovery. Last, it increases syntactic diversity using score-based paraphrasing. To generate each type of NL dataset, we design each prompt to be maximally helpful by selecting the most appropriate strategies (Table 3).

4.1 Pre-processing Vega-Lite Specifications

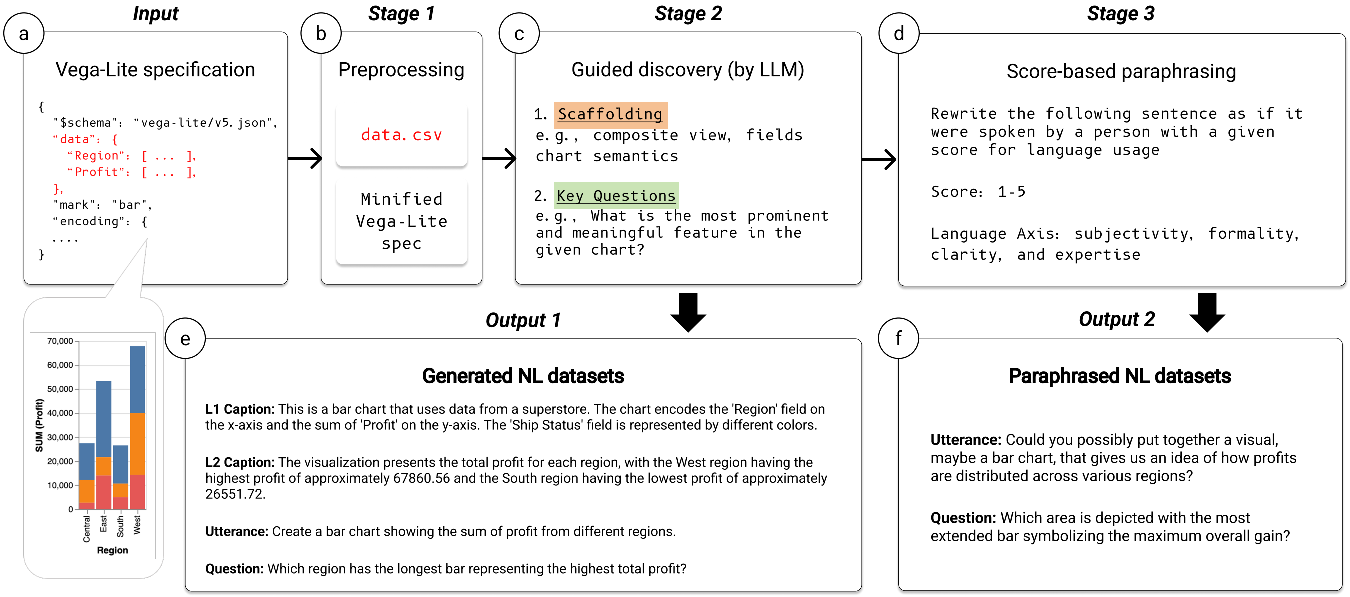
Using raw Vega-Lite specifications is not appropriate for prompting, because some of them include the dataset they use within the specification, resulting in excessively file length. Therefore, we save the data as an external files with the most suitable data formats (e.g., `.csv`, `.json`). Subsequently, the location of the saved files is overwritten with their URLs, rather than being embedded in the specification. In our current implementation of the framework, we only support `.csv` data format. Therefore, we have converted `.json` files into `.csv` files. Last, we minify the Vega-Lite specifications by removing all line breaks and indentations to reduce the number of tokens sent through API usage.

4.2 Ensuring Accuracy and Relevance

Our framework leverages the concept of guided discovery [7] based on Chain-of-Thought prompting [87] to harness the maximum reasoning capability of LLMs. We employ two strategies of guided discovery: providing scaffolds [23]. and posing key questions [15]. To analyze and integrate the chart semantics necessary for generating a specific NL dataset, we assist LLMs by offering scaffolds. Additionally, we furnish LLMs with key questions to guide their self-directed progress. This maximizes the use of LLMs' reasoning abilities, allowing them to make decisions on which aspects to focus on and delve into when creating a particular data. Below, we denote each step where we italicize the relevant phrases, and utilize symbols for ($\mathcal{S}$) scaffolding and ($\mathcal{K}$) key question to make them easily identifiable.

To demonstrate how we can ensure relevance and accuracy in generating different types of NL datasets, we have selected three datasets commonly used in NLIs for data visualization research. We made these selections based on their significance in conjunction with related tasks, as indicated in a recent survey paper [74]: captions (L1, L2), utterance (command, query, question), question (visual-lookup, visual-compositional, nonvisual-lookup, nonvisual-compositional, open-ended). The detailed generation process for each NL dataset can differ from one another. Here, we design each step in prompting to be merged or separated when generating different NL datasets so they can best capture each of their characteristics. We only explain the high-level descriptions of each, and the detailed and full prompting used for generating each NL dataset is presented in Appendix A.

4.2.1 L1 Caption. Considering real-world Vega-Lite specifications, we first understand whether the given chart is ($\mathcal{S}$) a composite view (e.g., layered, trellis, and multiple views), to enable top-down analysis of each chart one by one. The prompt follows a template with three questions to answer: Is it a composite view?; If it is, identify its type among layered, trellis, and multiple views; and determine the number of plots in the chart. Next, it analyzes each



```
{
  "$schema": "vega-lite/v5.json",
  "data": {
    "Region": [ ... ],
    "Profit": [ ... ],
  },
  "mark": "bar",
  "encoding": {
  ....
}
```

```
1. Scaffolding
e.g., composite view, fields
chart semantics

2. Key Questions
e.g., What is the most prominent
and meaningful feature in the
given chart?
```

```
Rewrite the following sentence as if it
were spoken by a person with a given
score for language usage

Score: 1-5

Language Axis: subjectivity, formality,
clarity, and expertise
```

L1 Caption: This is a bar chart that uses data from a superstore. The chart encodes the 'Region' field on the x-axis and the sum of 'Profit' on the y-axis. The 'Ship Status' field is represented by different colors.

L2 Caption: The visualization presents the total profit for each region, with the West region having the highest profit of approximately 67860.56 and the South region having the lowest profit of approximately 26551.72.

Utterance: Create a bar chart showing the sum of profit from different regions.

Question: Which region has the longest bar representing the highest total profit?

Utterance: Could you possibly put together a visual, maybe a bar chart, that gives us an idea of how profits are distributed across various regions?

Question: Which area is depicted with the most extended bar symbolizing the maximum overall gain?

Figure 3: LLM Framework to Generate NL Datasets for Visualizations. We start by (b) preprocessing underlying datasets and minifying Vega-Lite specifications. Subsequently, (c) we employ scaffolding and key questions, (e) to generate NL datasets like L1/L2 captions, utterances, and questions. (d) This is followed by score-based paraphrasing, (f) allowing us to produce syntactically paraphrased NL datasets.

Table 4: Results of automatic qualitative coding [21]. From previous NL datasets of captions, utterances, and questions, we identified four language axes of syntactic diversity: subjectivity, formality, clarity, and expertise. The top five most frequently occurring codes within each axis are presented along with their respective frequencies in parentheses.

Axes	Formality	Clarity	Expertise	Subjectivity
Directions	Colloquial/Standard	Implicit/Explicit	Non-technical/Technical	Subjective/Objective
Example codes	interrogative form (540) formal (384) passive voice (211) analytical (195) command-oriented (166)	specificity (221) specific (91) ambiguity (68) conciseness (64) abstract (63)	economic (159) geographical context (125) financial (106) business-oriented (81) business terminology (65)	descriptive (611) negative connotation (58) subjectivity (22) negative (15) third person perspective (11)

chart individually based on the provided scaffold of $(\mathcal{S})$ chart semantics: Data, Transform, Mark, Chart-Type, Encoding, Style, and Interaction, using the information about composite view. Here, we access the underlying dataset to provide $(\mathcal{S})$ fields that are presented in the Vega-Lite specification, along with their synonyms (i.e., $(\mathcal{S})$ titles also found in the same Vega-Lite specification) and their $(\mathcal{S})$ unique values if they are categorical variables. After analyzing all of these semantics, the LLM finally generates the L1 caption by combining them.

4.2.2 L2 Caption. L2 captions, unlike L1 captions that provide an overall description of the chart, offer the flexibility to selectively focus on specific features that capture the viewer's interest. To craft informative and insightful captions, we follow a structured approach centered around a key question: $(\mathcal{K})$ What is the most prominent and meaningful feature in the given chart? Once we identify this feature, we delve deeper by exploring the mathematical operations required to analyze it: $(\mathcal{K})$ What is the mathematical operation(s) required to describe the feature? Subsequently, based on these operations, we generate $(\mathcal{K})$ a series of questions to analyze the feature (e.g., for the simple line chart with a red border in Figure 2, the following questions are generated: What was the highest stock price of Google?; What was the lowest stock price of Google?; What is the difference between the highest and lowest stock prices of Google?). This process allows us to create captions that provide valuable insights into the chart's content. When answering questions, we utilize backing datasets and LangChain [9] to perform required calculation that bolster the factual integrity of the generated captions. This step is crucial, as large language models (LLMs) have been known to produce hallucinations in response to mathematical problems [28]. Once each question is answered, the collected information is subsequently incorporated into the

Table 5: Example of score-based paraphrasing with two axes. We used a sample LLM-generated utterance to create the chart shown in Figure 3. We perform linear interpolation to manipulate formality and expertise scores from 1 to 5, one at a time to generate 25 paraphrased sentences. While the content (i.e., semantics) of the utterance remains consistent, the tone and voice (i.e., syntax) change linearly in accordance with the provided scores.

Sample Utterance: Create a bar chart showing the sum of profit from different regions.

