CHART QUESTION ANSWERING WITH AN UNIVERSAL VISION-LANGUAGE PRETRAINING APPROACH

PARSA KAVEHZADEH

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO
AUGUST 2023

Abstract

Charts are widely used for data analysis, providing visual representations and insights into complex data. To facilitate chart-based data analysis using natural language, several downstream tasks have been introduced recently including chart question answering. However, existing methods for these tasks often rely on pretraining on language or vision-language tasks, neglecting the explicit modeling of chart structures. To address this, we first build a large corpus of charts covering diverse topics and visual styles. We then present UniChart, a pretrained model for chart comprehension and reasoning. We propose several chart-specific pretraining tasks that include: (i) low-level tasks to extract the visual elements (e.g., bars, lines) and data from charts, and (ii) high-level tasks to acquire chart understanding and reasoning skills. Our experiments demonstrate that pretraining UniChart on a large corpus with chart-specific objectives, followed by fine-tuning, yields state-of-the-art performance on four downstream tasks. Moreover, our model exhibits superior generalizability to unseen chart corpus, surpassing previous approaches that lack chart-specific objectives and utilize limited chart resources.

Acknowledgements

To begin with, I would like to express my utmost appreciation to my thesis advisor, **Prof. Enamul Hoque Prince**, for his unwavering support since the very beginning of my master's degree and throughout. His invaluable feedback and suggestions have significantly enhanced the quality of this thesis and enabled its submission to a prestigious forum. I am sincerely grateful to him for his mentorship in conducting high-quality research in the field of ML/NLP. Collaborating with him over the span of two years has been a tremendous honor for me.

I extend my gratitude to **Prof. Song Wang** for serving as a committee member for my thesis.

I would also like to extend my thanks to **Prof. Shafiq Joty**, who is one of the co-authors of this work, for providing invaluable feedback and insightful comments. Furthermore, I appreciate the collaboration and assistance of **Ahmed Masry and Do Xuan Long** in this endeavor.

Lastly, I am deeply grateful to my parents for their unwavering support and encour-

agement, as well as to my brother for his constant backing throughout my journey in this graduate program.

Preface

This thesis work was mainly done by **Parsa Kavehzadeh** under the supervision of **Prof. Enamul Hoque Prince**. The work is currently under submission for the EMNLP 2023 conference [1]. My involvement in this project involved participating in brainstorming sessions and problem definition, curating the dataset, conducting post-processing and analysis, running multiple experiments, analyzing the results, and writing the manuscript. My coauthors, **Enamul Hoque** and **Shafiq Joty**, provided valuable feedback and assisted in the writing process. Additionally, coauthor **Ahmed Masry** contributed to proposing the model architecture, running experiments, analyzing the results, and writing the manuscript. Coauthor **Do Xuan Long** contributed to dataset curation and manuscript writing. In another publication [2], where I am listed as the second author, I did a comprehensive literature review on chart question answering domain, while **Prof. Hoque** guided me to provide the survey outline and also write the paper.

1. Ahmed Masry[*1], **Parsa Kavehzadeh**[*], Do Xuan Long, Enamul Hoque, Shafiq Joty,

[1] Equal Contribution.

”UniChart: A Universal Vision-language Pretrained Model for Chart Comprehension and Reasoning”, Submitted for 2023 Conference on Empirical Methods in Natural Language Processing (**EMNLP 2023**).

2. Enamul Hoque, **Parsa Kavehzadeh**, Ahmed Masry ”Chart Question Answering: State of the Art and Future Directions”, Accepted in 24th EuroGraphics Conference on Visualization (**EuroVis 2022**).

