ChartInstruct: Instruction Tuning for Chart Comprehension and Reasoning

Ahmed Masry[♣,*], **Mehrad Shahmohammadi**[♣,*],
Md Rizwan Parvez[♥], **Enamul Hoque**[♣], **Shafiq Joty**[♦♠]
[♣]York University, Canada, [♥]Qatar Computing Research Institute (QCRI)
[♦]Salesforce Research, [♠]Nanyang Technological University, Singapore
{msm97, enamulh}@yorku.ca, mparvez@hbku.edu.qa
sjoty@salesforce.com, ahmed.elmasry24653@gmail.com

Abstract

Charts provide visual representations of data and are widely used for analyzing information, addressing queries, and conveying insights to others. Various chart-related downstream tasks have emerged recently, such as question-answering and summarization. A common strategy to solve these tasks is to fine-tune various models originally trained on vision tasks language. However, such task-specific models are not capable of solving a wide range of chart-related tasks, constraining their real-world applicability. To overcome these challenges, we introduce *ChartInstruct*: a novel chart-specific vision-language Instruction-following dataset comprising 191K instructions generated with 71K charts. We then present two distinct systems for instruction tuning on such datasets: (1) an end-to-end model that connects a vision encoder for chart understanding with a LLM; and (2) a pipeline model that employs a two-step approach to extract chart data tables and input them into the LLM. In experiments on four downstream tasks, we first show the effectiveness of our model–achieving a new set of state-of-the-art results. Further evaluation shows that our instruction-tuning approach supports a wide array of real-world chart comprehension and reasoning scenarios, thereby expanding the scope and applicability of our models to new kinds of tasks.

1 Introduction

Information visualizations, such as bar charts and line charts, play a pivotal role in data analysis, offering critical insights and aiding in informed decision-making (Hoque et al., 2022). However, discerning key patterns and trends from these visualizations and addressing complex queries can pose significant challenges. Recent research has introduced various tasks to assist users in chart analysis, including chart question answering (Masry et al., 2022a; Kantharaj et al., 2022b), summarizing insights from visualizations (Obeid and Hoque, 2020a; Shankar et al., 2022), reasoning over chart images, fact-checking (Akhtar et al., 2023a), and automated visual data storytelling (Shi et al., 2020).

Early work attempts to tackle these tasks by fine-tuning models originally trained on language and vision tasks (Raffel et al., 2020; Masry et al., 2022a; Lee et al., 2022; Kantharaj et al., 2022c). However, such models may not be optimal for chart-specific tasks as they overlook explicit modeling of chart structures such as relationships between chart elements like bars, legends, and axes. Recent models such as UniChart (Masry et al., 2023), Chart-T5 (Zhou et al., 2023), and MatCha (Liu et al., 2022b) are specifically designed for charts by considering visual and mathematical reasoning over chart elements and values. However, they often consider charts from a limited range of sources and focus on a narrow set of tasks, constraining their real-world applicability. Indeed, for real-world widespread adoption, we cannot presume how and on what tasks these models will be used.

A promising solution to this challenge is *instruction tuning*, as demonstrated by language models like InstructGPT (Ouyang et al., 2022a), FLAN-T5 (Chung et al., 2022), Alpaca (Alpaca, 2023), (Chiang et al., 2023) and LLaMA-chat (Touvron et al., 2023). They show that training LLMs on instruction-following datasets significantly enhances their alignment with user intent across various tasks, including tasks that are unseen during training. Recent advances in vision-language tasks (Li et al., 2023a; Dai et al., 2023) have adopted similar methodologies, fine-tuning vision-language models (VLMs) with visual instructions to better match user intentions and improve efficacy. However, to our knowledge, instruction tuning for chart comprehension and reasoning remains underexplored. Existing methods (Liu et al., 2023a; Han et al., 2023), which are concurrent to our work,

[*]Equal contribution.

Findings of the Association for Computational Linguistics: ACL 2024, pages 10387–10409
August 11-16, 2024 ©2024 Association for Computational Linguistics

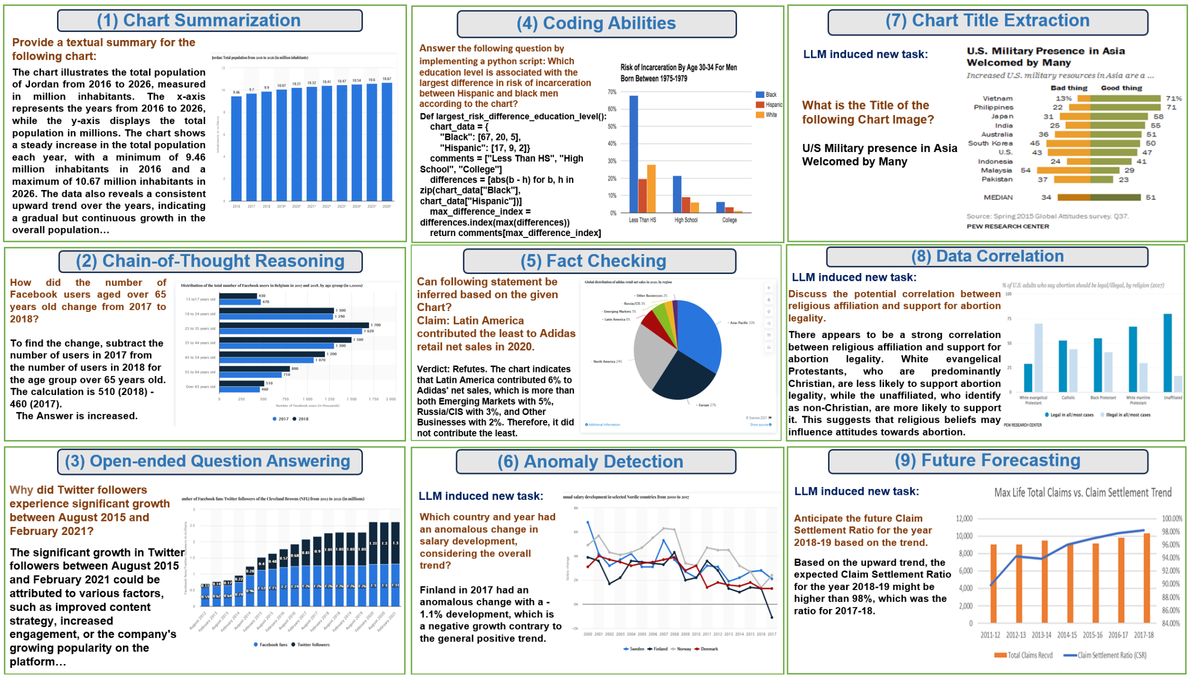


Figure 1: Examples of different chart-related tasks from our generated instruction dataset. Examples 1-5 are generated based on predefined tasks similar to previously developed downstream tasks like chart summarization, chart question answering, and fact-checking, while examples 6-9 introduce new types of tasks distilled by LLMs.

lack variety and scope in instruction-tuning tasks, limiting their effectiveness in real-world chart understanding scenarios.

In this paper, we introduce Chart Instruction Tuning (*ChartInstruct*), to pave the way towards building general-purpose chart comprehension and reasoning assistant based on VLMs. To this end, we have developed a new chart instruction-tuning dataset featuring real-world charts collected from 157 online platforms, covering wide and diverse visual styles. Leveraging advanced LLMs such as GPT-3.5 (OpenAI-Blog, 2022) , GPT-4 (OpenAI, 2023), and Gemini (Team et al., 2023)), we generate **191K** instructions covering a broad array of tasks reflecting real-world applications (Figure 1).

As charts are unique and pose challenges distinct from general multi-modal data, a structured approach is crucial for enabling VLMs to effectively leverage the instruction dataset, optimizing their performance in chart analysis tasks. We introduce two innovative VLM designs in this regard. The first system modifies the LLaVA architecture (Li et al., 2023a), substituting its CLIP vision encoder with the UniChart (Masry et al., 2023) vision encoder pre-trained specifically on chart images. For language modeling in this design, we experiment with two different models: Llama2 (7B) – a decoder-only model (Touvron et al., 2023) and Flan-T5-XL (3B) – an encoder-decoder model (Chung et al., 2022). Our second design adopts a two-step pipeline approach: first extract the underlying data table from the chart image, then provide it as input to the LLM. This range of models provides a spectrum of efficient solutions, making our systems adaptable to various real-world scenarios and computational demands.

Our comprehensive evaluation across **four** benchmarks: ChartQA (Masry et al., 2022b), Chart2Text (Obeid and Hoque, 2020b), OpenCQA (Kantharaj et al., 2022b), ChartFC (Akhtar et al., 2023a) demonstrates our models' state-of-the-art performance in chart understanding and reasoning tasks. Human evaluation further suggests the effectiveness of our instruction-tuning approach in supporting a wide array of real-world chart comprehension and reasoning scenarios, broadening its adaptability to numerous new tasks.

Our main contributions include: (i) A new instruction-following corpus with real-world charts and a wide range of tasks by utilizing LLMs, (ii) two distinct systems specifically tailored for chart understanding tasks; (iii) extensive evaluations that demonstrate the state-of-the-art performance of ChartInstruct across existing chart-related benchmark tasks while also expanding its applicability to new tasks. We have made our code and chart corpus publicly available at https://github.com/vis-nlp/ChartInstruct.

2 Related Work

2.1 Chart Modeling

Chart understanding methods fall into two main categories: those directly fine-tuned from language or vision-language models (Masry et al., 2022b; Masry and Hoque, 2021; Lee et al., 2022) and those specifically crafted for chart-specific tasks (Masry et al., 2023; Liu et al., 2022b; Zhou et al., 2023). Models in the former category often exhibit limited performance due to a lack of chart-specific pretraining. In contrast, models in the latter group are pretrained on primitive chart-specific tasks like question answering and summarization which constrains their applicability across diverse real-world chart scenarios. Recent works (Wang et al., 2023a; Do et al., 2023; Huang et al., 2023) have also employed LLMs such as GPT-3, Llama (Touvron et al., 2023), and GPT-4 for chart-related tasks. They utilize pipelines that extract data values from chart images using specialized models like UniChart (Masry et al., 2023) and Deplot (Liu et al., 2022a), which are then used by the LLMs in different downstream tasks like question answering and summarization. However, these methods either rely on proprietary models (Brown et al., 2020; OpenAI, 2023) or public models without chart-specific training, limiting their effectiveness and generalization.