		Expertise (Non-technical: 1, Technical: 5)				
		1	2	3	4	5
Formality (Colloquial: 1, Standard: 5)	1	Hey, can you whip up a bar graph showing how much dough we've made from different places?	Hey, can you make a bar graph showing the total profit from different regions?	Can you put together a bar chart showing the aggregate profit from various geographical areas?	Can you construct a bar chart illustrating the cumulative profit derived from distinct regions?	Can you generate a bar chart delineating the summation of fiscal gain from disparate geographical sectors?
	2	Could you create a bar chart that shows how much money we've made from different places?	Could you create a bar chart that shows the total profit from different regions?	Could you create a bar chart that illustrates the aggregate profit from various geographical areas?	Could you create a bar chart that delineates the cumulative profit derived from distinct regions?	Could you create a bar chart that represents the summation of fiscal gain from disparate geographical sectors?
	3	Please create a bar chart showing how much money we've made from different places.	Please create a bar chart showing the total profit from different regions.	Please create a bar chart illustrating the aggregate profit from various geographical areas.	Please create a bar chart delineating the cumulative profit derived from distinct regions.	Please create a bar chart representing the summation of fiscal gain from disparate geographical sectors.
	4	It is requested that you create a bar chart showing the money made from different places.	It is requested that you create a bar chart showing the total profit from different regions.	It is requested that you create a bar chart illustrating the aggregate profit from various geographical areas.	It is requested that you create a bar chart delineating the cumulative profit derived from distinct regions.	It is requested that you create a bar chart representing the summation of fiscal gain from disparate geographical sectors.
	5	You are required to construct a bar chart demonstrating the monetary gain from various locations.	You are required to construct a bar chart demonstrating the total profit from different regions.	You are required to construct a bar chart illustrating the aggregate profit from various geographical areas.	You are required to construct a bar chart delineating the cumulative profit derived from distinct regions.	You are required to construct a bar chart representing the summation of fiscal gain from disparate geographical sectors.

final prompting stage for generating L2 captions. It's important to note that, unlike previous work [84], we do not use any of the L1 captions when generating L2 captions. Instead, this is performed as an independent process.

4.2.3 Utterance. Similar to L1 captions, we begin by analyzing whether the chart is a $(\mathcal{S})$ composite view. We then proceed to generate $(\mathcal{S})$ instructions for each plot independently. This process entails creating a comprehensive set of step-by-step instructions for constructing each plot. To enhance readability and user-friendliness, we ensure that each instruction focuses on a single specific action, aligning with the same semantics used when generating L1 captions. For example, in the case of the simple line chart with a red border shown in Figure 2, the following instructions are generated:

- Data: Use Google's stock price data;
- Chart-Type: Create a line chart;
- Mark: Use a line mark;
- Encoding: Encode the x-axis with the date field, using a temporal type and a time unit of year, month, date, hours, and minutes, and scale it using UTC;
- Encoding: Encode the y-axis with the price field, using a quantitative type.

However, it is important to note that the generated instructions may sometimes feature overly technical variable names from the chart, which might not align with users' NL usage patterns. In such cases, we leverage information from synonyms found in the underlying dataset. Specifically, we use $(\mathcal{S})$ title of the Vega-Lite specification and $(\mathcal{S})$ values from the fields to replace the technical terms, resulting in more user-friendly instructions.

Next, we ask a key question to $(\mathcal{K})$ identify primary and secondary information. In this context, we anticipate that LLMs are able to automatically prioritize crucial semantics to paint a comprehensive picture of the chart, such as chart type or encoding, over additional instructions like style or interaction. This thought is based on Wang et al.'s observations [86], who noted that the typical workflow for creating visualizations often starts with this information (e.g., 'show me the price over time as a line chart'). Once we have all these components ready, we proceed to generate each type of utterance one by one, adhering to $(\mathcal{S})$ specific rules for each type. For commands, we employ the imperative voice. For queries, we use only variables, fields, attributes, mathematical formulas, abbreviations, and prepositions, while avoiding verbs and articles. For questions, we formulate inquiries in the form of questions. Across all types, we maintain the following rules: express each utterance in a single sentence, utilize only primary information, and keep the language concise and straightforward.

4.2.4 *Question.* In general, we conduct chart question answering to facilitate decision-making [32]. Thus, the process involves analyzing charts through a question-answering, which ultimately leads to a conclusion and informs the decision-making. To generate questions, we employ a reverse thought process. This entails first identifying the decisions that can be derived from the charts (i.e., ($\mathcal{K}$) What higher-level decision can be made by analyzing this chart?), followed by formulating a possible conclusion that leads to such a decision (i.e., ($\mathcal{K}$) What is a possible conclusion that can be reached from this decision?). Finally, we determine what needs to be analyzed (i.e., ($\mathcal{K}$) What specific value can be retrieved to reach this conclusion? What are the mathematical operations to reach the conclusion?). We generate non-visual lookup and compositional questions using the provided values and mathematical operations. To transform these into visual questions, we identify the necessary visual attributes and incorporate them into the generated questions (i.e., ($\mathcal{K}$) What visual attributes are required to paraphrase this question?). Finally, we formulate an open-ended question designed to lead to the same conclusion obtained in the previous step.

4.3 Increasing Syntactic Diversity

4.3.1 *Automatic Qualitative Coding.* Before increasing the syntactic diversity of NL datasets, we need to analyze which meaningful axes of diversity to address. To this end, we collected sample NL sentences from existing sources, which consist of 2,147 captions, 893 utterances, and 629 questions [32, 45, 79]. Next, following the automatic coding process that Hämäläinen et al. have proposed [21], we utilized these sample sentences to conduct a thematic analysis using LLMs, generating five different codes for each caption, utterance, and question (see the prompt in Appendix B). We manually checked the generated codes to eliminate irrelevant and erroneous ones, resulting in 15,271 valid codes out of 18,345. Then, we retained 2,759 unique codes and vectorized them using Sentence-Bert [65]. Afterwards, we applied dimensionality reduction technique to project them into a lower dimensional space using UMAP [54], reducing the 100-dimensional vectors to 5-dimensional vectors. Next, we employed HDBSCAN [53] to cluster them into a few classes for detailed investigation. We aggregated clusters into a higher-level cluster, except for the codes that are not clustered through HDBSCAN, to derive the final themes.

We identified a total of six themes, but selected four meaningful axes related to NL syntax−clarity, expertise, formality, subjectivity (Table 4). Two themes were removed−1) measurement, and 2) chart and data analytics−as they are not directly related to the syntax of NL datasets but rather to the semantic properties of charts. Clarity represents a language axis with two opposite meanings—implicit and explicit. Implicit language relies on context, shared knowledge, and non-verbal cues to convey meaning, while explicit language is clear and direct, leaving little room for interpretation or misunderstanding. The expertise axis also has two opposite meanings—non-technical and technical. Technical language includes specialized terminology and jargon, whereas non-technical language is more accessible to a general audience and avoids the use of complex terms. Formality, the third language axis, ranges from colloquial, which is informal and used in everyday conversation, to standard, which follows established rules and conventions. Finally, the subjectivity axis encompasses subjective language, which expresses personal opinions, feelings, or judgments, and objective language, which presents facts or information without bias or personal interpretation.

4.3.2 *Score-based Paraphrasing.* Our paraphrasing technique is inspired by a linear interpolation in the latent space for image generation and manipulation as demonstrated in many system and application papers [1, 3, 38, 58]. This technique enables a smooth transition from one expression to another by focusing on creating controllable and meaningful syntactic variations of a single sentence. The key idea is that we assign language axesand employ a five-point Likert-scale to each. Here, we focus on altering only the sentence's syntax, while maintaining its meaning. In detail, we provide LLMs with a sentence we want to paraphrase, and an explanation about one of the defined axes and its two directions. We assign a specific value on a Likert scale ranging from one to five, to paraphrase the sentence as if it were spoken by a person using a language with a certain degree indicated by the score. This technique can be extended to involve multiple axes and scores (refer to an example result with two axes in Table 5). The detailed prompts we used are presented in Appendix C.

5 EXPERIMENTS

In this section, we introduce quantitative analysis of our generated NL datasets, lexical analysis on generated utterances, and types of low-level tasks in generated questions.

5.1 Experimental Setup

Our experiment aims to investigate the effectiveness of our framework in generating diverse NL datasets from Vega-Lite specifications, with a focus on accuracy and diversity. To achieve this, we apply tailored metrics to each NL dataset, taking into account their different characteristics. L1/L2 captions are independent of the perception of humans or machines because they focus on conveying objective information [45]. Thus, we measured accuracy to determine how precisely each caption level contained relevant information. We assess the diversity of utterances and questions, as it is important to reflect inclusive language usage among individuals with different background. The results are presented in Table 6 and Table 7 where each type of NL dataset is classified with capital English letter (A-G).

5.1.1 *Benchmarks.* For utterances and questions, we utilized crowdsourced NL datasets gathered in prior studies [32, 79] (F-BM and G-BM). In case of utterance dataset, we only used the singleton case, so it was 804 sentences instead of 893. However, when it comes to captions, we could not find suitable benchmarks for comparing with different caption levels. Previous research employed bitmap images of charts [45, 84], whereas our approach leverages Vega-Lite specifications. This difference in data format prevented us from making an exact comparison.

5.1.2 *Gold Standard Datasets.* Given that benchmarks mostly focus on simple and medium level complexity with confined diversity, we decided to make a gold standard dataset to test the generalizable performance of our framework over diverse and complex charts.

Table 6: Accuracy of the generated chart semantics and L1/L2 captions for 48 sample charts (Figure 2). Although 41 out of the 48 sample charts used in our experiment are complex and extra complex, LLMs were able to capture chart semantics and generate L1/L2 captions successfully in general.

Metadata				Accuracy	
	NL Type (#)	Source	Chart/NL #	w/ Strict criteria	w/ Lenient criteria
A.	Chart Semantics (9)	LLM	48/432	89.4%	96.9%
B.	L1 Caption (1)	LLM	48/48	76.0%	95.8%
C.	L2 Caption (1)	LLM	48/48	76.0%	87.5%

Table 7: Quantitative comparison of benchmarks and LLM-generated utterances and questions. Two type of metrics were adopted, cross-distribution, which is to compare the two distributions to get the similarity and difference, and within-distribution, which is to compare the diversity within a single distribution. Each NL dataset has come from 4 sources, gold standard or benchmarks, LLM, LLM.P (paraphrased), LLM.P2 (paraphrased with 2 axes). The best metric from all sources are bold, while the best metric in ours (LLM, LLM.P, LLM.P2) are underlines.