Table of Contents

List of Tables

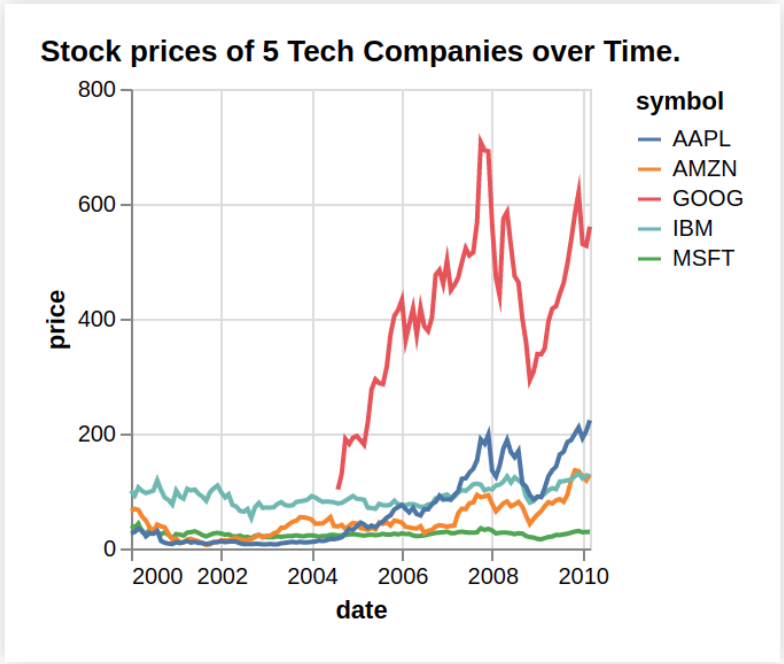
List of Figures

1 Introduction

1.1 Motivation

Information visualizations such as bar charts and line charts are commonly used for analyzing data and making informed decisions. To analyze data, often people ask various complex questions about charts [47]. However, answering such questions about charts is not always easy. In order to answer complex questions, one must have various analytical skills and would need to combine various low-level operations (e.g. retrieve values from bars, find extremes, aggregate values) which can be mentally taxing. For example, to answer the question "When the difference between the Apple and Google stocks was the highest?" in Figure 1.1, one needs to compute the differences between all data points of two red and blue lines and then find the highest one.

Automatic chart question answering is a challenging task because of the richness and ambiguities of natural language and complex reasoning that may be required to predict the answer [96]. The problem is also very inter-disciplinary in nature, falling in the intersection of information visualization, natural language processing, and human



Question: ***When the difference between the Apple and Google stocks was the highest?***
Answer: ***2008***

Figure 1.1: An example of a natural language question about a line chart that shows stock prices of some tech companies over time.

computer interactions. Some early approaches on NLIs and chart question answering applied natural language processing techniques by largely depending on heuristics or grammar-based parsing techniques [96, 103, 36, 27]. Recently, there have been also some attempts for applying deep learning models for understanding natural language queries about visualizations [9, 100, 90]. Several benchmark datasets have also been developed to evaluate the effectiveness of these approaches [43, 45, 9].

A dominant strategy to tackle these downstream tasks is to utilize pretrained models [105, 59, 49, 13] trained on langauge and vision tasks [22]. However, although effective,

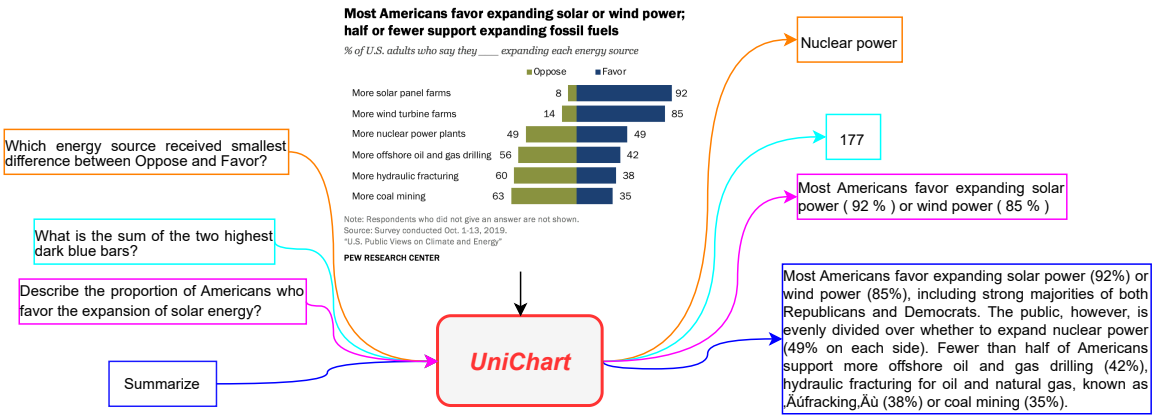


Figure 1.2: UniChart is pretrained on a large chart corpus and is capable to perform various chart comprehension and reasoning tasks.