2.2 Visual Instruction Tuning

Instruction tuning in LLMs has shown benefits in aligning models with human intent and enhancing task generalization (Chung et al., 2022; Ouyang et al., 2022b; Wang et al., 2023b; Alpaca, 2023; Chiang et al., 2023). These techniques have also been extended to the vision-language space (Li et al., 2023a; Zhu et al., 2023; Ye et al., 2023; Li et al., 2023b; Dai et al., 2023). However, visual instruction tuning approaches in the chart domain are rare. Although a few studies have implemented multimodal instruction tuning (Liu et al., 2023a; Han et al., 2023), they depend on CLIP (Radford et al., 2021) for vision encoding, designed for natural images rather than charts. Moreover, their training often relies on synthetic charts or a narrow selection of real-world charts, focusing on a narrow set of instruction-tuning tasks. In contrast, our work presents two systems explicitly designed for the chart domain, trained on a diverse array of chart images and covering a wide range of real-world chart applications.

2.3 Chart Domain Downstream Tasks

Interest in chart-related tasks is rising, focusing on understanding and generating information from charts. Chart question answering (CQA) addresses queries about charts with some datasets (Methani et al., 2020) and (Masry et al., 2022a) focusing on visual and arithmetic reasoning, while others focus on open-ended explanatory question answering (OpenCQA) (Kantharaj et al., 2022b). Additionally, Chart-to-Text involves creating natural language summaries from charts (Shankar et al., 2022), and Chart-to-Table focuses on converting charts into data tables (Choi et al., 2019; Masry et al., 2023). Automated Fact-Checking (AFC) for images, including charts, aims to check claims against data (Akhtar et al., 2023a,b). In this paper, we evaluate our models on these different downstream tasks and also generate new kind of chart reasoning tasks through the instruction data generation process.

3 Chart Instruction Data Generation

We build an instruction tuning dataset for enhancing VLMs' capabilities in tackling diverse understainding and generation tasks related to chart analysis. In this section, we describe the chart corpus collection followed by the instruction tuning data generation process. Figure 2 provides an overview of the instruction tuning process.

3.1 Chart Corpora Collection

Our goal is to build a diverse chart dataset using real-world data to enhance our model's generalizability. In order to build that we collect chart images from two main sources: existing public datasets and web-crawled charts. We chose UniChart (Masry et al., 2023) from existing datasets, as it provides one of the largest and most diverse chart pretraining corpora, containing 611K charts with metadata like data tables, titles, and captions (refer to Masry et al. (2023) for details). However, these charts come from a few specific online sources such as Pew (Pew, -), Statista (statista, -), OECD (OCED, -), and OWID (OWID, -), limiting the variety of visual styles and data domains covered.

To address this limitation, we contribute with a new corpus, WebCharts, which contains **41K** di-

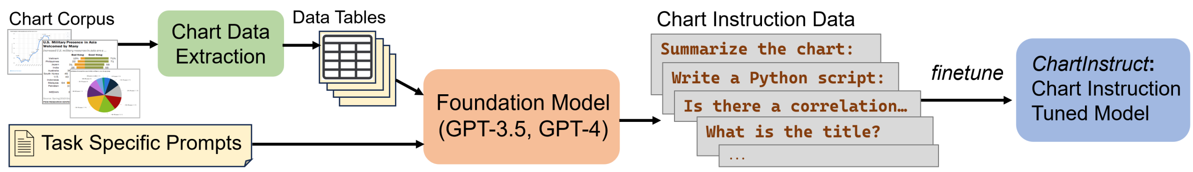


Figure 2: Instruction tuning process for chart collection. For the WebChart Corpus, the chart data is extracted automatically using Gemini Vision Pro. For distilling new tasks we use GPT-4, for other task generation we either use GPT 3.5 or GPT 4.

Dataset	CoT Reasoning	Chart Summarization	Fact Checking	Open-ended QA	Coding Abilities	Novel Tasks	#Unique Charts
Statista	4,363	4,159	4,188	4,906	2,348	-	9992
PlotQA	4,159	3977	4,333	12,105	2,306	-	8199
OECD/OWID	4290	3999	4,080	13,213	2,994	-	10949
WebCrawled	14,459	41741	11,574	12,246	11,924	23,410	41742
Total	27,271(14.3%)	53,876(28.24%)	24,175(12.67%)	42,470(22.26%)	19,572(10.26%)	23,410(12.27%)	70,882

Table 1: The number of generated examples for each tasks based on data samples of the mentioned dataset. Some of the charts are used in multiple tasks. On the last Column, we show the number of distinct charts used for instruction generation samples.

verse chart images. We started with a seed list of web domains containing charts (Hoque and Agrawala, 2019) and then use the top image search results from these domains using queries such as "chart images," "graphs," and "visual data". We then develop a binary VIT classifier (Dosovitskiy et al., 2021) to distinguish chart images from non-chart images in our search results, followed by manual removal of any remaining non-chart images to refine the dataset. However, these charts lack the underlying data tables which are critical for instruction generation on various chart data analysis tasks. Therefore, we automatically extract the data tables and chart titles using Gemini Pro Vision (Team et al., 2023). The choice of Gemini was influenced by the cost and the unlimited API rate features. More details about the web charts collection process are provided in Appendix A.1.

3.2 Instruction Data Generation

To enhance LLMs' performance in chart-related tasks via instruction tuning, we develop our chart instruction Dataset. This dataset has 190,774 instructions corresponding to 70,882 charts, covering various aspects of chart comprehension and reasoning (see examples in Figure 1). Below, we describe the process of generating instruction data.

***(i)* Tasks Selection:** To cover diverse aspects of chart reasoning and comprehension, we identify a set of tasks that are similar to some existing downstream tasks such as chart summarization and question answering (QA) but also included other tasks such as code generation and Chain of thought reasoning. Additionally, we prompted LLMs to propose novel tasks to enrich the dataset. Below we briefly explain these tasks.

• **Summarization and QA** The summarization task aims to generate a chart caption that captures the key insights such as trends and patterns from a given chart (Kantharaj et al., 2022c). We also include Open-ended QA (Kantharaj et al., 2022a) in which the model generates an explanatory answer to the given question about a chart.

• **Fact Checking** task (Akhtar et al., 2023a) is included to improve our model's ability to reduce errors and interpret chart data accurately. It takes a claim as input and then generates the verdict ('refute' or 'accept') along with an explanation.

• **Chain-of-thought (CoT) Reasoning** aims to enhance the model's ability to perform complex mathematical and visual reasoning. However, open-source models (e.g., Touvron et al. (2023), Jiang et al. (2023)) often incur errors in numerical computations (Islam et al., 2024a). To address this, we devised two types of questions: (1) Variable Dependent questions, which use tools to compute statistics, inspired by ToolFormer (Schick et al., 2023), and (2) Variable Independent Questions, focusing on retrieval, comparison, and basic math analysis.

• **Code Generation** task is included to generate executable Python scripts to answer user queries, drawing inspiration from the success of this approach demonstrated by PAL (Gao et al., 2023).

• **Novel Tasks** play a crucial role in enhancing the diversity of the instruction set. We tasked an LLM to propose different possible chart-related tasks. To prevent overlap with existing tasks like chart summarization and QA, we instructed the LLM not to replicate these tasks. The generated instructions involve tasks that may require new forms of reasoning and analysis (e.g., future value predictions).

***(ii)* Prompt Design:** To create the instructions

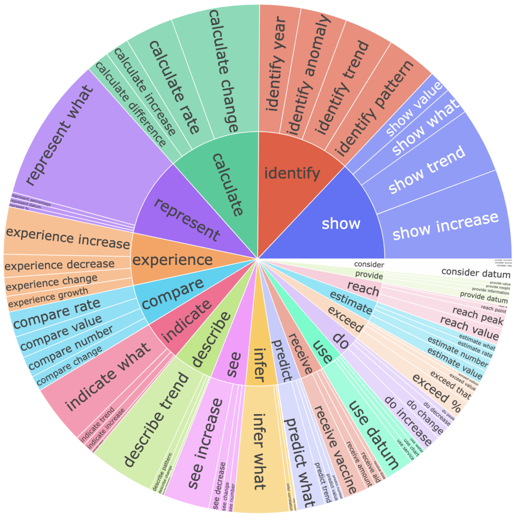


Figure 3: Top 20 most common root verb (inner circle) and corresponding four object verb pairs for all the generated instructions of our dataset.

for different tasks, we first design a set of prompt templates, where each template contains: (1) **task description**, (2) the **input** chart data table, along with metadata such as the chart title, (3) output **constraints** (if any), (4) **output format**. An example prompt is shown in Figure 5.

***(ii)* Input and Output Generation:** After designing prompts, we generate instructions, creating input-output instances for each prompt template. We concatenate each chart's underlying data table and title with one of the prompts designed for the expected task by utilizing OpenAI's GPT3.5 Turbo and GPT4 (see Table 4). The choice between these APIs for different tasks is dictated by the task complexity, with GPT-4 being employed for complex reasoning tasks, and GPT3.5 for tasks with moderate complexity. Moreover, in order to reduce the generation cost, and increase the variety of generated samples in each call, in our prompts, we ask the LLMs to come up with multiple samples for each chart. The input prompt for each task is provided through Tables 8-14 in Appendix A.10.

3.3 Dataset Analysis

We present key statistics and analyze the diversity and quality of the instruction dataset.

Statistics: Our Chart corpus (WebCharts) is highly diverse, encompassing a variety of bar and line charts, pie and donut charts, and even unconventional chart types not prevalent in existing chart corpora (see Figure 8). The generated instructions set is dominated by Chart summarization and open-ended QA to improve the chart comprehension ability but also augmented with reasoning tasks and creative new tasks generated by LLM (see Table 1). We placed particular emphasis on WebCharts dataset due to its diversity, constituting 67.5% (157,190 samples) of our dataset.

Diversity: To investigate the diversity of generated instructions, we employed the Berkeley Neural Parser (Berkeley, 2024) to identify the verb closest to the root along with its first direct noun object in each instruction. The analysis reveals a broad spectrum of comprehension and reasoning tasks expressed in the instructions (Figure 3). We further analyzed the newly proposed tasks generated by GPT-4 by clustering the instructions using the K-means algorithm. From Table 6 in Appendix A.4, we observe that Pattern and Outlier Detection is the most common type of task, followed by various statistical analyses. Notably, the dataset includes interesting tasks not typically captured by existing downstream tasks, such as identifying correlations, predicting values and trends, and distribution analysis. Overall, it suggests that the generated instructions set is indeed diverse and creative. We also visualize diversity in the length of the instructions' inputs, and instance outputs in Figure 9 and 10.