Metadata				Cross-Distribution			Within-Distribution					
	NL Type (#)	Source	Chart/NL #	FD (↓)	Precision (↑)	Recall (↑)	RC (↑)	Chamfer (↑)	MST (↑)	Span (↑)	Sparsness (↑)	Entropy (↑)
D.	Utterance (3)	Gold	48/144	·	·	·	3.15	**0.19**	47.59	3.26	2.34	**2.60**
		LLM	48/144	0.58	0.81	0.31	3.31	0.19	49.81	3.41	2.48	2.54
		LLM.P		0.45±0.01	0.81±0.01	0.66±0.01	**3.65±0.36**	0.16±0.01	**54.58±3.63**	**3.52±0.21**	**2.70±0.22**	2.09±0.51
		LLM.P2		0.46±0.00	0.80±0.02	0.67±0.03	3.43±0.38	0.16±0.01	52.91±3.11	3.50±0.22	2.49±0.20	2.27±0.35
E.	Question (5)	Gold	48/240	·	·	·	3.47	**0.17**	70.28	3.45	2.64	2.44
		LLM	48/240	0.35	0.84	0.56	**6.20**	0.09	**105.16**	**5.92**	**4.40**	1.68
		LLM.P		0.35±0.00	0.76±0.02	0.64±0.03	4.17±0.20	0.13±0.01	70.00±3.47	4.28±0.30	3.13±0.11	**2.51±0.10**
		LLM.P2		0.36±0.00	0.74±0.02	0.64±0.03	4.29±0.34	0.14±0.01	77.36±5.91	4.44±0.27	3.12±0.19	2.21±0.34
F.	Utterance (3)	BM [79]	30/804	·	·	·	10.42	**0.07**	177.63	10.56	8.41	2.42
		LLM	30/90	·	·	·	·	·	·	·	·	-
		LLM.P	30/804	1.11±0.27	0.63±0.03	0.51±0.05	**12.36±0.18**	0.06±0.00	209.59±7.07	11.74±0.45	9.66±0.38	2.40±0.06
		LLM.P2	30/804	0.87±0.56	0.58±0.04	0.43±0.08	12.24±0.18	0.06±0.00	**227.70±15.66**	**11.86±0.37**	**9.79±0.19**	**2.45±0.07**
G.	Question (4)	BM [32]	52/629	·	·	·	8.66	**0.07**	202.83	11.36	6.12	1.96
		LLM	52/208	·	·	·	·	·	·	·	·	-
		LLM.P	52/619	0.33±0.00	0.52±0.01	0.13±0.01	**11.95±0.27**	0.05±0.00	**247.16±11.72**	12.46±0.49	**9.07±0.32**	**2.41±0.14**
		LLM.P2	52/629	0.33±0.00	0.50±0.01	0.18±0.01	11.61±0.18	0.06±0.00	222.39±8.21	**12.69±0.24**	8.73±0.25	2.39±0.06

We referred to previous works [33, 39] that have demonstrated how to create gold standard datasets. We selected 48 Vega-Lite specifications (Figure 2) by stratified sampling, taking into account their complexity level and whether they included interaction or composite views. Subsequently, three visualization experts (first three authors) collaborated to develop three guidelines for generating utterances and questions. These guidelines were crafted by referring to relevant suggestions and guidelines from prior research [32, 34, 79, 86]. We began by creating sample utterances and questions for the same chart using the initial drafts, and jointly revised each guideline by reviewing the generated NL datasets. After making consensus about the final guidelines Appendix D, we divided the charts into thirds, with each person tasked with generating NL datasets for their assigned charts. This resulted in 48 utterances (comprising 16 commands, 16 queries, and 16 questions) and 80 questions (including 16 non-visual lookup, 16 non-visual compositional, 16 visual lookup, 16 visual compositional, and 16 open-ended questions) per each expert. After one expert created NL datasets for the assigned charts, the other two individuals conducted verification to find any issues or errors within these generated NL datasets. In cases where issues or errors were detected, all three experts convened to discuss and reach a consensus on how to address them. This collaborative effort resulted in the generation of 144 utterances with three different phrasings and 240 questions categorized into five types (D-Gold and E-Gold).

5.1.3 LLM-generated Datasets. To generate our datasets, we used an official API of GPT4[4] with the `gpt-4-0613` model. We set the temperature to 0.0, to solely observe the influence of our paraphrasing technique on diversity. We used different prompt for generating each dataset and paraphrasing the generated NL datasets (see Appendix A and Appendix C). Here, we generated all types of chart semantics, captions, utterances, and questions for the 48 sample charts, as well as all types of utterances for 30 charts from the benchmark. This resulted in a total of 432 chart semantics (A-LLM), 48 L1 captions (B-LLM), 48 L2 captions (C-LLM), 144 utterances (D-LLM), and 240 questions (E-LLM) for the 48 sampled charts, and

[4] https://platform.openai.com/docs/models/gpt-4

Table 8: Types of low-level tasks in questions Top four ranked in both datasets were identical, while LLM-generated dataset has more questions assigned to these ranks.

Low-level Analytical Task	Gold # (%)	LLM # (%)
Retrieve Value	94 (39.2%)	103 (42.9%)
Find Extremum	35 (14.6%)	65 (27.1%)
Correlate	31 (12.9%)	41 (17.1%)
Compute Derived Value	31 (12.9%)	17 (7.1%)
Filter	23 (9.6%)	3 (1.3%)
Find Anomalies	14 (5.8%)	0
Characterize Distribution	5 (2.1%)	4 (1.7%)
Sort	3 (1.3%)	0
Cluster	1 (0.4%)	0
Determine Range	1 (0.4%)	5 (2.1%)
ETC	2 (0.8%)	2 (0.8%)
Sum	240 (100%)	240 (100%)

90 utterances for the 30 benchmark charts (F-LLM). Since the benchmark [32] did not include open-ended questions, we generated only four types of questions. This led to a total of 208 questions for the 52 charts (G-LLM).

We augmented our NL datasets for utterances and questions using the generated NL datasets (*-LLM) and the score-based paraphrasing technique, resulting in augmented paraphrased NL datasets (*-LLM.P and *-LLM.P2). With four language axes and five Likert-scale values (1-5), it is possible to generate 20 different versions (4*5) of paraphrased sentences for each original sentence (i.e., LLM.P). Likewise, in case of two axes, there are six combinations chosen from the four axes. Since there are five Likert-scale options for each axis, this leads to the generation of 150 (6*5*5) different paraphrased sentence versions per original sentence (i.e., LLM.P2). We meticulously generated all possible paraphrases and selected five distinct sets of NL datasets to mitigate any sampling bias. Thus we calculated metrics and their averages and standard deviations across these five sample sets.

When sampling the paraphrased sentences, our goal is to compare the syntactic diversity of different NL datasets while aligning the semantic diversity of the two datasets being compared to ensure a fair comparison. To this end, we adjust the frequency of each chart-NL pair in both datasets. This is necessary because the benchmark data exhibit biases in NL sentence distribution for each chart. For instance, one chart has 30 associated questions, while another chart has only one question. We count the frequency of each chart-NL pair and reflect the same frequency when augmenting the datasets. This became an issue when creating G-LLM.P, since one chart has 30 questions, which exceeds the maximum number of paraphrases possible (limited to 20) through our single-axis paraphrasing method. As a result, our overall number of NL datasets reaches 619.

Last, we included open-ended questions in E-LLM.P and E-LLM.P2, as these questions were available in E-Gold. However, we did not include them in G-LLM.P and G-LLM.P2 in Table 7 to preserve the semantic diversity of the datasets.

5.1.4 Procedure. We manually grade chart semantics and L1/L2 captions to compute their accuracy. To enhance the reliability of our scoring, two experts (the first and second authors) independently scored them and calculated the average score. Specifically, the chart semantics include whether they contain composite views, the type of composite view, the number of plots, chart type, mark, transform, encoding, style, and interaction. We scored whether each of them is correct or not. However, during our evaluation of style, we encountered many cases where multiple width or height values were defined within the Vega-Lite specification. In such cases, we chose to exclude the width and height information from our style evaluation. Moreover, we encountered many cases that were hard to definitively categorize as either correct or incorrect. For instance, situations where nine lines were drawn on the same chart but divided into separate layers, resulting in a count of nine plots instead of one. As a result, we adopted two different scoring approaches, consisting of strict and lenient criteria. Strict criteria only considers those that were 100% accurate. For instance, if a stacked bar chart was categorized as a bar chart, it was deemed incorrect. Conversely, with lenient criteria, we adopted a more flexible approach, considering the aforementioned cases as correct. We extended these criteria to the evaluation of L1/L2 captions as well as their formal definitions [45]. As they contain objective information, we applied the same two criteria and reasoning to assess their accuracy.

To assess the quality of utterances and questions in comparison to both the benchmark and the gold standard dataset, we employ two types of statistical metrics: within-distribution and cross-distribution metrics. The within-distribution metrics are designed to calculate the similarity and divergence between a given dataset and another dataset by means of comparison. Examples of such metrics include Frechet distance (FD), precision, and recall. By utilizing these metrics, we can evaluate how closely a given distribution aligns with the benchmark distribution. These metrics have already been applied in the comparison of human-generated and LLM-generated datasets [21]. To this end, we vectorize the gold standard, benchmarks, and LLM-generated as well as paraphrased datasets, transforming them into sets of vectors for quantitative comparison.

However, we recognize that the aforementioned metrics may not provide a comprehensive measure of the quality of LLM-generated and -paraphrased NL datasets. These metrics mainly focus on the coverage of distribution rather than emphasizing diversity. It is crucial to delve deeper into a single distribution, as duplicate or highly similar data points may be present within it [37, 76]. To address this, we incorporate cross-distribution metrics [66] that allow us to quantify the diversity within a single distribution. These metrics include remote-clique (average of mean pairwise distances), Chamfer distance (average of minimum pairwise distances), MST dispersion (sum of edge weights of MST), span (Pth percentile distance to centroid), sparseness (mean distance to medoid), and entropy (Shannon-Wiener index for points in a grid partition).

5.2 Quantitative Results

We first report the accuracy of chart semantics and L1/L2 captions. Under the strict criteria, the accuracy rates for chart semantics, L1 captions, and L2 captions were 89.4%, 76.0%, and 76.0%, respectively.