such models may not be optimal for chart-specific tasks because they are trained on large text corpus and/or image-text pairs without any specific focus on chart comprehension. In reality, charts differ from natural images in that they visually communicate the *data* using *graphical marks* (e.g., bars, lines) and *text* (e.g., titles, labels, legends). Readers can discover important patterns, trends, and outliers from such visual representation [78]. Existing pretrained models do not consider such unique structures and communicative goals of charts. For instance, Pix2Struct [52] is a pretrained image-to-text model designed for situated language understanding. Its pretraining objective focuses on screenshot parsing based on HTML codes of webpages, with a primary emphasis on layout understanding rather than reasoning over the visual elements. MatCha [63] extends Pix2Struct by incorporating math reasoning and chart data extraction tasks, but it still lacks training objectives for text generation from charts and it was trained on a limited number of charts.

1.2 Our Approach

In this work, we present UniChart, a pretrained model designed specifically for chart comprehension and reasoning. We address this goal in multiple steps throughout the project. At the first stage, we provide a large pretraining corpus including 611K charts that we collected from multiple real-world sources. During the curating process, we also augment our corpus with visualization-generation tools such as D3 and Vega-Lite and create charts based on data tables we crawled from online sources. This assist us to manually diversify the visual style of the charts by manipulating the relevant features during generating charts. In general, our corpus has two main category which are (i) charts with available underlying data table and (ii) charts without available underlying data table. For specific pretraining objectives, we utilize datasets from each category, with some objectives requiring gold data tables and others not requiring them.

UniChart is pretrained on a large corpus of charts and it aims to serve as a Universal model for various chart-related downstream tasks including those relevant to question answering domain. Inspired by the model architecture from Kim et al. [48], UniChart consists of two modules: *(1)* a chart encoder, which takes the chart image as the input, and *(2)* a text decoder, trained to decode the expected output based on the encoded image and the text input fed in the decoder as task prompt. This architecture enables us to devise an end-to-end approach for chart understanding tasks which involve natural language

interaction. Kim et al. [48] also pretrained their model on Optical Character Recognition (OCR) objective on millions of document images, which already provides the model with a significant ability in addressing chart question answering task [76].

Various essential capabilities are necessary for a model to effectively address diverse queries posed by users regarding charts. Some questions ask for a single element value, which requires retrieving a value from *(i) underlying chart data table*. Compositional queries which requires *(ii) numerical/visual reasoning* steps on chart elements are also another common type of questions asked by users. Sometimes a single word or phrase cannot satisfy a user query and the model needs to generate *(iii) an explanatory answer* with factually correct statements. Users can also simply ask for *(iii) a high-level brief summary covering important trends and patterns* in the chart. Our pretraining objectives include both low-level tasks focused on extracting visual elements and data from chart images, as well as high-level tasks, intended to align more closely with downstream applications. One key challenge for pretraining was that most charts in the corpus do not come with informative summaries. To address this challenge, we leverage knowledge distillation techniques by promoting LLMs to opportunistically collect a large set of text summaries from charts which are utilized during pretraining. Figure 1.2 demonstrates some samples of different downstream tasks can be addressed by UniChart.

We also conducted extensive experiments and analysis on various chart-specific downstream tasks to evaluate the effectiveness of our approach. Specifically, we evaluated

UniChart on two chart question answering datasets, ChartQA [76] and OpenCQA [46], and found that it outperformed the state-of-the-art models in both cases. For chart summarization, UniChart achieves superior performance in both human and automatic evaluation measures such as BLEU [85] and ratings from ChatGPT [81]. Moreover, UniChart achieved state-of-the-art results in the Chart-to-Table downstream task. Finally, our model showed improved time and memory efficiency compared to the previous state-of-the-art model, MatCha, being more than 11 times faster with 28% fewer parameters.