Quality: We asked an expert annotator to evaluate the quality of the generated data on a random set of 100 instructions. We find that in general, the instructions describe a valid task (87%) and the input matches the task description (86%) among generated instructions. In 61% and 8% cases, outputs for the generated inputs were fully and partially correct respectively. We list a number of correct and incorrect examples in Figure 11. We note that even when the outputs may be incorrect (e.g., contain factual errors), the corresponding task instructions can provide informative training signal as found by others (e.g., Honovich et al. (2022)).

4 Modeling

In this section, we describe our two architectures for chart instruction tuning.

4.1 End-to-End System

Our end-to-end system utilizes the LLaVA (Liu et al., 2023b) architecture, which incorporates CLIP (Radford et al., 2021) for visual encoding, an LLM for language generation, and an adapter module for transforming the encoded visual features to the LLM's input embedding space (Figure 4). LLaVA is originally designed for natural image understanding. We make the following modifications to adapt it for chart understanding. First, we substitute the CLIP vision encoder with the UniChart vision encoder (Masry et al., 2023), which is pretrained and optimized for chart image understanding. For the LLM, we investigate two model types: the decoder-only architecture of Llama2 (Han et al., 2023) and encoder-decoder structure of Flan-T5 (Chung et al., 2022). In the Llama2 setup, projected visual features are injected directly into the language decoder, whereas in the Flan-T5 model, these features, along with the instructions, are first processed by the language encoder before the decoder generates a text. We experimented with both the 7B variant of Llama2 and the 3B variant of Flan-T5 to explore different model sizes suitable for various applications and understand their learning differences (Parvez et al., 2023).

In this end-to-end design, before fine-tuning the model on instructional data, we first fine-tune only the adaptor module keeping the vision encoder and LLM frozen. This critical *alignment stage* is necessary to align the visual features from the UniChart vision encoder with the input embedding space of the LLM, and enables the LLM to accurately interpret chart images. We focus on two specific tasks for this phase: generating data tables from charts and summarizing chart contents. After alignment, we finetune the model on instruction tuning data, keeping the vision encoder frozen while training the weights of both the adaptor and LLM.

4.2 Pipeline System

In this approach, a data table generation module first converts the chart image into a textual data table representation.This generated data table is then combined with the input instruction and fed into an LLM. We utilize UniChart (Masry et al., 2023), which has been shown to be able to generate high-quality data tables from chart images, ensuring the textual representation closely mirrors the original chart's information. For the LLM, we conduct experiments with both Llama2, and Flan-T5 models similar to our end-to-end approach. Unlike the end-to-end system, this setup skips the alignment step since the visual features are not directly fed into the LLM. Hence, we directly finetune the models on the instruction data (see Figure 6).

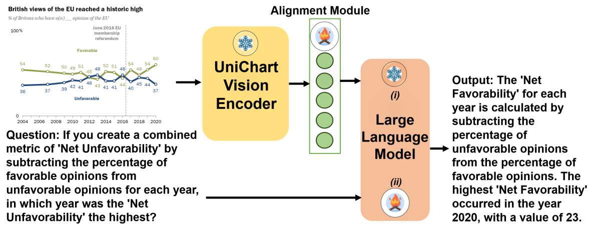


Figure 4: The architecture for our end-to-end system models: the LLM is frozen in our (i) pre-training step, while it updates its parameters in the (ii) instruction-tuning step. We either use Flan-T5-XL or Llama2 as LLM for this architecture. We show our pipeline system architecture in 6 in A.5

5 Experiments and Results

We evaluate the usefulness of ChartInstruct and demonstrate that our models built upon ChartInstruct achieve excellence in chart understanding and generation tasks. In addition to the existing downstream tasks, they also posit superior capabilities in new tasks. Below, we first discuss the setups, then experiments on downstream benchmarks and new chart tasks, and finally present an error analysis and challenges. Our evaluation complements the trivial quantitative approach based on automated metrics with detailed human evaluation on both seen and new tasks in multiple aspects, reflecting the true understanding of effectiveness of ChartInstruct . To ensure the reproducability of our research, we present the hyperparameters of our instruction tuning and downstream tasks experiments in Table 5. All experiments were conduced on a 4-A100 GPUs (80GB) machine.

5.1 Experimental Setup

Downstream Tasks: To assess the generalizability of our models across a spectrum of practical chart applications, we evaluate them on **four** established downstream tasks in the literature: (i) ChartQA (Masry et al., 2022b) – a factoid chart question answering dataset, (ii) OpenCQA (Kantharaj et al., 2022b) – an open-ended chart question answering dataset, (iii) Chart2Text (Shankar et al., 2022) – a chart captioning dataset collected from two sources: Statista (statista, -) and Pew Research Center (Pew, -), and (iv) ChartFC (Akhtar et al., 2023a) – a chart fact checking dataset. Furthermore, we conduct a human evaluation to explore their adaptability in real-world scenarios beyond these benchmarks.

Baselines: We compare ChartInstruct against seven open-source baselines: (1) *T5* (Raffel et al., 2020), a unified seq2seq Transformer model; (2) *VL-T5* (Cho et al., 2021), a T5-based model

Model	#Params	ChartQA (*RA*) aug.	human	avg.	OpenCQA (*BLEU*) OpenCQA	Chart-to-Text (*BLEU*) Pew	Statista	ChartFC (*Accuracy*) ChartFC
Open Source								
VisionTaPas (Masry et al., 2022a)	-	61.44	29.60	45.52	–	-	-	-
T5 (Masry et al., 2022a)	222M	56.96	25.12	41.04	9.28	10.49	35.29	-
VL-T5 (Masry et al., 2022a)	-	56.88	26.24	41.56	14.73	-	-	-
ChartBERT (Akhtar et al., 2023a)	-	-	-	-	-	-	-	63.8
Pix2Struct (Lee et al., 2022)	282M	81.6	30.5	56.0	-	10.3	38.0	-
Matcha(Liu et al., 2022b)	282M	90.2	38.2	64.2	-	12.2	39.4	-
UniChart (Masry et al., 2023)	201M	88.56	43.92	66.24	14.88	12.48	38.21	-
Closed LMMs								
Gemini Pro (Team et al., 2023)	-	-	-	74.1	6.84	28.5	25.8	65.8
GPT4-V (OpenAI, 2023)	-	-	-	78.5	3.31	35.9	18.2	69.6
End-to-End System								
ChartInstruct-Flan-T5-XL	3B	85.04	43.36	64.2	**16.71**	12.92	42.42	70.27
ChartInstruct-Llama2	7B	87.76	45.52	66.64	15.59	**13.83**	**43.53**	69.57
Pipeline System								
ChartInstruct-Flan-T5-XL	3B	**93.84**	**50.16**	**72.00**	14.81	9.93	40.08	**72.65**
ChartInstruct-Llama2	7B	82.40	40.64	61.52	14.78	12.81	39.39	64.99

Table 2: Evaluation results on four public benchmarks: ChartQA, Chart-to-Text, OpenCQA, and ChartFC. All the results are calculated after finetuning ChartInstruct.

for Vision-Language (VL) tasks; (3) *VisionTapas* (Masry et al., 2022a), an extension of TaPas (Herzig et al., 2020) for chart question answering; (4) *ChartBERT* (Akhtar et al., 2023b), a BERT-based model utilizing textual and visual information of charts for fact verification; (5) *Pix2Struct* (Lee et al., 2022), a pretrained image-to-text model; (6) *MatCha* (Liu et al., 2022b), an adaptation of Pix2Struct for charts pretrained on math reasoning; and (7) *UniChart* (Masry et al., 2023), achieving SoTA on Chart-to-Text, ChartQA, and OpenCQA. Another open source model is MMC (Liu et al., 2023a), but we exclude it from our evaluation results due to its poor performance. In addition to that, we also compare ChartInstruct against two closed-source LLMs: Gemini Pro (Team et al., 2023) and GPT4-V (OpenAI, 2023). In particular, we report the accuracy on ChartQA from the technical reports while the remaining datasets' metrics were obtained from (Islam et al., 2024b).

Evaluation Metrics We use Relaxed Accuracy (RA) for ChartQA (following Methani et al. (2020)), Accuracy for ChartFC (Akhtar et al., 2023b) and BLEU for text-generation tasks (Chart-to-Text and OpenCQA) (Post, 2018). However, BLEU focuses mainly on n-gram matching, overlooking factors like informativeness and factual correctness (Goyal et al., 2022). To address this, we conduct human evaluations to compare these aspects (see Section §5.3).

5.2 Results and Findings

We present the experimental results on downstream tasks in Table 2 and compare with existing baselines. ChartInstruct models (both end-to-end and pipeline) outperforms previous open-source state-of-the-art models, UniChart on all ChartQA and Chart-to-Text datasets. In particular, the Flan-T5-XL version excels on the ChartQA including the challenging human-written question set (Masry et al., 2022a), which suggests that the model learned more complex mathematical and visual reasoning through the relevant instruction tuning tasks such as CoT reasoning, and coding abilities. ChartInstruct also achieved a higher BLUE score compared to UniChart on OpenCQA benchmark, which demonstrates our model's capability to generate explanatory answers for questions about charts. Finally, ChartInstruct surpasses ChartBERT by a wide margin (8.85%) on the recently released fact-checking task. Overall, these results establish ChartInstruct as the SoTA open-source model for chart comprehension and reasoning tasks. Compared to closed-source models, ChartInstruct models outperforms both Gemini Pro (Team et al., 2023) and GPT4-V (OpenAI, 2023) on ChartFC, OpenCQA, and Chart-to-Text Statisa, while our ChartInstruct Flan-T5 achieves comparable performance on ChartQA.

Our observations reveal that the end-to-end system for ChartInstruct-LLama2 generally surpasses the corresponding LLama2 pipeline system across all benchmarks. This performance disparity is likely due to the fact that the data table alone does not capture all the nuanced information present in the charts, thus becoming a limiting factor in the pipeline system's effectiveness. Similarly, the end-to-end system of ChartInstruct-Flan-T5-XL performs better than the pipeline system on both OpenCQA and Chart-to-Text benchmarks. One notable exception is the reasoning-intensive tasks like ChartFC and ChartQA on which the pipeline Flan-T5-XL system exhibits better performance. Furthermore, we notice that both ChartInstruct-Flan-T5-XL and ChartInstruct-Llama2 achieve comparable performance, even tho the former has 4B fewer parameters. This efficiency makes ChartInstruct-Flan-T5-XL more suitable for real-world applications with computational constraints.