In detail, accuracy under strict criteria reveals that 'chart-type' achieved the lowest accuracy at 75%, while 'mark' and 'interaction' showed the highest accuracy at 96.9%. Under lenient criteria, the accuracy rates for chart semantics, L1 captions, and L2 captions significantly improved to 96.9%, 95.8%, and 87.5%, respectively. Specifically, the lowest accuracy for chart semantics was observed in the 'number of plots' (88.5%), while 'mark' and 'interaction' maintained the highest accuracy at 100%. Additionally, the accuracy of chart type substantially improved to 97.9%. A summary of these results is provided in Table 6.

We next report the diversity of utterance and question. In terms of cross-distribution metrics, LLM.P exhibited the highest quality in terms of precision (D), precision and recall (F), and precision and FD (G). In case of datasets containing five question types (E), the metric results were not consistent. Specifically, LLM.P performed the best in FD, LLM was the best for precision, and LLM.P2 achieved the highest recall. When considering within-distribution metrics, LLM-generated and paraphrased datasets demonstrated greater diversity compared to the gold standard and benchmark datasets. On average, higher diversity was observed in 4.75 out of six metrics. For both question and utterance datasets (E, F), paraphrased datasets with two axes demonstrated greater diversity than paraphrased datasets with one axis in four out of six metrics. Conversely, in the other two datasets (D, G), paraphrased datasets with one axis exhibited higher diversity in four out of six metrics. In the utterance dataset (D), paraphrasing increased diversity in four out of six metrics, whereas in the question dataset (E), paraphrasing reduced diversity in four metrics. A summary of the results is presented in Table 7.

5.3 Lexical Analysis in Utterances

To gain a deeper understanding of the syntactic diversity in LLM-generated datasets, we conducted a lexical analysis on three NL datasets (F-BM, F-LLM.P, F-LLM.P2) to investigate the types of words used within each dataset. Our pre-processing steps encompassed sentence tokenization, converting all text to lowercase, removal of stopwords, and lemmatization. As evidenced by the quantitative outcome presented in the previous section, the LLM.P exhibited a notable richness in its lexical diversity. It contained a total of 555 unique words, surpassing the benchmark dataset's count of 349 unique words. Also, the total word count in the LLM.P, amounting to 7,132 words, exceeded that of the benchmark dataset, which consisted of 4,480 words. In case of LLM.P2, it demonstrated an even greater number of unique words, totaling 608, surpassing both the benchmark and LLM.P datasets in this regard. However, the overall word count in LLM.P2 was lower at 6,645 words compared to the LLM.P dataset (7,132 words).

There were some additional patterns in the use of specific words employed within the LLM-generated datasets. First, the paraphrased dataset introduced a multitude of new action verbs. For instance, when issuing commands, terms such as construct, fabricate, organize, and arrange were employed to create charts (e.g., 'Fabricate a line diagram'). In previous work [79], there was a tendency among crowd workers to adhere to specific terminology, thus researchers have to be careful when providing instructions for collecting datasets. Our paraphrasing technique effectively addresses this issue by promoting diverse syntax through the use of various action verbs automatically. Second, the datasets incorporate words that may be adopted by people of specific groups or domains, but not used often by ordinary people, such as domain-specific jargon (e.g., provenance, bifurcated, barometric, pecuniary). Last, certain words have been adopted to introduce diverse tones and voices of the speaker. These encompass terms of a more personal and informal nature, as well as expressions that convey uncertainty and speculation (e.g., maybe, seems, might, quite, sure), as well as words that have been included to enhance conversational aspects (e.g., possibly, would, could).

Table 9: The result of finetuning experiment. The LLMs trained with NL datasets generated by our framework either matched or surpassed the performance of LLMs (C, D, E) compared to when using only the benchmark dataset (A).

	Source	Train #	Test #	Accuracy (#)
A.	BM [79]	723	81	76.3% (61.8)
B.	LLM.P	723	81	58.8% (47.6)
C.	BM + LLM.P	723	81	**76.8% (62.2)**
D.	BM + LLM.P	1446	81	**83.2% (67.4)**
E.	BM + LLM.P + LLM.P2	2169	81	**85.4% (69.2)**

5.4 Types of Low-level Tasks in Questions

Based on a taxonomy [2] comprising ten low-level analytical tasks, we conducted an analysis of the question types present in the gold standard and LLM-generated questions (E-Gold and E-LLM). This analysis aimed to assess the dissimilarities or similarities between these questions. To this end, we associated each low-level analytical task with individual questions within both datasets.

Both datasets exhibited a congruent pattern, with identical rankings for the top four elements. The task with the highest frequency in both datasets is retrieve value, which is unsurprising, as it consists of 40% of lookup questions in the dataset. Notably, in the LLM-generated dataset, the second most prevalent task is find extremum at 27.1%. This percentage closely aligns with Kim et al.'s observation [32], where they reported a similar prevalence of questions related to extrema at 26.7%. Furthermore, it is worth highlighting that, akin to their research, there is a clear bias towards certain task types, including retrieve value, find extremum, correlate, and compute derived value Table 8.

6 APPLICATION

6.1 Finetuning LLMs for Data Visualization

We demonstrate that the NL datasets generated by our framework can be used to augment the performance of ML models. It is important to note, however, that the effectiveness of our datasets is contingent upon the availability of a sufficient number of human-generated datasets that exhibit similar distributions to the test datasets. We believe our approach serves as a cost-effective and efficient way to be used in conjunction with the conventional method of crowdsourcing human-generated NL datasets. In essence, our framework's output can be strategically employed as supplementary datasets for finetuning LLMs.

To be specific, to replicate the benchmark dataset's experiment [79], we performed an experiment to classify ten chart types (e.g., colored scatterplots, stacked & grouped bar charts, multiseries line charts, etc.) using utterances. This classification task is important as it can be further used for building visualization systems like chart type recommendation. We prepared five datasets for finetuning: A. the benchmark dataset (723 utterances), B. the utterances generated and paraphrased with one axis by our framework (723 utterances), C. half of A and half of B (362 from A + 361 from B), D. A and B (723 from A + 723 from B), E. D as well as the utterances generated and paraphrased with two axes by our framework (1446 from D + 723 additional utterances). Only 90% of the benchmark dataset is used for finetuning and the rest 10% were used for the test. Similarly, we used only 90% of our datasets to maintain an equal number of utterances as in the benchmark dataset. Following common ML practices, we selected OpenAI's `babbage-002` model for training smaller models on downstream tasks, setting hyperparameters to default configurations (i.e., number of epochs as 3, learning rate multiplier as 2, and the batch size were 1, 1, 1, 2, 4 for each case). Each experiment was repeated five times to calculate the average accuracy to mitigate the stochastic behavior of LLMs.

As denoted in Table 9, we observed an increase in performance when using LLM-generated NL datasets alongside the benchmark dataset for finetuning the models. Using only the benchmark datasets resulted in an accuracy of 76.3% (61.8 accurate prediction on average out of 81, Table 9-A). When we combined the benchmark datasets with our dataset, the accuracy slightly improved to 76.8% (62.2 out of 81, see Table 9-C), indicating that the addition of a non-human-generated datasets did not negatively impact accuracy. Moreover, the performance increased to 83.2% when we leveraged additional NL datasets generated by LLMs (67.4 out of 81, Table 9-D). The accuracy was the highest when we used more NL datasets paraphrased with two language axes by our framework, which is 85.4% (69.2 out of 81, Table 9-E). Last, using only LLM-generated NL datasets showed decreased accuracy, which is 58.8% (47.6 out of 81, Table 9-B).

The results suggest that using the NL datasets, generated and paraphrased by our framework, can enhance the performance of ML models in downstream tasks. We believe a key factor in this improved performance is the increased syntactic diversity of the generated utterances, which also accurately mimic semantic characteristics. Our results align with a previous finding that utilizing AI-generated datasets can become a more cost-effective strategy for training scalable ML models with significantly fewer human labels [4]. This suggests that the synergistic use of both human efforts and our automated framework can substantially enhance the quality of training data and the performance of the models.

6.2 Leveraging Fully-automatic and Mixed-initiative Modes in VL2NL

To further explore how visualization researchers can use our framework, we performed a case study with two experts: E1, a professor, and E2, a postdoctoral researcher. Both have earned their Ph.D. in visualization and have conducted research for 10 and 7 years, respectively.

For the case study, we implemented a system with two modes: fully-automatic and mixed-initiative (Figure 4). In the fully-automatic mode, the scaffolding is set by us and key questions are generated automatically by the LLMs. Therefore, users had no control, but could click the button to generate NL datasets for their chosen charts. In the mixed-initiative mode, users can select which scaffolding to consider and provide additional information as answers to key questions. They can actively contribute by specifying directions to steer its focus accordingly. For example, in case of L1 caption, they can choose which chart semantics to consider or add more when generating it. Similarly, for question, users can make a high-level decision themselves, specifying where or what to focus on when analyzing the charts.