1.3 Contributions

Our primary contributions are:

(*i*) A pretrained model for chart comprehension with unique low-level and high-level pretraining objectives specific to charts. These objectives range from numerical/visual reasoning and data table extraction to generating comprehensive summaries about charts, which provide the model with capabilities required in different chart-specific downstream tasks.

(*ii*) A large-scale chart corpus for pretraining, covering a diverse range of visual styles and topics. We introduce UniChart pretraining corpus by crawling multiple online sources providing chart data. We employ data augmentation to incorporate hundreds of thousands of charts with diverse visual styles to the corpus. We also utilize state-of-the-art LLMs through a knowledge distillation approach in order to generate high-quality summaries

for the charts in our corpus which do not come with gold summaries.

(*iii*) Extensive automatic and human evaluations that demonstrate the state-of-the-art performance of UniChart on a wide range of chart-specific benchmark tasks. We make our code and chart corpus publicly available at https://github.com/vis-nlp/UniChart.

1.4 Outline

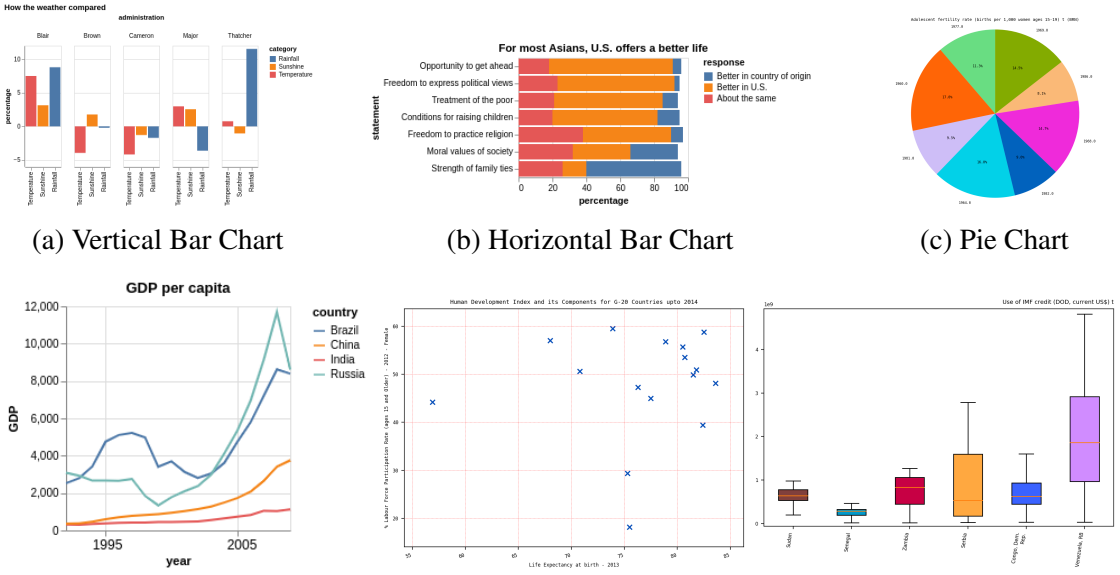
In this section, we provide a brief overview of the organization of the chapters in this thesis. Chapter 2 begins with a review of prior work in chart question answering and vision-language pretraining. Chapter 3 details the process of curating our pretraining corpus and Chapter 4 elaborates the design of our proposed model and our experiments, including the downstream tasks, evaluation metrics, and ablation studies. The conclusion and future work are outlined in Chapter 5.

2 Literature Review

This chapter focuses on two main domains: chart question answering and document-understanding pretraining. First, we extensively explore various aspects of the chart question answering problem by analyzing the input and output dimensions. Additionally, we examine the existing works that try to address the challenges in this domain. Another area we explore is document-understanding pretraining, which is closely related to our pretraining approach. We delve into the latest methods used to create pretrained models that are applicable to different downstream tasks involving document understanding. Moreover, we investigate how these techniques can be effectively applied to enhance chart comprehension.

2.1 Chart Question Answering

In chart question answering, people express their information needs through a variety of textual queries as shown in Table 2.1. Note that these categories are not orthogonal; for example, a question can be factual, visual, and compositional simultaneously.