To further assess the impact of our different instruction tuning tasks on our model's performance, we conducted ablation studies on the ChartQA dataset using our best performing model, ChartInstruct-Flan-T5-XL (Pipeline System). Our ablation studies reveal that excluding tasks like Chart Summarization or Open-ended Question Answering results in a minor decline in performance (Table 7). This performance dip becomes significantly pronounced upon the removal of the reasoning tasks (CoT and Coding), emphasizing their pivotal role in enhancing the model's reasoning capabilities. More details about the experiments can be found in Appendix A.6.

5.3 Human Evaluation on Chart Tasks

Reference-based evaluation metrics like BLEU-score may not align with human-perceived text quality attributes (Liu et al., 2023c; Smith et al., 2016). To ensure accurate evaluation of our approach, we conducted a human experiment, assessing the generated responses from UniChart and our ChartInstruct-Llama2 model across three metrics: (a) Informativeness, (b) Relevance, and (c) Factual Correctness.

For the study, we chose 150 samples that are unseen by both UniChart and ChartInstruct-Llama2. Half of them are randomly from the ChartQA test set, while the other half is from a small set of web-crawled charts not used in the instruction generation pipeline. These samples contain queries from Open-ended QA, Chart Summarization, and novel instruction samples that involve a diverse set of tasks for evaluation. In terms of task distribution, 75 (50%) of the study samples belonged to novel tasks, while the other half comprised Chart-to-Text and OpenCQA tasks (40 and 35 samples). We use UniChart and ChartInstruct-Llama2 to generate responses for these samples. We asked 2 different annotators to rate the sample's responses based on the mentioned factors from 1-5, having 100 samples in common to measure their agreement level[1] toward the responses. We presented the responses randomly to prevent any biases toward models.

	Informativeness	Relevance	Factual
UniChart(Masry et al., 2023)	3.2	2.74	2.756
ChartInstruct-Llama2	3.848	4.06	3.664
$p-value$	7.43×10^{-4}	4.42×10^{-5}	1.31×10^{-8}

Table 3: Human evaluation results for comparing between the outputs of UniChart and ChartInstruct-Llama2. The first two rows show the average of samples across each metric. The last row shows the p-values resulted from performing Mann-Whitney U Tests.

From Table 3, we observe that ChartInstruct-Llama2 significantly outperforms UniChart across all three measures of human evaluation, especially in relevance (4.06 vs. 2.74). Upon manual examination, we observed that UniChart often provides a general summary of the chart without addressing specific task instructions (sometimes repeating the same tokens), particularly evident in novel and OpenCQA task samples. In contrast, ChartInstruct-Llama2 consistently offers relevant answers for these cases (see an example in Figure 7). Overall, these findings affirm that our instruction-tuning approach enhances the model's ability to adhere to task instructions, thereby expanding its capacity to address a wide array of new real-world chart-related scenarios beyond the capabilities of the state-of-the-art pre-trained model for the chart domain.

5.4 Error Analysis and Challenges

We reviewed our model's results across various samples to highlight the challenging aspects encountered.

[1]We found Cohen's Kappa of 0.447 as the agreement level.

Value Estimation and Comparison Charts with crowded or minimal details pose challenges in pairing visual elements (e.g., bars) with their associated values, estimating data values, and making comparisons based on visual attributes (compare based on height). For instance, errors occurred in Q1 and Q2 of Figure 12, where the correct value associated with specified items was not identified.

Factual Errors Although our models have shown improved text generation quality and better utilization of available information, they still produce statements unsupported by the chart or factually incorrect. In Q3 of Figure 12, the model produces coherent text but also introduces factual errors.

Numerical Reasoning Despite advancements, LLMs sometimes struggle with dependable mathematical operations (Gao et al., 2022; Liu et al., 2022b). While achieving state-of-the-art performance in ChartQA (Masry et al., 2022a) and attempting to teach the model to use external tools, LLMs still exhibit inconsistencies in calculations. Q4 in Figure 12 illustrates the unreliability of LLMs in some numerical reasoning tasks.

6 Conclusion

We present ChartInstruct, an automatically generated dataset of chart-related instructions and two instruction systems designed for a broad range of chart-related tasks. To the best of our knowledge, this is the first instruction tuned dataset that not only includes pre-defined tasks but also many new types of tasks automatically distilled by LLMs. Our model sets the state-of-the-art performance on four different downstream tasks on various automatic measures while the human evaluation further confirms the effectiveness of our approach on many new kinds of tasks. We believe that our models and instruction-tuning dataset will be valuable resources for future research and encourage further exploration into the unique problem domain of chart understanding and reasoning.

Limitations

First, while our research covers key tasks such as Chart Summarization, Chart Question-Answering, Open-ended Chart Question-Answering, and Chart Fact Checking, it does not cover other tasks, e.g., Chart-to-table. Second, while our manual inspection of instruction-tuning dataset suggests that the novel tasks distilled by LLM are generally valid and answerable, occasionally the outputs are incorrect which may influence the instruction-tuning process. Third, although our instruction tuning approach significantly enhances the model's ability to follow instructions compared to its counterpart, it does not entirely prevent the model from deviating from instructions. Fourth, despite the state-of-the-art performance on the numerical reasoning task, ChartQA, our model still struggle with complex numerical questions. Finally, the model may produce factually incorrect statements in the text generation tasks.

Ethics Statement

During the dataset collection process, we were mindful of several ethical considerations. The first three sources of our chart corpus (Statista[2], OWID[3], OECD[4]) grant publication rights for academic use of their content. Moreover, the PlotQA dataset (Methani et al., 2020) is a publicly available dataset published under MIT license[5]. We plan to release the chart images collected from these resources along with their metadata. For the WebCharts corpus, we plan to release only the URLs from which the chart images were collected following relevant large-scale vision-language datasets (e.g., LAION[6]). Furthermore, we release our models for only academic research purposes.

In our commitment to exclude harmful content from our chart images, we employed Google search for sourcing the chart images, leveraging its strict policies against harmful content[7]. Moreover, all sourced chart images underwent an initial automatic filtering process using a chart classifier, followed by a manual review phase. Additionally, the WebCharts images were processed through the Gemini API, which is designed to block unsafe content[8], thereby providing an additional layer of assurance regarding the appropriateness of the content included in our dataset.

Given the generative nature of our models, there is a potential risk of misuse where users can generate harmful or factually incorrect outputs, poten-

[2]https://www.statista.com/getting-started/publishing-statista-content-terms-of-use-and-publication-rights

[3]https://ourworldindata.org/faqs#can-i-use-or-reproduce-your-data

[4]https://www.oecd.org/termsandconditions/

[5]https://github.com/NiteshMethani/PlotQA

[6]https://laion.ai/

[7]https://blog.google/products/search/when-and-why-we-remove-content-google-search-results/

[8]https://ai.google.dev/docs/safety_setting_gemini

tially leading to the spread of misinformation. We urge users to exercise responsibility and caution, restricting their use of our models to academic and research purposes only.

The human evaluation was performed by the authors and their collaborators who were involved with this research. As the focus of the research was solely on assessing models' capabilities, effectiveness, and limitations in several chart understanding tasks, the human evaluation performed by the authors does not add any ethical issues or unwanted biases. We present our instructions to the human evaluators as well as a sample in Figures 14 and 15, respectively. Moreover, there were no paid participants involved in the study. Finally, no information has been used that can directly relate to the identification of any person while evaluating the responses from the models.

Acknowledgement

We would like to thank the anonymous reviewers for their helpful feedback. This research was supported by the Natural Sciences Engineering Research Council (NSERC) of Canada and Canada Foundation for Innovation (CFI).

References

Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. 2023a. Reading and reasoning over chart images for evidence-based automated fact-checking. *arXiv preprint arXiv:2301.11843*.

Mubashara Akhtar, Nikesh Subedi, Vivek Gupta, Sahar Tahmasebi, Oana Cocarascu, and Elena Simperl. 2023b. Chartcheck: An evidence-based fact-checking dataset over real-world chart images. *arXiv preprint arXiv:2311.07453*.

Alpaca. 2023. Alpaca. https://crfm.stanford.edu/2023/03/13/alpaca.html.

Berkeley. 2024. Berkeley neural parser.

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

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *ICML*.

J. Choi, Sanghun Jung, Deok Gun Park, J. Choo, and N. Elmqvist. 2019. Visualizing for the non-visual: Enabling the visually impaired to use visualization. *Computer Graphics Forum*, 38.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning.

Xuan Long Do, Mohammad Hassanpour, Ahmed Masry, Parsa Kavehzadeh, Enamul Hoque, and Shafiq Joty. 2023. Do llms work on charts? designing few-shot prompts for chart question answering and summarization.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3.

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. Chartllama: A multimodal llm for chart understanding and generation. *arXiv preprint arXiv:2311.16483*.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.

Enamul Hoque and Maneesh Agrawala. 2019. Searching the visual style and structure of d3 visualizations. *IEEE transactions on visualization and computer graphics*, 26(1):1236–1245.

Enamul Hoque, Parsa Kavehzadeh, and Ahmed Masry. 2022. Chart question answering: State of the art and future directions. *Journal of Computer Graphics Forum (Proc. EuroVis)*, pages 555–572.

Kung-Hsiang Huang, Mingyang Zhou, Hou Pong Chan, Yi R. Fung, Zhenhailong Wang, Lingyu Zhang, Shih-Fu Chang, and Heng Ji. 2023. Do lvlms understand charts? analyzing and correcting factual errors in chart captioning.

Md Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. 2024a. Mapcoder: Multi-agent code generation for competitive problem solving. *arXiv preprint arXiv:2405.11403*.

Mohammed Saidul Islam, Raian Rahman, Ahmed Masry, Md Tahmid Rahman Laskar, Mir Tafseer Nayeem, and Enamul Hoque. 2024b. Are large vision language models up to the challenge of chart comprehension and reasoning? an extensive investigation into the capabilities and limitations of lvlms.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Shankar Kantharaj, Xuan Long Do, Rixie Tiffany Leong, Jia Qing Tan, Enamul Hoque, and Shafiq Joty. 2022a. OpenCQA: Open-ended question answering with charts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11817–11837, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shankar Kantharaj, Xuan Long Do, Rixie Tiffany Ko Leong, Jia Qing Tan, Enamul Hoque, and Shafiq Joty. 2022b. Opencqa: Open-ended question answering with charts. In *Proceedings of EMNLP (to appear)*.