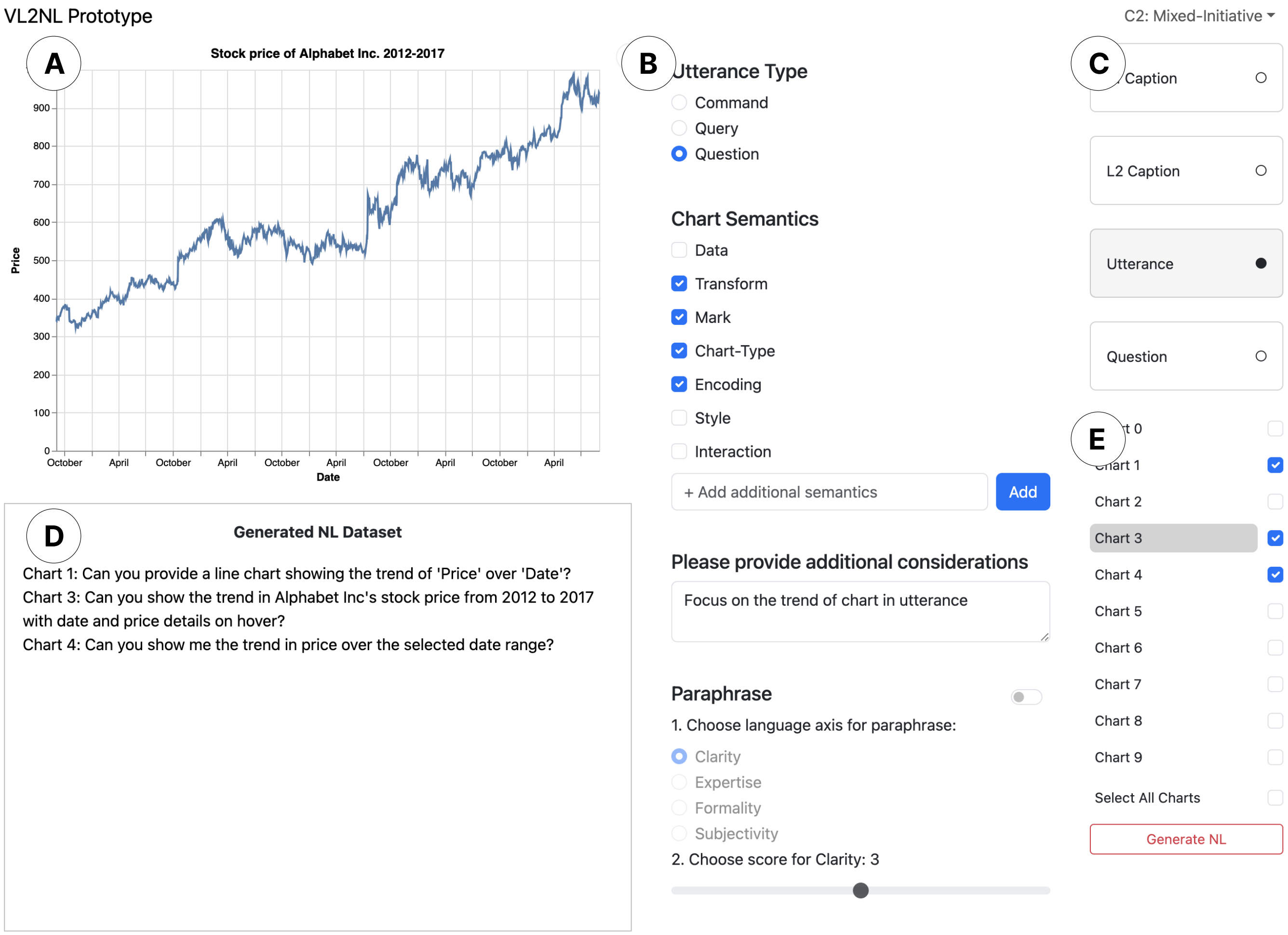
To clarify our study protocol, we first provided the experts with an overview of our framework's concept. Following this, they were given a task to create NL utterances for 10 line charts, which depicted stock prices of various technology companies [92]. This was conducted using two modes: fully-automatic and mixed-initiative. We emphasized to the participants that the utterances they generated would be instrumental in training an ML model to translate these utterances back into the corresponding line charts. We also highlighted the significance of utterance diversity in enhancing the performance of ML models, based on our discussion in (Section 6.1). Finally, the experts provided feedback on their anticipated use of both modes for generating utterances. Each expert spent approximately 45 minutes for the study.

Both experts agreed on using both modes to generate utterances more effectively for training ML models. Specifically, E1 suggested the following scenario: initially, researchers generate a large number of utterances automatically to observe their distribution. Next, they identify areas lacking in diversity, which then become the focus for generating additional utterances subsequently. By repeatedly testing and generating utterances in these sparse areas, particularly using the mixed-initiative mode, they can achieve a more diverse and evenly distributed utterances. This process, iterated over multiple times, could improve the performance of the ML models. Similarly, E2 also advocated starting with the fully-automatic mode before using the mixed-initiative mode. E2 said this approach allows experts to better understand the model's behavior and the nature of the utterances it generates. This step is crucial to avoid 'option paralysis,' a state of cognitive overload that may occur when faced with a lot of choices without a clear strategy for improvement. With a deeper understanding of the model's behavior, they can proceed more effectively.

7 DISCUSSION

7.1 Strengths and Weaknesses of VL2NL

7.1.1 VL2NL Can Guide Itself via Key Questions. We observed several interesting key questions discovered by LLMs that play a vital role in guiding themselves. They were formulated through a meticulous analysis of chart contents. Various decision-making processes were identified, spanning diverse domains such as financial decision-making (e.g., assessing whether to invest in a company's stock), public policy planning (e.g., formulating policies based on employment trends across different age groups and countries), and location-based business strategies (e.g., selecting optimal sites for



Utterance Type

○ Command
○ Query
● Question

Chart Semantics

□ Data
☑ Transform
☑ Mark
☑ Chart-Type
☑ Encoding
□ Style
□ Interaction

+ Add additional semantics Add

Please provide additional considerations

Focus on the trend of chart in utterance

Paraphrase

1. Choose language axis for paraphrase:

● Clarity
○ Expertise
○ Formality
○ Subjectivity

2. Choose score for Clarity: 3

Caption ○
L2 Caption ○
Utterance ●
Question ○

t 0 □
Chart 1 ☑
Chart 2 □
Chart 3 ☑
Chart 4 ☑
Chart 5 □
Chart 6 □
Chart 7 □
Chart 8 □
Chart 9 □
Select All Charts □

Generate NL

Generated NL Dataset

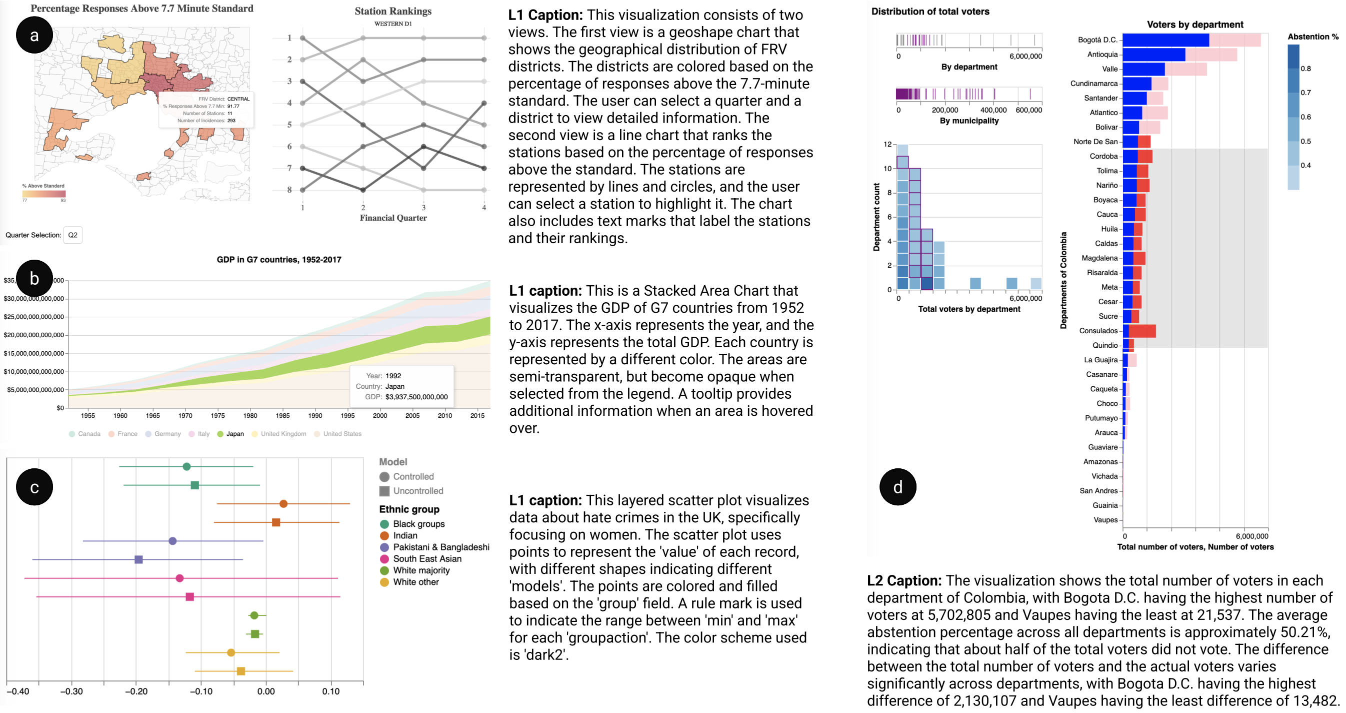
Chart 1: Can you provide a line chart showing the trend of 'Price' over 'Date'?
Chart 3: Can you show the trend in Alphabet Inc's stock price from 2012 to 2017 with date and price details on hover?
Chart 4: Can you show me the trend in price over the selected date range?

Figure 4: A system with two modes (fully-automatic and mixed-initiative) to generate NL datasets using VL2NL. The mixed-initiative mode encompasses several features. First, users can select the types of NL datasets they want to generate (C). They can inspect each chart (A) and subsequently choose the specific ones they wish to use for generating NL datasets (E). Users can change or provide information that the system utilizes (B). Once these are completed, the system returns the generated NL datasets (D). In contrast, the fully-automatic mode does not include (B). As a result, dataset generation in this mode strictly follows the scaffolding defined by researchers, along with key questions and answers generated by LLMs.

a new shoe factory relative to the distribution of existing facilities). These key questions served as the foundation for eliciting subsequent conclusions, retrieving specific values, and deciding mathematical operations for generating interesting questions.

7.1.2 VL2NL Works Robustly on Different Chart Complexity. Considering that the 48 sampled charts mostly belong to the categories of complex and extra-complex charts, our observations indicate that the reported accuracy (Table 6) pertaining to chart semantics and L1/L2 captions does not exhibit a dependence on the complexity levels of the Vega-Lite specifications. This finding underscores the robustness of the system. For instance, it successfully generated an accurate L1 caption for a chart comprising two views interconnected through selection interactions (see Figure 5-a). Similarly, it effectively generated an L2 caption for a chart containing multiple plots, allowing the selection of a data range in the main bar plot to trigger the highlighting of related data points in other plots (see Figure 5-d).

7.1.3 VL2NL Depends Highly on Vega-Lite Specifications. We observed that the framework is highly dependent on the Vega-Lite specifications in generating NL datasets. In many cases, this dependency is advantageous as it enables a focus on intricate functionalities such as interactions. For a particular chart, determining the presence of interactions was challenging because the selection interaction was indicated solely by the color label of the chart. Nevertheless, the framework successfully captured this (see Figure 5-b). Similarly, charts lacking titles or descriptions can pose a challenge in comprehending the content of charts. However, it



L1 Caption: This visualization consists of two views. The first view is a geoshape chart that shows the geographical distribution of FRV districts. The districts are colored based on the percentage of responses above the 7.7-minute standard. The user can select a quarter and a district to view detailed information. The second view is a line chart that ranks the stations based on the percentage of responses above the standard. The stations are represented by lines and circles, and the user can select a station to highlight it. The chart also includes text marks that label the stations and their rankings.