(a) Vertical Bar Chart (b) Horizontal Bar Chart (c) Pie Chart

(d) Line Chart (e) Scatter Plot (f) Box Chart

Figure 2.1: Examples of different types of input charts (plots a, b, d are adapted from [47] and plots c, e, f are adapted from [9])

2.1.1 Questions as Textual Input

We discuss the different types of textual queries along with how current CQA systems handle them below.

- **Factual vs. Open-ended**: Most existing works focus on answering questions that require **factual** answers (e.g. *"Which country has the highest GDP?"*) [9, 47, 90, 43]. Factual questions require the system to compute the answer by analyzing the question and the chart and then subsequently performing some arithmetic and logical operations (e.g., compare values, find extremes). Several works have attempted to understand and analyze such questions using a combination of heuristic approaches and syntactic

Table 2.1: Different types of question inputs and possible answers. Each question is referring to a particular chart in Figure 2.1, as specified at the end of the question.

Question Types	Answer Types		
	Fixed Vocabulary		
	Numeric		Word(s)/Phrase(s)
Factual	How many years have fertility rate more than 11 percent? 2.1c		Which years have more than 11 percent of fertility? 2.1c
Open-ended	-		-
Visual	What is the median of smallest box? 2.1f		Which administration has the longest blue component?2.1a
Non-visual	What is the median of Sudan? 2.1f		Which statement satisfies Asians most regarding living in U.S.? 2.1b
Simple	What is the percentage of fertility in 1982? 2.1c		Which country experienced the gdp higher than 10,000? 2.1d
Compositional	What is the overall percentage of fertility in 1982 and 1968? 2.1c		Which country has the highest median? 2.1f
	Open Vocabulary		
	Numeric	Word(s)/Phrase(s)	Explanatory Sentence(s)
Factual	How much is life expectancy's mean bigger than labour force's? 2.1e	To which continent Venezuela belongs to? 2.1f	-
Open-ended	-	-	Explain the surge of Russia's GDP in the end of 1990's. 2.1d
Visual	What is the ratio of the longest orange bar to the shortest? 2.1a	To which continent the red colored box belong to? 2.1f	Why Thatcher has the longest blue component? 2.1a
Non-visual	What is the ratio of Sudan's median to Serbia's? 2.1f	What is the population of the country with highest gdp? 2.1d	Why higher life expectancy included higher female labour? 2.1e
Simple	What is the percentage of rainfall in Blair? 2.1a	Was the blue line increasing or decreasing from 2000 to 2005? 2.1d	What is the definition of GDP? 2.1d
Compositional	What is the mean of total life expectancy rates? 2.1e	In which continent is the country with the highest GDP last year? 2.1d	Why is China's GDP less than Brazil's? 2.1d

parsing techniques [96, 103, 79]. Others have focused on leveraging various machine learning approaches for table question answering [47, 75] and feature extraction from chart images [9, 90, 43]. For example, several research works utilize the recurrent neural model (RNN) with Long short-term memory (LSTM) architecture to encode the question [9, 90, 77, 44, 120, 43]. However, more recently transformer architecture has been found to be more effective than RNN for long input sequences, which inspired some researchers

Table 2.2: The table summarizes how existing research papers on CQA cover different categories of input and output dimensions. In the left most column, the papers coded with blue color represent those works that solely focus on CQA, while the ones with magenta color are the works that do cover some question answering aspects but focus more broadly on supporting natural language commands and web-search like queries for interactions with visualizations (such as filtering, sorting, zooming, and highlighting).