Shankar Kantharaj, Rixie Tiffany Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022c. Chart-to-text: A large-scale benchmark for chart summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4005–4023, Dublin, Ireland. Association for Computational Linguistics.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2022. Pix2struct: Screenshot parsing as pretraining for visual language understanding. *arXiv preprint arXiv:2210.03347*.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2023a. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pretraining with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

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

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

Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. 2023a. Mmc: Advancing multimodal chart understanding with large-scale instruction tuning. *arXiv preprint arXiv:2311.10774*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023c. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Ahmed Masry and Enamul Hoque. 2021. Integrating image data extraction and table parsing methods for chart question answering. *Chart Question Answering Workshop, in conjunction with the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–5.

Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. 2023. UniChart: A universal vision-language pretrained model for chart comprehension and reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (to appear)*. Association for Computational Linguistics.

Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022a. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

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

Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Jason Obeid and Enamul Hoque. 2020a. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 138–147, Dublin, Ireland. Association for Computational Linguistics.

Jason Obeid and Enamul Hoque. 2020b. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 138–147. Association for Computational Linguistics.

OCED. -. Organisation for economic co-operation and development (oecd). `https://www.oecd.org`. Accessed: Jan 2024.

OpenAI. 2023. GPT-4 Technical Report.

OpenAI-Blog. 2022. Chatgpt: Optimizing language models for dialogue.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022a. Training language models to follow instructions with human feedback.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022b. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

OWID. -. Our world in data (owid). `https://ourworldindata.org/`. Accessed: Jan 2024.

Md Rizwan Parvez, Jianfeng Chi, Wasi Ahmad, Yuan Tian, and Kai-Wei Chang. 2023. Retrieval enhanced data augmentation for question answering on privacy policies. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 201–210.

Pew. -. Pew research center. `https://www.pewresearch.org/`. Accessed: Jan 2024.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

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

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools.

Kantharaj Shankar, Leong Rixie Tiffany Ko, Lin Xiang, Masry Ahmed, Thakkar Megh, Hoque Enamul, and Joty Shafiq. 2022. Chart-to-text: A large-scale benchmark for chart summarization. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2022*.

Danqing Shi, Xinyue Xu, Fuling Sun, Yang Shi, and Nan Cao. 2020. Calliope: Automatic visual data story generation from a spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):453–463.

Aaron Smith, Christian Hardmeier, and Joerg Tiedemann. 2016. Climbing mont BLEU: The strange world of reachable high-BLEU translations. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 269–281.

statista. -. Statista. `https://www.statista.com/`. Accessed: 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, and Jiahui Yu et al. 2023. Gemini: A family of highly capable multimodal models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Peifang Wang, Olga Golovneva, Armen Aghajanyan, Xiang Ren, Muhao Chen, Asli Celikyilmaz, and Maryam Fazel-Zarandi. 2023a. Domino: A dual-system for multi-step visual language reasoning. *arXiv preprint arXiv:2310.02804*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. mplug-owl: Modularization empowers large language models with multimodality.

Mingyang Zhou, Yi Fung, Long Chen, Christopher Thomas, Heng Ji, and Shih-Fu Chang. 2023. Enhanced chart understanding via visual language pre-training on plot table pairs. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1314–1326, Toronto, Canada. Association for Computational Linguistics.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Appendices

A Chart Instruction Data Generation

A.1 Chart Corpora Collection

WebCharts Collection: For collecting charts, we conducted image queries on Google and collected chart images from 157 unique source domains. The list of keywords we use are: "chart images," "charts ","graphs," "visual data", and "data visualization". In each query, we included the web domain (e.g., "site: nytimes.com charts") and retrieved the top image search results. We developed a binary VIT classifier (Dosovitskiy et al., 2021) to distinguish chart images from non-chart images in our search results. For training, we manually labeled 1,200 images and split them into 8:1:1 ratios for train, validation, and test sets. The classifier achieved 91% accuracy on the test set. Using this classifier, we filtered out non-chart images from the WebCharts dataset, followed by manual removal of any remaining non-chart images to finalize the dataset. However, these charts lacked the underlying data tables necessary for instruction generation. Therefore, we automatically extracted the data tables and chart titles using Gemini Pro Vision (Team et al., 2023).

A.2 LLMs used for Instruction Generation

Table 4 shows the models used to generate the data for different tasks.

A.3 Input Prompts for Instruction Generation

Figure 5 shows an example prompt to LLM and the corresponding output for a fact-checking task.

A.4 Instruction Dataset Analysis

WebCharts corpus: Figure 8 shows the chart type statistics for WebCharts corpus. Since we do not have access to the chart types in this corpus, we manually tagged random 200 images from it to estimate the chart type distribution.

Downstream Task/Model	GPT-3.5	GPT-4
Chart Summarization	✓	✗
Open-ended QA	✓	✗
Fact Checking	✓	✗
Chain-of-Thought Reasoning	✗	✓
Code Generation	✓	✓
Novel tasks	✗	✓

Table 4: Models used to generate the data for each different task. Choices are based on task complexity and costs.

	End-to-End System				Pipeline System			
Experiment	**# Epochs**	**Learning Rate**	**Batch Size**	**Hours**	**# Epochs**	**Learning Rate**	**Batch Size**	**Hours**
	Alignment							
Flan-T5-XL	4	2e-3	128	7	-	-	-	-
Llama 2	3	2e-3	64	24	-	-	-	-
	Instruction Tuning							
Flan-T5-XL	3	2e-5	32	8	3	1e-4	64	17
Llama 2	3	2e-5	32	20	3	1e-4	64	21
	Finetuning on downstream tasks (Flan-T5-XL)							
ChartQA	10	1e-4	128	3	10	1e-4	128	7
OpenCQA	10	1e-4	128	1.5	10	1e-4	128	3
Chart-to-text Pew	10	1e-4	128	2	10	1e-4	128	3
Chart-to-text Statista	10	1e-4	128	4	10	1e-4	128	8
ChartFC	10	1e-4	128	1	10	1e-4	128	2
	Finetuning on downstream tasks (Llama 2)							
ChartQA	10	2e-5	32	6	10	1e-4	64	8
OpenCQA	10	2e-5	32	2	10	1e-4	64	4
Chart-to-text Pew	10	2e-5	32	2	10	1e-4	64	4
Chart-to-text Statista	10	2e-5	32	6	10	1e-4	64	9
ChartFC	10	2e-5	32	3	10	1e-4	64	3

Table 5: Training details for our instruction tuning and downstream tasks finetuning experiments.

'Input'
'Task Description'
Analyze a chart and generate pairs of claims and verdicts about its data. Avoid using terms like 'rows', 'columns', or 'elements' from the data table; refer to 'chart' or 'chart image' instead. Each response must state whether it 'supports' or 'refutes' the question, followed by a brief explanation.
'Datatable and other metadata'
Title: Premier league table

```
| Team            | P | W | Pts |
|-----------------|----|----|-----|
| Liverpool      | 10| 8 | 26  |
| Man City       | 10| 7 | 23  |
| Chelsea        | 10| 6 | 20  |
| Man United     | 10| 5 | 19  |
| Arsenal        | 10| 5 | 17  |
```

'Constraints':
Come up with 2-3 pairs of Claims and Verdicts with no additional text.
'Output Format'
Put ~ before starting each claim, and ^ before each verdict.

Output:
~ Liverpool has the highest number of points among the listed teams.
^ Supports: Liverpool indeed has the highest points (26) among the listed teams, indicating their strong performance relative to others.

~ Manchester City has the most wins among the listed teams.
^ Refutes: While Manchester City has a respectable number of wins (7), Liverpool actually has more wins (8), making them the team with the most wins in the chart.

~ Arsenal has fewer points than both Manchester United and Chelsea.
^ Supports: Arsenal has 17 points, which is fewer than Manchester United's 19 points and Chelsea's 20 points, indicating they are currently positioned below these teams in terms of pts.

Figure 5: An example prompt to LLM and the corresponding output for a fact-checking task. The input consists of the task description, chart data, any output constraints, and output format.

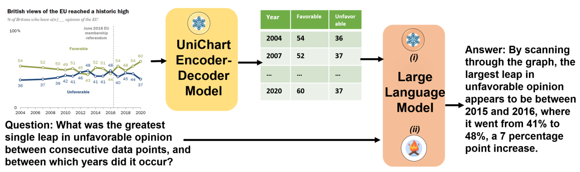


Figure 6: The architecture for our pipeline models: the encoder is frozen, while LLM updates its parameters.

A.5 Modeling

Figure 6 shows the pipeline system architecture we use for our models.

A.6 Ablation Studies

To understand the impact of the different instruction tuning tasks on the performance of our model, we conducted ablation studies on the ChartQA dataset using our top performing model, ChartInstruct-Flan-T5-XL (Pipeline System). These ablation experiments involved the removal of one task at a time, except for reasoning tasks, which were grouped and removed together. Due to computational constraints, we finetuned our model on the instruction tuning data for only 1 epoch only, as opposed to the 3 epochs used in the primary experiment. All other hyperparameters remained consistent with those detailed in our main experiments, as outlined in Table 5, including the fine-tuning experiments on the downstream tasks.

As depicted in Table 7, removing tasks like Chart

Task Group	#Examples
Pattern and Outlier Detection	16,977
Statistical Analysis	9,148
Extremum Identification	6,545
Data Correlations	6,142
General Comparison	4,709
Relative Change Calculation	4,108
Time Series and Future Value Forecasting	1,944
Data Point Identification	1,670
Performance and Result Analysis	1,396
Data Categorization	533
Distribution analysis	280

Table 6: Number of generated examples for various groups of new tasks created by GPT-4.

Model	ChartQA (RA)
ChartInstruct-Flan-T5-XL	70.08
No Open-ended Question Answering	69.68
No Chart Summarization	69.76
No Fact Checking	66.28
No Novel Tasks	69.20
No CoT Reasoning/Coding	63.36

Table 7: ChartInstruct ablations on ChartQA benchmark.

Summarization and Open-ended Question Answering had a negligible effect on the performance on ChartQA. However, a more significant performance decline was observed upon the exclusion of the fact-checking task, which is important for enhancing the model's data retrieving and reasoning capabilities. This decline was further amplified when reasoning-associated tasks (CoT and Coding) were removed, underscoring their critical role in improving the numerical reasoning capabilities of our model.

A.7 Human Evaluation Study

Figure 7 Shows a comparison between ChartInstruct-Llama2 and UniChart.

A.8 Error Analysis

Figure 12 show a few samples which our model faced a challenge generating factual and accurate responses for. In samples Q1 and Q2, our model fails to find the right value for the expected target either by estimation, matching or comparison to other visual elements. In Q3, Although It generates a cohesive summary, it produces some statements that are not true. Q4 shows a numerical error that ChartInstruct-Llama2 didn't perform the subtraction operation correctly.