L1 caption: This is a Stacked Area Chart that visualizes the GDP of G7 countries from 1952 to 2017. The x-axis represents the year, and the y-axis represents the total GDP. Each country is represented by a different color. The areas are semi-transparent, but become opaque when selected from the legend. A tooltip provides additional information when an area is hovered over.

L1 caption: This layered scatter plot visualizes data about hate crimes in the UK, specifically focusing on women. The scatter plot uses points to represent the 'value' of each record, with different shapes indicating different 'models'. The points are colored and filled based on the 'group' field. A rule mark is used to indicate the range between 'min' and 'max' for each 'groupaction'. The color scheme used is 'dark2'.

L2 Caption: The visualization shows the total number of voters in each department of Colombia, with Bogota D.C. having the highest number of voters at 5,702,805 and Vaupes having the least at 21,537. The average abstention percentage across all departments is approximately 50.21%, indicating that about half of the total voters did not vote. The difference between the total number of voters and the actual voters varies significantly across departments, with Bogota D.C. having the highest difference of 2,130,107 and Vaupes having the least difference of 13,482.

Figure 5: Four examples of generated L1/L2 captions with corresponding charts. We found that VL2NL can successfully generate captions even on complex charts with varying interactions and multiple views.

appears that the framework was able to extract additional information, even utilizing the URL of the data included in the specification (e.g., `how-did-levels-of-uk-hate-crime-change-during-and-after-covid-19/data/f5.csv`), enabling the generation of informative and coherent captions (see Figure 5-c).

However, we have identified certain cases where relying solely on Vega-Lite specifications proves disadvantageous. First, in some instances, the generated NL datasets include information that was not visually represented in the chart but was present in the Vega-Lite specifications. For instance, additional categories or values that exceed the specified axis range were presented in the NL dataset. Second, if certain information is not explicitly stated within the Vega-Lite specification, it cannot be incorporated into the NL dataset. For example, when generating trellis plots, the number of plots is determined using the unique count of elements. However, since the number is not explicitly provided in the specification, our framework is unable to predict the exact plot number accurately. Last, any errors present in the Vega-Lite specifications are faithfully represented in the generated NL datasets. For instance, a specification contained a typo that divided facets into 3 parts but was mistakenly denoted as 4, our framework predicted the number of plots as 4 instead of the correct 3. Similarly, the Vega-Lite specification included code that is non-functional, which is reflected in the generated captions, resulting in inaccuracies.

7.1.4 VL2NL Predicts Only Common Chart Types. Our categorization relied on the chart type taxonomy presented by Borkin et al. [6], which led to different categorizations even when the same mark was used. For instance, although the same point mark was employed, it could be interpreted as either a distribution chart (e.g., dot array) or a scatter plot. Furthermore, we conducted an analysis of the charts by considering their detailed sub-types rather than grouping them into larger categories. For instance, a chart featuring a stacked area chart was considered incorrect according to strict criteria if it was predicted as an area chart. However, we observed that in most cases, the LLM framework tended to assign the charts to the most prevalent and common chart types such as scatterplots, area charts, and bar charts, rather than classifying them as distribution charts, stacked area charts, or stacked bar charts (Figure 5-c). With this reasons, the chart type exhibited the highest accuracy gap between strict and lenient criteria.

7.2 Limitations and Future Work

7.2.1 Enriching Capabilities of VL2NL through External Resources. While Vega-Lite specifications serve as powerful inputs for generating various types of NL datasets, it is inherently challenging to extract information that does not exist within these specifications. Although our framework can operate in both fully-automatic and mixed-initiative manner, it does not rely on external resources. This limitation can potentially impact the performance of NL generation, as it aligns with observations in guided discovery, where insufficient prior knowledge can hinder learners from formulating hypotheses, interpreting data effectively, and engaging in systematic experimentation [15]. To enhance the capabilities of VL2NL, we suggest accessing external information to guide the process during NL dataset generation. For instance, generating L3/L4 captions often necessitates access to common or domain-specific knowledge [45]. In this regard, employing tools like ReAct [94] becomes advantageous, as it facilitates reasoning to assist the model in deducing, tracking, and updating action plans while also handling exceptions. This enables us to proactively retrieve information from the web when required.

7.2.2 Augmenting Vega-Lite Specifications. While we have presented the largest amount of Vega-Lite specifications and acknowledge their ability as input for generating diverse NL datasets, it is noteworthy that the quantity of Vega-Lite specifications is significantly smaller compared to bitmap images. This is mainly because collecting Vega-Lite specifications is more challenging when compared to other formats. This limitation hinders the effective training or fine-tuning of machine learning models to achieve robust performance. Consequently, we posit the need for methods to augment Vega-Lite specifications. Various augmentation techniques have been introduced and adopted for bitmap images of charts to increase both their quantity [29] and diversity [98]. However, to the best of our knowledge, we have not found any pertinent research that addresses the augmentation of Vega-Lite specifications. As part of our future work, we aim to tackle this gap by developing a reverse engineering technique [63] specifically designed for Vega-Lite specifications.

7.2.3 Covering Additional NL datasets. Our framework exhibits potential for generalization across multiple NL datasets. However, we recognize that it covers only limited number of types, which we aim to expand in our future research. Specifically, we plan to create conversational NL datasets to facilitate interactive communication with NLIs, given the growing significance of conversational agents. We believe a dataset based on deeper analysis of users' conversational characteristics will be immensely beneficial for researchers. We also plan to address reference datasets linking charts with text to help make interactive documents. We believe this will make the connection between them clearer, and the reading experience more enjoyable and engaging.

8 CONCLUSION

We introduce VL2NL designed to generate diverse NL datasets aimed at enhancing NLIs for data visualization research. Our framework takes a Vega-Lite specification as input and employs guided discovery to accurately generate various NL datasets, including captions, utterances, and questions. We also propose a score-based paraphrasing approach to enhance the syntactic diversity of the generated NL datasets. We also present a dataset comprising 1,981 Vega-Lite specifications. This dataset surpasses the baselines in terms of complexity and diversity. Our experimental results substantiate that the framework excels in accurately generating both L1 and L2 captions, while achieving higher diversity in the generation of utterances and questions compared to the baselines. Last, we introduce real-world scenarios of using LLM-generated NL datasets and our framework. We hope our framework and chart collection can advance research in developing NLIs for data visualization.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2023R1A2C200520911). This work was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)).

REFERENCES

[1] Rinat Abdrashitov, Fanny Chevalier, and Karan Singh. 2020. Interactive Exploration and Refinement of Facial Expression Using Manifold Learning. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 778–790. https://doi.org/10.1145/3379337.3415877

[2] Robert Amar, James Eagan, and John Stasko. 2005. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* IEEE, 111–117.

[3] Toshiki Aoki, Rintaro Chujo, Katsufumi Matsui, Saemi Choi, and Ari Hautasaari. 2022. EmoBalloon-Conveying Emotional Arousal in Text Chats with Speech Balloons. In *CHI Conference on Human Factors in Computing Systems.* 1–16.

[4] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073* (2022).

[5] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology.* 313–322.

[6] Michelle A Borkin, Azalea A Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. 2013. What makes a visualization memorable? *IEEE transactions on visualization and computer graphics* 19, 12 (2013), 2306–2315.

[7] Ann L Brown and Joseph C Campione. 1994. *Guided discovery in a community of learners.* The MIT Press.

[8] Stuart K Card, Jock Mackinlay, and Ben Shneiderman. 1999. *Readings in information visualization: using vision to think.* Morgan Kaufmann.

[9] Harrison Chase. 2022. *LangChain.* https://github.com/hwchase17/langchain

[10] Ritwick Chaudhry, Sumit Shekhar, Utkarsh Gupta, Pranav Maneriker, Prann Bansal, and Ajay Joshi. 2020. Leaf-qa: Locate, encode & attend for figure question answering. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 3512–3521.

[11] Chen Chen and Zhicheng Liu. 2023. The State of the Art in Creating Visualization Corpora for Automated Chart Analysis. *Computer Graphics Forum* (2023). https://doi.org/10.1111/cgf.14855

[12] Xi Chen, Wei Zeng, Yanna Lin, Hayder Mahdi Ai-Maneea, Jonathan Roberts, and Remco Chang. 2020. Composition and configuration patterns in multiple-view visualizations. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1514–1524.

[13] Kenneth Cox, Rebecca E Grinter, Stacie L Hibino, Lalita Jategaonkar Jagadeesan, and David Mantilla. 2001. A multi-modal natural language interface to an information visualization environment. *International Journal of Speech Technology* 4 (2001), 297–314.

[14] Kenny Davila, Srirangaraj Setlur, David Doermann, Bhargava Urala Kota, and Venu Govindaraju. 2020. Chart mining: A survey of methods for automated chart analysis. *IEEE transactions on pattern analysis and machine intelligence* 43, 11 (2020), 3799–3819.

[15] Ton De Jong and Wouter R Van Joolingen. 1998. Scientific discovery learning with computer simulations of conceptual domains. *Review of educational research* 68, 2 (1998), 179–201.

[16] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology.* 845–854.

[17] Victor Dibia. 2023. LIDA: A Tool for Automatic Generation of Grammar-Agnostic Visualizations and Infographics using Large Language Models. *arXiv preprint arXiv:2303.02927* (2023).

[18] Victor Dibia and Çağatay Demiralp. 2019. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE computer graphics and applications* 39, 5 (2019), 33–46.

[19] Siwei Fu, Kai Xiong, Xiaodong Ge, Siliang Tang, Wei Chen, and Yingcai Wu. 2020. Quda: natural language queries for visual data analytics. *arXiv preprint arXiv:2005.03257* (2020).

[20] 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. 489–500.

[21] Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. 2023. Evaluating large language models in generating synthetic hci research data: a case study. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–19.