Related Papers	Dimensions									
	Input					Output				
	Text			Visualization	Multimodal	Text		Visualization	Multimodal	Conversational
	Factual/ Open-ended	Visual/ Non-visual	Simple/ Compositional	Bitmap/ Data Table		Fixed Vocab Numeric/ Word	Open Vocab Numeric/ Word/ Sentence	New/ Existing		
LeafNet[9]	✓/✗	✓/✓	✓/✓	✓/✗	✗	✓/✓	✗/✗/✗	✗/✗	✗	✗
STL-CQA[100]	✓/✗	✓/✓	✓/✓	✓/✗	✗	✓/✓	✗/✗/✗	✗/✗	✗	✗
DVQA[43]	✓/✗	✓/✓	✓/✓	✓/✓	✗	✓/✓	✗/✗/✗	✗/✗	✗	✗
PReFIL[44]	✓/✗	✓/✓	✓/✓	✓/✗	✗	✓/✓	✗/✗/✗	✗/✗	✗	✗
Kim et al.[47]	✓/✗	✓/✓	✓/✓	✗/✓	✗	✓/✓	✓/✓/✓	✗/✗	✗	✗
FigureNet[90]	✓/✗	✓/✓	✓/✓	✓/✓	✗	✗/✓	✗/✗/✗	✗/✗	✗	✗
Affinity[120]	✓/✗	✓/✓	✓/✓	✓/✗	✗	✓/✓	✗/✗/✗	✗/✗	✗	✗
FigureQA[45]	✓/✗	✓/✓	✓/✓	✓/✓	✗	✓/✓	✗/✗/✗	✗/✗	✗	✗
PlotQA[77]	✓/✗	✓/✓	✓/✓	✓/✗	✗	✓/✓	✓/✓/✗	✗/✗	✗	✗
ChartQA[76]	✓/✗	✓/✓	✓/✓	✓/✓	✗	✓/✓	✓/✓/✓	✗/✗	✗	✗
Eviza[96]	✓/✗	✗/✓	✓/✓	✗/✓	✓	✗/✗	✗/✗/✗	✗/✓	✗	✓
Evizeon[36]	✓/✗	✓/✓	✓/✓	✗/✓	✓	✗/✗	✗/✗/✗	✗/✓	✗	✓
DataTone[27]	✓/✗	✗/✓	✓/✓	✗/✓	✓	✗/✗	✗/✗/✗	✗/✓	✗	✗
Orko[103]	✓/✗	✗/✓	✓/✓	✗/✓	✓	✗/✗	✗/✗/✗	✗/✓	✓	✓
FlowSense[116]	✓/✗	✗/✓	✓/✓	✗/✓	✗	✗/✗	✗/✗/✗	✓/✓	✗	✓
NL4DV[79]	✓/✗	✗/✓	✓/✓	✗/✓	✗	✗/✗	✗/✗/✗	✓/✗	✗	✗
ADVISor[61]	✓/✗	✗/✓	✗/✓	✗/✓	✗	✗/✗	✗/✗/✗	✓/✗	✗	✗
NL2VIS [71]	✓/✗	✗/✓	✓/✓	✗/✓	✗	✗/✗	✗/✗/✗	✓/✗	✗	✗

to adopt such architecture for answering factual questions with charts [100, 75, 76, 61]. Unlike factual questions, **open-ended** questions are exploratory in nature (e.g., *"Why the country with the highest GDP changed in 2016?"*) and typically the expected answer is an explanatory text. Generating such explanatory answers is very challenging and to our knowledge, there is no existing work that explores this problem. The closest works to this problem are the ones that automatically summarize the key insights from charts as text [18, 102, 80, 98]. For example, Obeid and Hoque [80] adopted a transformer-based model to generate an explanatory text describing the chart. Future works may explore

such data-to-text generation approaches by taking the question as additional input.

• **Visual vs. Non-visual**: Visual questions include reference to visual attributes such as *color*, *height*, and *length* of graphical marks (e.g., *bars*) in the chart. In contrast, non-visual questions do not contain such references. For instance, the question *"Which bar is the longest?"* is a visual question since it is referring to the *length* attribute of the mark type *bar*. Kim et al. [47] handled the visual question by translating it to an equivalent non-visual question through a pipeline that first recognizes all phrases in the input question that refers to marks and their visual attributes and then replaces these phrases with equivalent non-visual terms. Then they pass the question and the data table of the chart to a table parsing model to obtain the answer. Others have extracted visual features from the chart and then build classification models to answer the visual questions [45, 43].