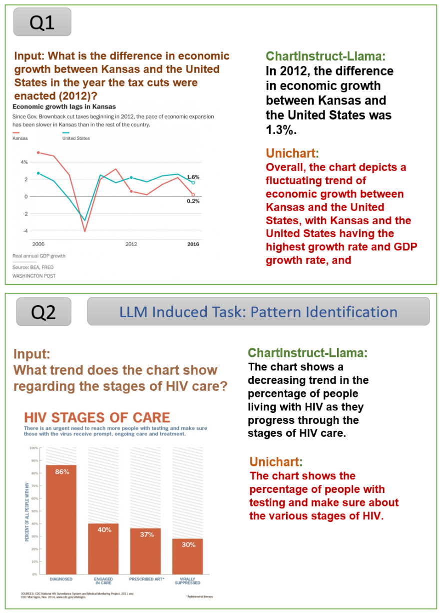


Figure 7: Comparison of ChartInstruct-Llama2 and Unichart over two WebChart novel task samples

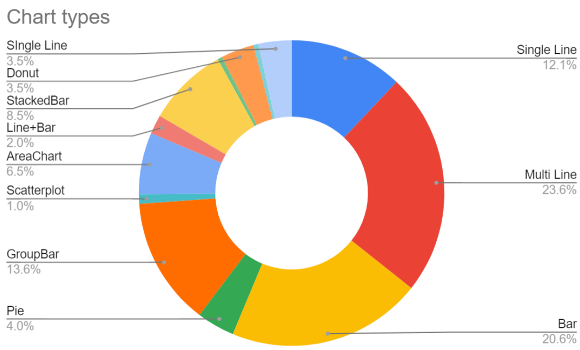


Figure 8: Chart types in WebChart Corpus.

A.9 Sample Outputs from ChartInstruct

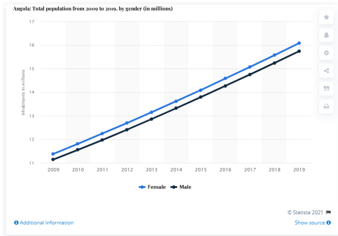
In Figure13, we provide some sample outputs on various tasks.

A.10 Generated Data Samples

The input prompt for each specific task is provided through Tables 8-14.

Fact Checking Prompt for GPT-3.5

Prompt:



Analyze a chart and generate pairs of claims and verdicts about its data. Half of the claims should be supported by the chart's data, while the other half are refuted. Avoid using terms like 'rows', 'columns', or 'elements' from the data table; refer to 'chart' or 'chart image' instead. Each response must state whether it 'supports' or 'refutes' the question, followed by a brief explanation. The claims should cover comparisons of values or trends, basic statistical values (maximum, minimum, mean, median, mode) without using exact numbers from the chart. Ensure a diverse range of claims addressing various visual aspects of the chart, resulting in 5-6 turns of C&V. Put ˜ before starting each claim, and ˆ before each answer.

Title:

Angola: Population from 2009 to 2019 by gender(in million)

Table:

Characteristic, Female, Male
2019, 16.08, 15.74
2018, 15.57, 15.24
2017, 15.07, 14.75
2016, 14.58, 14.27
2015, 14.09, 13.79
2014, 13.62, 13.32
2013, 13.15, 12.86
2012, 12.69, 12.41
2011, 12.25, 11.97
2010, 11.81, 11.55
2009, 11.38, 11.14

Extracted Instruction:

Claim:

In 2019, there was a significant decrease in the total population of males compared to 2018.

Verdict:

Refutes. The chart indicates a slight increase in the population of males from 2018 to 2019, as the value for males in 2019 (15.74 million) is higher than that in 2018 (15.24 million), contradicting the claim of a significant decrease.

Table 8: An example of prompt for Fact Checking Instruction Generation using GPT-3.5-turbo and an extracted sample

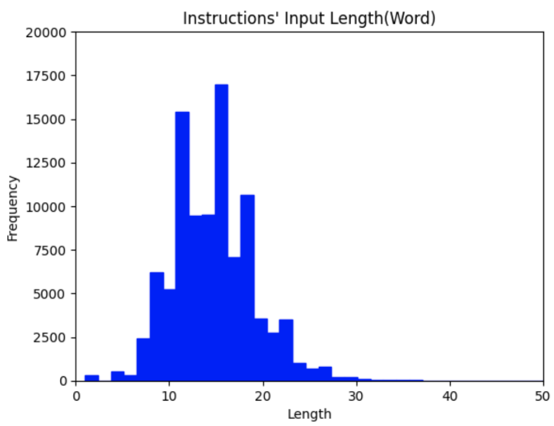


Figure 9: A histrogram of Instruction's Input Length distributions

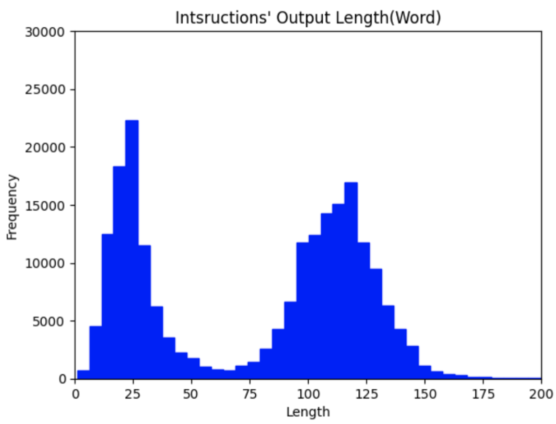


Figure 10: A histrogram of Instruction's Output Length distributions

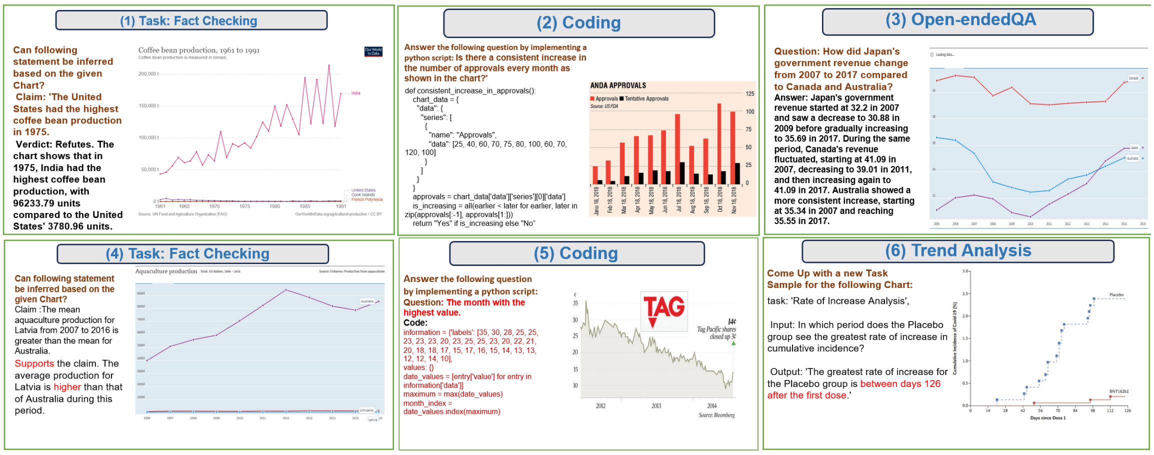


Figure 11: Few generated instruction samples. The Top row shows correct generated instructions. The bottom row shows some of the instructions involve errors: (4) generated wrong answer, (5) not following the description - wrong answer, (6) wrong answer. The errors are shown in red.

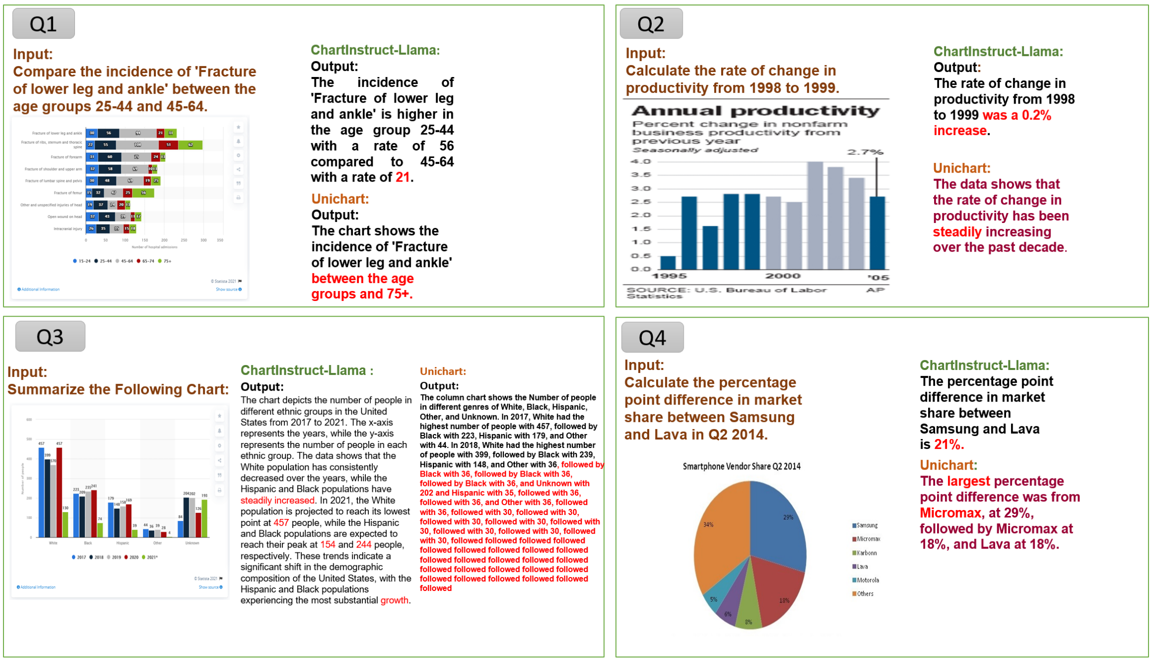
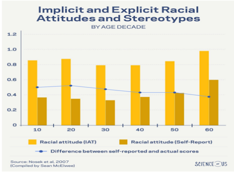


Figure 12: Some samples that our ChartInstruct-Llama2 and Unichart had errors in their outputs. The errors are shown in red. Unichart often does not follow the queries and occasionally repeats tokens. On the other hand, ChartInstruct-Llama2 tends to adhere better to the instructions but occasionally encounters factual and other errors.