[22] Jonathan Harper and Maneesh Agrawala. 2017. Converting basic D3 charts into reusable style templates. *IEEE transactions on visualization and computer graphics* 24, 3 (2017), 1274–1286.

[23] Cindy E Hmelo-Silver, Ravit Golan Duncan, and Clark A Chinn. 2007. Scaffolding and achievement in problem-based and inquiry learning: a response to Kirschner, Sweller, and. *Educational psychologist* 42, 2 (2007), 99–107.

[24] Enamul Hoque, Parsa Kavehzadeh, and Ahmed Masry. 2022. Chart question answering: State of the art and future directions. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 555–572.

[25] Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. 2017. Applying pragmatics principles for interaction with visual analytics. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 309–318.

[26] Kevin Hu, Diana Orghian, and César Hidalgo. 2018. DIVE: A mixed-initiative system supporting integrated data exploration workflows. In *Proceedings of the workshop on human-in-the-loop data analytics*. 1–7.

[27] Maeve Hutchinson, Aidan Slingsby, Radu Jianu, and Pranava Madhyastha. 2023. Towards Visualisation Specifications from Multilingual Natural Language Queries using Large Language Models. In *EuroVis 2023 - Posters*, Christina Gillmann, Michael Krone, and Simone Lenti (Eds.). The Eurographics Association. https://doi.org/10.2312/evp.20231072

[28] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* 55, 12 (2023), 1–38.

[29] Daekyoung Jung, Wonjae Kim, Hyunjoo Song, Jeong-in Hwang, Bongshin Lee, Bohyoung Kim, and Jinwook Seo. 2017. Chartsense: Interactive data extraction from chart images. In *Proceedings of the 2017 chi conference on human factors in computing systems*. 6706–6717.

[30] Sean Kandel, Ravi Parikh, Andreas Paepcke, Joseph M Hellerstein, and Jeffrey Heer. 2012. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. 547–554.

[31] Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. Chart-to-text: A large-scale benchmark for chart summarization. *arXiv preprint arXiv:2203.06486* (2022).

[32] Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. 2020. Answering questions about charts and generating visual explanations. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.

[33] Dae Hyun Kim, Enamul Hoque, Juho Kim, and Maneesh Agrawala. 2018. Facilitating document reading by linking text and tables. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 423–434.

[34] Dae Hyun Kim, Vidya Setlur, and Maneesh Agrawala. 2021. Towards understanding how readers integrate charts and captions: A case study with line charts. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11.

[35] Robert Kincaid and Graham Pollock. 2017. Nicky: Toward a virtual assistant for test and measurement instrument recommendations. In *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*. IEEE, 196–203.

[36] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 43–52.

[37] Mark Klein and Ana Cristina Bicharra Garcia. 2015. High-speed idea filtering with the bag of lemons. *Decision Support Systems* 78 (2015), 39–50.

[38] Hyung-Kwon Ko, Subin An, Gwanmo Park, Seung Kwon Kim, Daesik Kim, Bohyoung Kim, Jaemin Jo, and Jinwook Seo. 2022. We-toon: A Communication Support System between Writers and Artists in Collaborative Webtoon Sketch Revision. In *The 35th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[39] Nicholas Kong, Marti A Hearst, and Maneesh Agrawala. 2014. Extracting references between text and charts via crowdsourcing. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 31–40.

[40] Yixuan Li, Yusheng Qi, Yang Shi, Qing Chen, Nan Cao, and Siming Chen. 2022. Diverse interaction recommendation for public users exploring multi-view visualization using deep learning. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 95–105.

[41] Zhenyu Li, Sunqi Fan, Yu Gu, Xiuxing Li, Zhichao Duan, Bowen Dong, Ning Liu, and Jianyong Wang. 2023. FlexKBQA: A Flexible LLM-Powered Framework for Few-Shot Knowledge Base Question Answering. *arXiv preprint arXiv:2308.12060* (2023).

[42] Can Liu, Yun Han, Ruike Jiang, and Xiaoru Yuan. 2021. Advisor: Automatic visualization answer for natural-language question on tabular data. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*. IEEE, 11–20.

[43] Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2022. DePlot: One-shot visual language reasoning by plot-to-table translation. *arXiv preprint arXiv:2212.10505* (2022).

[44] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. 2022. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. *arXiv preprint arXiv:2212.09662* (2022).

[45] Alan Lundgard and Arvind Satyanarayan. 2021. Accessible visualization via natural language descriptions: A four-level model of semantic content. *IEEE transactions on visualization and computer graphics* 28, 1 (2021), 1073–1083.

[46] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. Deepeye: Towards automatic data visualization. In *2018 IEEE 34th international conference on data engineering (ICDE)*. IEEE, 101–112.

[47] 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 the 2021 International Conference on Management of Data*. 1235–1247.

[48] Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. 2021. Natural language to visualization by neural machine translation. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 217–226.

[49] Jock Mackinlay. 1986. Automating the design of graphical presentations of relational information. *Acm Transactions On Graphics (Tog)* 5, 2 (1986), 110–141.

[50] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show me: Automatic presentation for visual analysis. *IEEE transactions on visualization and computer graphics* 13, 6 (2007), 1137–1144.

[51] Anita Mahinpei, Zona Kostic, and Chris Tanner. 2022. LineCap: Line Charts for Data Visualization Captioning Models. In *2022 IEEE Visualization and Visual Analytics (VIS)*. IEEE, 35–39.

[52] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244* (2022).

[53] Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* 2, 11 (2017), 205.

[54] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[55] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems* 35 (2022), 462–477.

[56] Valerie S Morash, Yue-Ting Siu, Joshua A Miele, Lucia Hasty, and Steven Landau. 2015. Guiding novice web workers in making image descriptions using templates. *ACM Transactions on Accessible Computing (TACCESS)* 7, 4 (2015), 1–21.

[57] Dominik Moritz, Chenglong Wang, Greg L Nelson, Halden Lin, Adam M Smith, Bill Howe, and Jeffrey Heer. 2018. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 438–448.

[58] Mohammad Amin Mozaffari, Xinyuan Zhang, Jinghui Cheng, and Jin LC Guo. 2022. GANSpiration: Balancing Targeted and Serendipitous Inspiration in User Interface Design with Style-Based Generative Adversarial Network. In *CHI Conference on Human Factors in Computing Systems*. 1–15.

[59] Arpit Narechania, Adam Fourney, Bongshin Lee, and Gonzalo Ramos. 2021. DIY: Assessing the correctness of natural language to sql systems. In *26th International Conference on Intelligent User Interfaces*. 597–607.

[60] Arpit Narechania, Arjun Srinivasan, and John Stasko. 2020. NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 369–379.

[61] Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. *arXiv preprint arXiv:2010.09142* (2020).

[62] Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442* (2023).

[63] Jorge Poco and Jeffrey Heer. 2017. Reverse-engineering visualizations: Recovering visual encodings from chart images. In *Computer graphics forum*, Vol. 36. Wiley Online Library, 353–363.

[64] Xin Qian, Eunyee Koh, Fan Du, Sungchul Kim, Joel Chan, Ryan A Rossi, Sana Malik, and Tak Yeon Lee. 2021. Generating accurate caption units for figure captioning. In *Proceedings of the Web Conference 2021*. 2792–2804.

[65] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[66] Samuel Rhys Cox, Yunlong Wang, Ashraf Abdul, Christian Von Der Weth, and Brian Y. Lim. 2021. Directed diversity: Leveraging language embedding distances for collective creativity in crowd ideation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–35.

[67] Andy Rosenbaum, Saleh Soltan, Wael Hamza, Amir Saffari, Macro Damonte, and Isabel Groves. 2022. CLASP: Few-shot cross-lingual data augmentation for

semantic parsing. *arXiv preprint arXiv:2210.07074* (2022).

[68] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization & Computer Graphics (Proc. InfoVis)* (2017). https://doi.org/10.1109/tvcg.2016.2599030

[69] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2023. *Vega Editor*.

[70] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2023. *Vega-Lite gallery*.

[71] Arvind Satyanarayan, Ryan Russell, Jane Hoffswell, and Jeffrey Heer. 2015. Reactive vega: A streaming dataflow architecture for declarative interactive visualization. *IEEE transactions on visualization and computer graphics* 22, 1 (2015), 659–668.

[72] Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. *arXiv preprint arXiv:2104.07540* (2021).

[73] 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*. 365–377.

[74] Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai, and Jianmin Wang. 2022. Towards natural language interfaces for data visualization: A survey. *IEEE transactions on visualization and computer graphics* (2022).

[75] Leixian Shen, Yizhi Zhang, Haidong Zhang, and Yun Wang. 2023. Data player: Automatic generation of data videos with narration-animation interplay. *arXiv preprint arXiv:2308.04703* (2023).

[76] Pao Siangliulue, Joel Chan, Steven P Dow, and Krzysztof Z Gajos. 2016. IdeaHound: improving large-scale collaborative ideation with crowd-powered real-time semantic modeling. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 609–624.

[77] Andrea Spreafico and Giuseppe Carenini. 2020. Neural data-driven captioning of time-series line charts. In *Proceedings of the International Conference on Advanced Visual Interfaces*. 1–5.

[78] Arjun Srinivasan, Steven M Drucker, Alex Endert, and John Stasko. 2018. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 672–681.

[79] 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 the 2021 CHI Conference on Human Factors in Computing Systems*. 1–10.

[80] Arjun Srinivasan and John Stasko. 2017. Natural language interfaces for data analysis with visualization: Considering what has and could be asked. In *Proceedings of the Eurographics/IEEE VGTC conference on visualization: Short papers*. 55–59.

[81] Arjun Srinivasan and John Stasko. 2017. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 511–521.

[82] Chris Stolte, Diane Tang, and Pat Hanrahan. 2002. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (2002), 52–65.

[83] Nicole Sultanum and Arjun Srinivasan. 2023. DataTales: Investigating the use of Large Language Models for Authoring Data-Driven Articles. *arXiv preprint arXiv:2308.04076* (2023).

[84] Benny J. Tang, Angie Boggust, and Arvind Satyanarayan. 2023. VisText: A Benchmark for Semantically Rich Chart Captioning. In *The Annual Meeting of the Association for Computational Linguistics (ACL)*. http://vis.csail.mit.edu/pubs/vistext

[85] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. 2015. Seedb: Efficient data-driven visualization recommendations to support visual analytics. In *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, Vol. 8. NIH Public Access, 2182.