• **Simple vs. Compositional** This categorization is based on the complexity of the analytical tasks [3] that the system needs to perform to answer the given question. To answer **simple** questions, the system needs to complete a single analytical task (e.g., retrieving a value). For instance, the question *"What percentage of people are Real Madrid's fan?"* is a simple question. On the other hand, **compositional** questions involve multiple mathematical/logical operations like *sum*, *difference* and *average*. Compositional questions are more challenging as the model needs to combine multiple operations together. For example, to answer this question "How many students achieved over 90%?" from a bar

chart that shows the grades of students, the system needs to perform a *filtering* operation to find students who achieved over 90% followed by applying a *count* operation. To handle the compositional questions, some researchers have applied a compositional semantic parsing technique called Sempre [83] that was originally trained on a table question answering dataset [47, 116]. However, solving compositional questions still remains a challenging task as the accuracy is still very low for such questions. For example, Kim et al. [47] found the accuracy of their model to be only 37% for compositional questions.

2.1.2 Charts as Visual Input

The input visualization to the CQA system can have different types and storage format. Below we discuss how current CQA methods process the input visualization.

• **Chart Types:** Most existing works on CQA focused on limited chart types. Bar charts [9, 90, 44, 120, 77], Line charts [9, 44, 120, 77], Pie charts [9, 90, 44, 120], Box plots [9], Scatter plots [9, 77] are most common chart types used in recent related works. Figure 2.1 shows examples of chart types. None of the CQA works in Table 2.2 (coded with blue color) supports map chart even though they are commonly used. However, some NLI systems such as Eviza support natural language queries with map charts [96]. Less common and unconventional chart types such as Parallel coordinates plot and radar charts are not supported in existing works.

• **Storage Format:** Storage format is another important aspect of the input chart. Charts

can be stored in different formats such as SVG vs. bitmap image. Several works take charts in bitmap image format which are more challenging to handle because the model does not have access to the underlying data table. They mainly utilized computer vision techniques to extract features from a bitmap chart image and then applied classification models [45, 9, 77, 90, 100, 44]. For example, FigureQA [45] extracts the features of the chart using a CNN. The problem with such CNN-based feature extraction is that the model treats the chart like a natural scene without extracting individual graphical marks (e.g., bars) and the underlying texts (e.g, x-axis-labels) in the chart. To overcome this limitation, others attempted to parse the chart by employing object detection models such as Mask-RCNN [32] to detect visual and textual elements [9, 100]. Subsequently, Optical Character Recognition (OCR) models are applied in order to dynamically encode the chart-specific tokens in the question in terms of the positional information of the textual elements in the chart image (e.g., 'x-axis-label-1', 'x-axis-label-2'). However, such dynamic encoding technique is prone to OCR errors and fails when the question refers to the chart texts using synonyms (e.g., 'US' vs 'United States').

One key question here is how the model should reason over both the question and the visualization together to generate the answer? FigureQA used Relation Network (RN) [92] to capture the relation between visual and textual features by concatenating all pairs of object representations from the chart image provided by a CNN with the question features [45]. However, the number of object pairs in RN can be very large, resulting in

computational inefficiency and restricting the performance. Some researchers attempted to address the problem either by making relation features in RN more concise [120] or by dividing the problem into various sub-tasks [90]. Others captured inter-relation between the chart and question features by applying attention mechanisms [9, 43] or through fusing the low and high level features from the chart image and the question in parallel to facilitate multi-stage reasoning process. However, these models ignore the chart structure (e.g., relationships between axis labels, bars, and legends). STL-CQA [100] attempted to better capture the relation between the chart element and the question using a multi-modal transformer-based model, which can better exploit the structural properties by learning the intra-modality relationships (among the chart elements) and cross-modality relationships (between the question and the chart elements)[107].