Q1

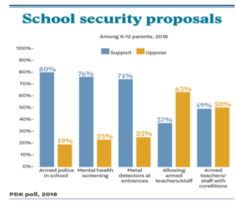
Input:
Provide an open-ended answer to the following question based on the chart image. Is there a correlation between age and the degree of discrepancy between self-reported and IAT racial attitudes? If so, what does this correlation suggest?



ChartInstruct-Llama:
Output:
Yes, there is a correlation between age and the degree of discrepancy between self-reported and IAT racial attitudes. This correlation suggests that younger individuals tend to have a higher degree of discrepancy compared to older individuals.

Q2

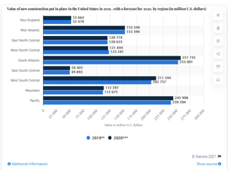
Input:
Is there higher support for armed police in schools or metal detectors at entrances among K-12 parents?



ChartInstruct-Llama:
Output:
Armed police in schools have higher support at 80% compared to metal detectors at entrances, which have 74% support.

Q3

Input:
Generate a python function that return the answer to the following questions based on the chart image. Which region had the largest increase or decrease in the value of new construction from 2019 to 2020, and by how much did it change?

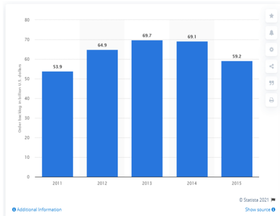


ChartInstruct-Llama :
Output:

```
def largest_change_region_2019_to_2020():
    information = {"New England": {"2019": 52647, "2020": 53348},
                   "Mid-Atlantic": {"2019": 153394, "2020": 155338},
                   "East North Central": {"2019": 121844, "2020": 123881},
                   "South Atlantic": {"2019": 50405, "2020": 50459},
                   "East South Central": {"2019": 49893, "2020": 50493},
                   "West South Central": {"2019": 40211, "2020": 40893},
                   "Mountain": {"2019": 113397, "2020": 112075},
                   "Pacific": {"2019": 23934, "2020": 24398}
    }
    largest_change = 0
    region = ""
    for region, value in information.items():
        change = value["2020"] - value["2019"]
        if change > largest_change:
            largest_change = change
            region = region
    return region, largest_change
```

Q4

Input:
Provide a summary for the following chart

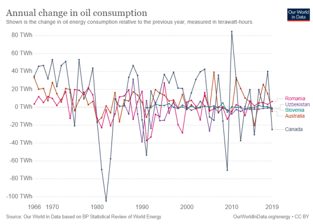


ChartInstruct-Llama:
Output:
The chart depicts the trend in the order backlog in U.S. dollars from 2011 to 2015. The x-axis represents the years, while the y-axis represents the order backlog in U.S. dollars. The maximum value of the order backlog was observed in 2013 at 69.7 U.S. dollars, while the minimum value was in 2011 at 53.9 U.S. dollars. The data shows a fluctuating trend in the order backlog over the years, with a peak in 2013 followed by a decrease in 2014 and 2015. Overall, the chart illustrates the variation in the order backlog over the five-year period, with 2013 standing out as the year with the highest backlog.

Figure 13: Sample outputs generated by ChartInstruct-Llama2 over various tasks such as LLM induced Data Correlation, Open-EndedCQA, Coding, and Chart Summarization

Summarization Prompt for GPT-3.5

Prompt:

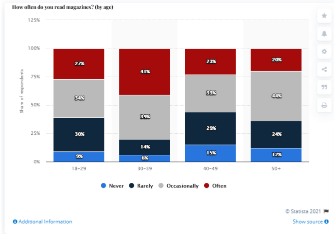


Summarize a chart based on a provided data table, highlighting key aspect such as data background, maximum and minimum values, value comparisons, and trends across categories. The summary should not directly mention the input table, instead referring to the 'chart' or 'chart elements'. You'll also be given the title of the chart to get a context on what is shown in the chart. Do not use the title as it is like "the chart is titles TITLE", always try to infersomething new based on it. Include a brief explanation of the elements represented on the x-axis and y-axis. The entire summary should be cohesive, in paragraph form without bullet points, and not exceed 150 words. Conclude with a final sentence that encapsulates the most significant findings from the chart.
Title:
Title: Annual Change in Oil Consumption
Table:
Characteristic Romania Uzbekistan Slovenia Australia Canada
1966 5.31, -, -, 34.29, 35.92
1970 11.43, -, -, 28.99, 54.29
1980 -10.61, -, -, -13.87, -15.1
1990 19.19, -36.32, -, 7.35, -12.24
2000 7.35, 1.64, 0.01, -0.81, 1.64
2010 4.5, -5.71, 1.64, 34.29, 31.84
Extracted Instruction:
Summary:
The chart depicts the annual change in oil consumption across five countries over several decades. On the x-axis, time spans from 1966 to 2010, while the y-axis represents the percentage change in oil consumption. Each line on the chart represents a different country: Romania, Uzbekistan, Slovenia, Australia, and Canada. Maximum values are observed in Australia and Canada in various years, while Romania shows relatively stable fluctuations. Uzbekistan experiences significant fluctuations, including a notable decrease in 1990. Slovenia and Canada display a generally increasing trend. Overall, Australia and Canada exhibit the highest peaks and most consistent growth in oil consumption over the period analyzed.

Table 9: An example of prompt for Summarization Instruction Generation using GPT-3.5-turbo and an extracted sample

OCQA Prompt for GPT-3.5

Prompt:



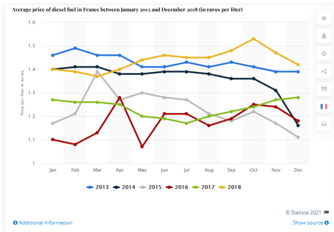
Generate question-answer pairs based on a data table representing a chart. You will also be given the title to have additional context. The questions should vary in complexity, ranging from simple 'what', 'which', 'when' questions to more challenging 'how' and 'why' questions. Refer to 'chart' or 'chart elements' instead of the data table. Ensure the answers are cohesive and fluent, using diverse vocabulary, with each answer spanning 4 to 5 sentences. The questions should encompass different aspects related to a chart figure. Each turn of conversation should consist of 5 or 6 question-answer pairs, covering a range of insights, explanations, and complexities as mentioned. Try to avoid generic questions such as what is the title or subtitle of the charts. Having questions with more challenging targets is preferable. There's no specific order for the difficulty of the questions. Put ˜ before starting each question, and ˆ before each answer.

Title:
How often do you read magazinzs(by age)
Table:
Characteristic Never Rarely Occasionally Often
18-29 9% 30% 34% 27%
30-39 6% 14% 39% 41%
40-49 15% 29% 33% 23%
50+ 12% 24% 44% 20%
What is the trend in magazine reading frequency among different age groups?
Magazine reading frequency varies significantly across age groups. Young adults aged 18-29 predominantly fall into the "Occasionally" category, with 34% indicating they read magazines occasionally. In contrast, those aged 30-39 and 40-49 are more likely to read magazines "Often," with 39% and 33% respectively. However, among individuals aged 50 and above, the frequency of magazine reading declines, with only 20% indicating they read often.

Table 10: An example of prompt for OCQA Instruction Generation using GPT-3.5-turbo and an extracted sample

COT Prompt for GPT-4

Prompt:



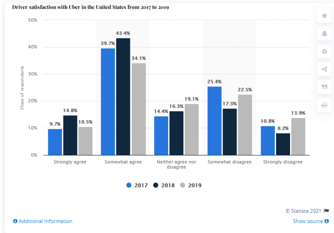
You'll be given a data table which is an underlying data table for a chart. Although you can't see the chart, you should mention any references that come up to the chart instead of the table. Your responses must be limited to question-and-answer pairs related to the data, strictly avoiding any conversational language for explanation or fillers. Questions must be at most two-part answer. You have to come up with six pairs of questions and answers that ask for a chart value, or comparisons not asking for a number, and a general trend. There must be at least two retrieval questions and one general trend. To come up with your answer, you have to break down your solution into several basic steps. In each step, you explain what are you going to do, and if that is a mathematical operation, you have to mention the formula and fill it with the respective numbers. Avoid using = operator in your calculations. If you are comparing two or several numbers retrieved from the table, you must mention them. Consider the following template: " The Answer is <ANSWER>." to wrap up your generated steps, replace <ANSWER> in the mentioned template with your final answer word in the most minimalistic way without ANY explanations or assignment. Please return the list of questions and responses in json format. I should be able to parse it. Each dictionary must contain 3 field 'question', 'steps', 'answer' filled with the requested formation.

Title:
Average price od diesel fuel in France between January 2023 and December 2018(in euroes per liter)
Table:
Characteristic 2013 2014 2015 2016 2017 2018
Dec 1.39 1.16 1.11 1.18 1.28 1.42
Nov 1.39 1.31 1.17 1.24 1.27 1.47
Oct 1.41 1.36 1.22 1.25 1.24 1.53
Sep 1.43 1.36 1.18 1.19 1.22 1.48
Aug 1.41 1.38 1.21 1.16 1.2 1.45
Jul 1.43 1.39 1.27 1.21 1.17 1.45
Jun 1.41 1.39 1.28 1.21 1.19 1.46
May 1.41 1.38 1.3 1.07 1.2 1.44
Apr 1.46 1.38 1.27 1.28 1.25 1.4
Mar 1.46 1.41 1.39 1.13 1.26 1.37
Feb 1.49 1.41 1.21 1.08 1.26 1.39
Jan 1.46 1.4 1.17 1.1 1.27 1.4
Extracted output:
{ "question": "What is the trend in diesel fuel prices from January to December in 2018?",
"steps": "To determine the trend, observe the prices from January to December in 2018. Start with January (1.4) and end with December (1.42), noting any significant increases or decreases throughout the year.",
"answer": "The Answer is increase." }

Table 11: An example of Variable-Independent prompt for COT instruction Generation for GPT-4-preview and an extracted sample

COT Prompt for GPT-4-preview

Prompt:



First of all, know that you must act like you can't do basic match even addition or subtraction. However, you might feel a need for calculation in your incoming tasks. In that case, you can tell me to calculate by calling Calculator(X = OPERATION). <OPERATION> can be executable python code and X is a variable. Use the defined variables in the next sentences when needed like a python code. Do not define the variables in the text. Use this format "DEFINE(variable_name=X)" where X is the value of the variable. You'll be given a datatable in json string format which is an underlying datatable for a chart. Although you can't see the chart, you should mention any references that come up to the chart instead of the table. Your responses must be limited to question and answer pairs related to the data, strictly avoiding any conversational language or fillers. You have to come up with six pairs of question and answers that ask for values that require calculations such as differences, totals, and statistical measures such as median, mode, and mean. To come up with your answer, you have to break down your solution into several basic steps. In each step, you explain what are you going to do, and if that is a mathematical operation, you have to mention the formula and fill it with the respected numbers. If you want to list some numbers, do it when there are less than 8 of them. Since you are unable to do calculations, you may return one of the defined variables from your previous steps as your final answer. Consider the following template: " The Answer is <ANSWER>." to wrap up your generated steps, replace <ANSWER> in the mentioned template with your final answer word without ANY explanations or assignment. Please return the list of questions and responses in a json format. I should be able to parse it. Each dictionary must contain 3 field 'question', 'steps', 'answer' filled with the requested formation. The answer section must only involve a python variable previously defined. No numbers should appear in this section.