[86] Yun Wang, Zhitao Hou, Leixian Shen, Tongshuang Wu, Jiaqi Wang, He Huang, Haidong Zhang, and Dongmei Zhang. 2022. Towards Natural Language-Based Visualization Authoring. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 1222–1232.

[87] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[88] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2015. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics* 22, 1 (2015), 649–658.

[89] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Towards a general-purpose query language for visualization recommendation. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–6.

[90] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2017. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 chi conference on human factors in computing systems*. 2648–2659.

[91] Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. 2023. WebUI: A Dataset for Enhancing Visual UI Understanding with Web Semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.

[92] Yumo Xu and Shay B Cohen. 2018. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1970–1979.

[93] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics* 10 (2022), 291–306.

[94] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*.

[95] Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *arXiv preprint arXiv:2202.07922* (2022).

[96] Jiacheng Ye, Chengzu Li, Lingpeng Kong, and Tao Yu. 2023. Generating Data for Symbolic Language with Large Language Models. *arXiv preprint arXiv:2305.13917* (2023).

[97] Bowen Yu and Cláudio T Silva. 2019. FlowSense: A natural language interface for visual data exploration within a dataflow system. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 1–11.

[98] Jian Zhao, Mingming Fan, and Mi Feng. 2020. Chartseer: Interactive steering exploratory visual analysis with machine intelligence. *IEEE Transactions on Visualization and Computer Graphics* 28, 3 (2020), 1500–1513.

A PROMPTS FOR NL GENERATION

In the prompt, certain variables are enclosed within curly brackets. We colored them blue for easy recognition. Here, we provide a detailed explanation of each variable and specify its usage in different NL generation prompts:

- `vl` [all]: Minified Vega-Lite specification;
- `ftt_str` [L1/L2 caption, utterance]: Information about fields, titles, types, and values;
- `prompt` [L2 caption]: Questions derived from the guided discovery process;
- `info` [L2 caption]: Answers for the questions computed through LangChain library [9];
- `info_first_concat` [utterance]: A list of primary instructions by analyzing the chart semantics.

A.1 L1 Caption

```
{vl}

Let's generate a level 1 NL description step by step.

Step 1. Determine if the visualization contains
composite views, such as layered plots, trellis plots,
or other types of multiple view displays, and provide
a count of the number of plots if any are present.
Step 2. Analyze the semantics of each chart
individually, including [Data], [Transform], [Mark],
[Chart-Type], [Encoding], [Style], and [Interaction].
Refer to this:
{ftt_str}
Step 3. Generate a level 1 NL description using the
semantics. It contains elemental and encoded
properties of the visualization (i.e., the visual
components that comprise a graphical representation's
design and construction).
```

```

##
Step 1. Composite Views:
- True/False:
- (If True) Type: (layered, trellis, multiple views)
- Number of plots:
Step 2. Chart Semantics:
- Data:
- Field (Value):
- Transform:
- Mark:
- Chart-Type:
- Encoding:
- Style:
- Interaction (e.g., tooltip):
Step 3. Level 1 NL Description:
```

A.2 L2 Caption

```
{vl}

Let's generate question(s) step by step.

Step 1. What is the most prominent and meaningful feature in the given chart?
Step 2. What is the mathematical operation(s) (e.g., max, min, sum, difference, and average) required to describe the feature?
Step 3. Generate question(s) using the mathematical operation(s) required to describe the feature. If there are multiple questions, separate them with semicolon(;).

##
Step 1. Features:
Step 2. Operations:
Step 3. Questions:
```

```
Refer to this: {ftt_str}
Do not draw any charts to answer the question.

Question: {prompt}
```

```
Information: {info}

{ftt_str}

Generate a concise level 2 NL description of a visualization, with 1 or 2 sentences. It contains statistical and relational properties of the visualization (e.g., descriptive statistics, extrema, outliers, correlations, point-wise comparisons).

##
Level 2 NL Description:
```

A.3 Utterance

```
{vl}

Step 1. Determine if the visualization contains composite views, such as layered plots, trellis plots, or other types of multiple view displays, and provide a count of the number of plots if any are present.
Step 2. Provide a list of instructions to create the chart using natural language.
- Write instructions for each view and separate with <%>
- Separate each instruction by a semicolon (;)
- Divide each instruction to contain only one specific action
- Use the following chart semantics to specify instructions: [Data], [Chart-Type], [Mark], [Encoding], [Transform], [Style], [Interaction]
Step 3. Given the information about the fields and their synonyms, please replace the field names with their corresponding synonyms.
{ftt_str}

##
Step 1. Composite Views:
- True/False:
- (If True) Type: (layered, trellis, multiple views)
- Number of plots:
Step 2. Instructions:
[View #]; [<Chart Semantic>]: <Instruction>; [<Chart Semantic>]: <Instruction>; ... <%>
Step 3. Instructions:
[View #]; [<Chart Semantic>]: <Instruction>; [<Chart Semantic>]: <Instruction>; ... <%>
```

```
{inst_first_concat}
The above are instructions to generate a chart. Let's generate combined instructions ([Command], [Query], [Question]) for each view step by step.

Step 1. Identify the primary information in each view.
Step 2. Identify the secondary information in each view.
Step 3. Generate a [Command] for each view using only the primary info. Please follow these rules:
- Use imperative voice
- Write in a single sentence
- Use only the primary info
- Make it concise and simple
Step 4. Generate a [Query] for each view using only the primary info. Please follow these rules:
- Refrain from using verbs and articles (e.g., a, the)
- Use only variables, fields, attributes, mathematical formulas (e.g., sum, avg, mix, max, count, order), abbreviations (e.g., vs), and prepositions (e.g., of, by, for, with, over, from, to)
- Write in a single sentence
- Use only the primary info
- Make it concise and simple
Step 5. Generate a [Question] for each view using only the primary info. Please follow these rules:
- Ask an inquiry as a question
- Write in a single sentence
- Use only the primary info
- Make it concise and simple

```

```
##
View #<Number>:
Step 1. Primary Information:
Step 2. Secondary Information:
Step 3. Command:
Step 4. Query:
Step 5. Question:
```

A.4 Question

```
{vl}

Let's generate a lookup question, a compositional
question, and an open-ended question for a given
Vega-Lite spec step by step. The lookup question
requires a single value retrieval. The compositional
question requires multiple operations.

Step 1. What higher-level decision can be made by
analyzing this chart?
Step 2. What is a possible conclusion that can be
reached from this decision?
Step 3. What specific value can be retrieved to reach
this conclusion?
Step 4. Generate a lookup question using this value,
without including any visual attributes such as color,
length, size, or position.
Step 5. What visual attributes are required to
paraphrase this question?
Step 6. Paraphrase the generated question using the
chart's visual attributes.
Step 7. What are the mathematical operations (e.g.,
max, min, sum, difference, and average) to reach the
conclusion in Step 2?
Step 8. Generate a compositional question using these
operations, without including any visual attributes
such as color, length, size, or position.
Step 9. What visual attributes are required to
paraphrase this question?
Step 10. Paraphrase the generated question using the
chart's visual attributes.
Step 11. Generate an open-ended question to reach the
conclusion in Step 2.

##
Step 1. Decision:
Step 2. Conclusion:
Step 3. Specific Value:
Step 4. Lookup Question:
Step 5. Visual Attributes:
Step 6. Paraphrased Question:
Step 7. Operations:
Step 8. Compositional Question:
Step 9. Visual Attributes:
Step 10. Paraphrased Question:
Step 11. Open-ended Question:
```

B PROMPT FOR AUTOMATIC QUALITATIVE CODING

When extracting codes, we omitted the words 'language' and 'use of' since they were frequently added to the code. We believe that these additions do not contribute any additional meaning to the thematic analysis.

```
Let's perform a thematic analysis in the field of
human-computer interaction. Generate characteristics
of languages leveraged in the given sentence. The
total number is five and each of them is separated by
semicolons. Do not add numbering or any explanations.

Sentence: {sent}

##
; ; ; ;
```

C PROMPTS FOR SCORE-BASED PARAPHRASING

We explain the variables used in our prompts:

- `Example Sentence`: A sentence we want to paraphrase;
- `Axis`: An explanation about each of the defined language axes;
- `Direction`: A set of two opposite directions of the given language axis;
- `Score`: A specific value on a Likert-scale ranging from one to five assigned to each of the language axis.

C.1 Paraphrasing with one axis

```
{Axis}

Score of 1 indicates a higher tendency to use
{Direction-1} language and a Score of 5 indicates a
higher tendency to use {Direction-2} language. Rewrite
the following sentence as if it were spoken by a
person with a given score for language usage.

Sentence: {Example Sentence}
Score: {Score}
```

C.2 Paraphrasing with two axes

```
{Axis-1}
{Axis-2}

Score-A of 1 indicates a higher tendency to use
{Direction-1-1} language and a Score-A of 5 indicates
a higher tendency to use {Direction-1-2} language.
Score-B of 1 indicates a higher tendency to use
{Direction-2-1} language and a Score-B of 5 indicates
a higher tendency to use {Direction-2-2} language.
Rewrite the following sentence as if it were spoken by
a person with a given score for language usage.

Sentence: {Example Sentence}
Score-A: {Score-A}, Score-B: {Score-B}
```

D GOLD REFERENCE GUIDELINES

D.1 Utterance

- Imagine writing utterances to display a visualization using a system like Excel, Tableau, or Microsoft Power BI;
- Refer to both the dataset and the chart to better understand the context in which the data has been used for the visualization and formulate more naturalistic utterances.
- Avoid referring to specific instructions to prevent acclimatization to the words or phrases in the instruction [79];
- Write utterances as singletons, which are basic types of utterances, but can be more than one sentence if necessary due to complexity, forming a sequential utterance that provides all necessary information;
- Write utterance for each view. If the chart is has layered plots, and they have different chart types, write utterance with according to the number of different chart types;
- Focus on primary information such as chart type and encoding rather than secondary information such as style and interaction [86].

D.2 Question

- Ask one question in one complete sentence;
- Keep questions clear and concise, avoiding overly broad or vague questions by focusing on specific aspects of the chart;
- Formulate questions that can elicit useful insights from the visualization to facilitate visual data analysis and decision-making [32].