Title:
Driver Satisfaction with Uber in United States from 2017 to 2019
Table:
Characteristic 2017 2018 2019
Strongly disagree 10.80% 8.20% 13.90%
Somewhat disagree 25.40% 17.30% 22.50%
Neither agree nor disagree 14.40% 16.30% 19.10%
Somewhat agree 39.70% 43.40% 34.10%
Strongly agree 9.70% 14.80% 10.50%
Extracted Output:
{ "question": "What is the increase in the percentage of drivers who somewhat agree from 2017 to 2018?",
"steps": "DEFINE(somewhat_agree_increase=43.4-39.7)", "First, I will subtract the percentage of drivers who somewhat agree in 2017, which is 39.7%, from the percentage of drivers who somewhat agree in 2018, which is 43.4%.",
"answer": "The Answer is somewhat_agree_increase." }

Table 12: An example of Variable Dependent prompt for Chain of Thought Instruction Generation using GPT-4 and an extracted sample

Rules:

Below you will see a series of charts, each chart accompanies one question/task and two models' answers. Your goal is to rate the quality of answers based on the following criteria.

(i) Informativeness: how much information from the chart does the answer cover? Ideally, an informative response should contain high-level insights from the chart
(ii) Relevance: how relevant the answer is to the input task?
(ii) Factual Correctness: how factually correct is the answer (facts mentioned are supported by the chart);

Please rate from 1(worst rating) to 5 (best rating).

Figure 14: Evaluation Rules of the human study for the three following metrics: informativeness, relevance, and factual correctness.

Coding Prompt for GPT-3.5

Prompt: You are a programmer expert in python. I want to analyze chart data. However, I can only provide you the underlying data as json line format representing the chart instead of its actual image. The underlying data involves different elements. I want you to come up with pairs of insightful questions, and a python function that answers the insightful question. Here are the requirements for the questions and answers:
- Do not refer to the table in the question. Mention " from the chart" if needed.
- Your output is limited to several questions followed by a Python script, strictly avoid providing any explanation before or after questions and answers such as stating the question type.
- Use ˜ at the beginning of each question and ˆ after finishing each questions.
- question and answers should appear one by one in the following format: Questionˆ, Python Script, Questionˆ, Python Script.
- Nothing in English should appear in the output other than the questions.
- Act like you don't have access to the table I have given you. As a result, you must define the given data at the beginning of each function.
- Inside each generated function, Define a dictionary called information that involves the data split such as labels and values.
- The Python script must only be a function and not anything else such as writing a body for the code and calling the function.
- The function should be independent and not rely on any sources out of that function.
- The generated script must be executable by Python, so avoid writing anything such as ... or // in between.
- The functions must not receive any input arguments.
- Everything that is needed must be defined in the function.
- Variable independent are defined as follows: Questions that want to return a string as response. The funtion must return a string which either "Yes"/"No" based on the conditions, trends, or a string that explains something like which characteristic does this number belong to?
- Variable dependent questions are defined as follows: Questions that ask for either a characteristic which is not known and must be found, or a statistical metric and measures that must be calculated. The output would be either a calculated number or a variable.
- Retrieval Questions: Questions that ask for characteristics or values represented in the chart.
- Hard questions: Questions that involve applying multiple filters and calculating advanced statistical metrics.

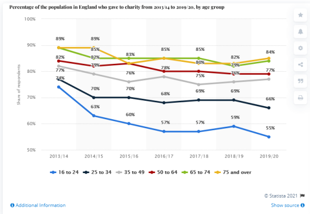
Come up with 8 question for each table in the following order, two variable independent questions, and a mix of three retrieval and variable dependent questions and three hard questions.

Title:

Percentage of Population in England who gave to charity from 2013/14 to 2019/20 by age group

Table:

Characteristic 16 to 24 25 to 34 35 to 49 50 to 64 65 to 74 75 and over
2013/14 74% 77% 82% 84% 89% 89%
2014/15 63% 70% 79% 82% 85% 89%
2015/16 60% 70% 76% 83% 85% 83%
2016/17 57% 68% 78% 80% 85% 85%
2017/18 57% 69% 75% 80% 85% 83%
2018/19 59% 69% 76% 79% 82% 83%
2019/20 55% 66% 77% 79% 84% 85%



Extracted Output:

Question: What is the standard deviation of the percentage of population giving to charity for the age group 35 to 49 across all years?

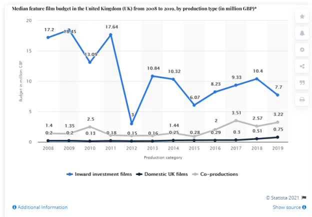
Answer:

```
def std_deviation_age_group_3549():
data = [ {"year": "2013/14", "16 to 24": 74, "25 to 34": 77, "35 to 49": 82, "50 to 64": 84, "65 to 74": 89, "75 and over": 89},
{"year": "2014/15", "16 to 24": 63, "25 to 34": 70, "35 to 49": 79, "50 to 64": 82, "65 to 74": 85, "75 and over": 89},
{"year": "2015/16", "16 to 24": 60, "25 to 34": 70, "35 to 49": 76, "50 to 64": 83, "65 to 74": 85, "75 and over": 83},
{"year": "2016/17", "16 to 24": 57, "25 to 34": 68, "35 to 49": 78, "50 to 64": 80, "65 to 74": 85, "75 and over": 85},
{"year": "2017/18", "16 to 24": 57, "25 to 34": 69, "35 to 49": 75, "50 to 64": 80, "65 to 74": 85, "75 and over": 83},
{"year": "2018/19", "16 to 24": 59, "25 to 34": 69, "35 to 49": 76, "50 to 64": 79, "65 to 74": 82, "75 and over": 83},
{"year": "2019/20", "16 to 24": 55, "25 to 34": 66, "35 to 49": 77, "50 to 64": 79, "65 to 74": 84, "75 and over": 85}
]
information = {"years": [d["year"] for d in data], "age_groups": ["16 to 24", "25 to 34", "35 to 49", "50 to 64", "65 to 74", "75 and over"]}
percentages = [d["35 to 49"] for d in data]
mean = sum(percentages) / len(percentages)
variance = sum((x - mean) ** 2 for x in percentages) / len(percentages)
std_dev = variance ** 0.5
return std_dev
```

Table 13: An example of our coding prompt for GPT-3.5 and an extracted sample

Novel Prompt for GPT-4

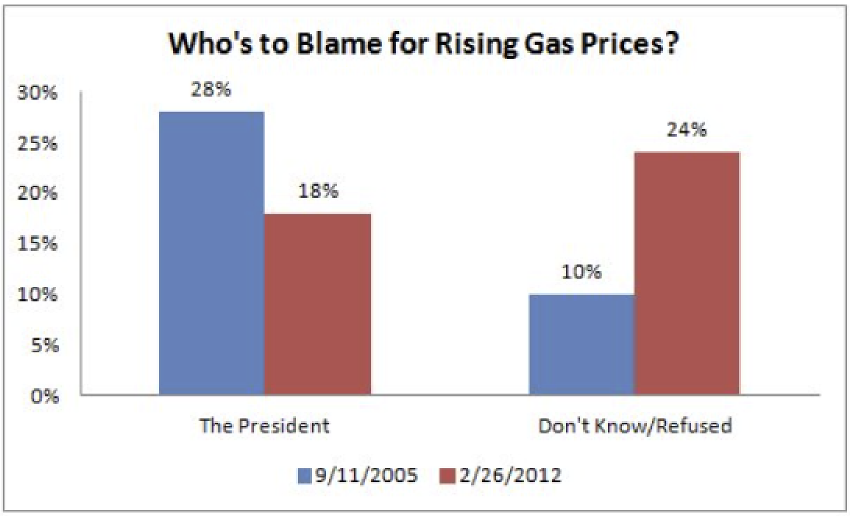
Prompt:



Generate different instruction-tuning tasks for an LLM that we are trying to tune for Chart Understanding. Your response should be in a json format where each example has three fields: task type, input: which only asks a question related to the task type and the given data, and expected output: which is the answer to the input question based on the input information. Use the following chart data to generate 10 unique tasks that do not overlap with two specific tasks: Summarizing the Chart, and single point information retrieval.
Title: Median Feature Film Budget In the United Kingdom (UK) from 2008 to 2019, by production type(in million GBP)
Table:
Production category Inward investment films Domestic UK films Co-productions
2008 17.2 0.2 1.4
2009 18.45 0.2 1.35
2010 13.09 0.13 2.5
2011 17.64 0.18 1.1
2012 3 0.15 1.03
2013 10.84 0.16 1.06
2014 10.32 0.25 1.44
2015 6.07 0.28 0.9
2016 8.23 0.29 2
2017 9.33 0.3 3.51
2018 10.4 0.51 2.57
2019 7.7 0.75 3.22
Extracted Instruction:
"Task: "Range Detection",
"input":What range of median budgets did domestic UK films have over the period?
"expected_output": "Domestic UK films had median budgets ranging from 0.13 million GBP in 2010 to 0.75 million GBP in 2019."

Table 14: An example of prompt for Novel Instruction Generation for GPT-4-preview and an extracted sample

Sample 44)



Input:
Explain the significance of the 2005 data point for 'The President' group in the context of the chart.

Output 1:
The 2005 data point for 'The President' group indicates that 28% of the population blamed the president for rising gas prices in the year 2005.

Output 2:
Overall, the data suggests that the majority of the data point for 'The President' group in the 2005 data point.

Evaluation:

Output 1
Informativeness: 4
Relevance:5
Factual Correctness:5

Output 2
Informativeness: 4
Relevance: 3
Factual Correctness: 2

Figure 15: An example of the human study evaluation setup.