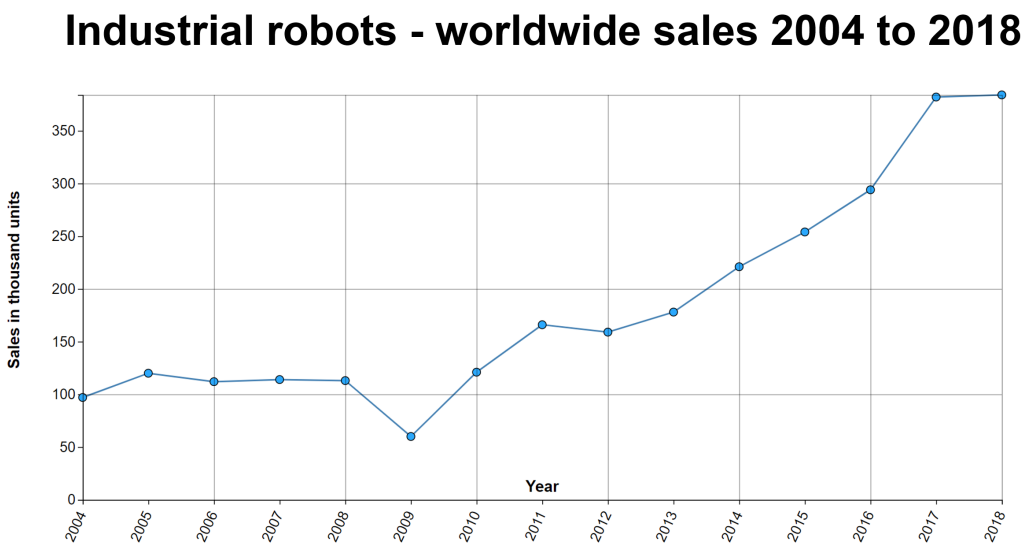
A common limitation of all the above models is that they simply utilize the image features without extracting the underlying data table and visual encoding information of the given chart, which makes it difficult for their models to answer complex visual and compositional questions. Moreover, since they are classification based models, they could only handle fixed-vocabulary type answers and can not apply mathematical operations on the chart data.

In contrast to the above body of work, some systems simplify the problem by assuming that the underlying data and visual encodings of charts are available [96, 103, 36, 61, 79]. Kim et al. [47] propose a pipeline that recovers the data table and encodings from an

input chart to derive the corresponding Vega-Lite specification [93]. It then extracts the data and transforms into a flat relational table and passes it to a table question answering model. If the data can be extracted from the chart, leveraging the pretrained transfer-based model for table question answering such as TaPas [34] or converting natural language question to SQL query [117] could be effective for CQA. However, directly applying table question answering method on chart data is not enough, rather it is necessary to effectively combine the chart features with the data table. Masry et al. [76] takes an initial step in this direction by extracting both the visual features extracted using ViT [21] and the underlying data table of the chart encoded by TaPas [34] and combining them using a cross modality encoder to infer the answer. Still, their model feeds the visual features and the data values separately to their model and does not relate between them. A promising direction here could be to create a better representation of the chart that combine and relate between the visual features and the data table is needed to solve complex visual reasoning questions.

2.1.3 Answers as Textual Output

Most CQA models focus on answering questions that require a textual answer. Usually, such an answer is either a **numeric token** or a **word/phrase** [9, 43, 45, 44]. Such textual outputs can be categorized into **fixed vocabulary** vs. **open vocabulary**. In case of the fixed vocabulary output setting, the answer comes from a small fixed-sized



Output: This statistic shows the total Industrial Sales in the worldwide from 2004 to 2018. In 2018, about 384 thousand units of Industrial were robots in the worldwide, up from 120 thousand units in 2005.

Figure 2.2: Example outputs of an automatic chart summarization model. The textual summary attempts to explain important insights from the chart such as trends and extreme values [80].

vocabulary (*e.g.,* chart axis labels, bar values, 'yes', 'no'). For example, early systems outputs only `yes` or `no` to the questions about the chart [90, 45]. Some systems support other possible output textual elements in chart and certain numeric values as they add them to the fixed vocabulary [43, 44, 9, 100, 120]. These models that only support *fixed vocabulary* questions usually treat the task as a classification problem and rely on dynamic encoding techniques where the questions and answers are encoded in terms of spatial positions of chart elements (*e.g., x-axis-label-1*, *x-axis-label-2* and so on). Such approaches do not produce correct outputs when the OCR model for extracting text from a

chart generates errors or when the question refers to chart elements using synonyms (e.g., `US` vs. `United States`). Overall, treating the CQA problem as a classification task with a fixed vocabulary output is very limited for practical purposes, as it ignores many complex reasoning questions where the answer is derived through various mathematical operations such as aggregation and comparison.

Open vocabulary questions could be more challenging as the system no longer treats the problem as a classification task and instead needs to derive the answer through analytical operations. However, the works focusing on generating open vocabulary outputs are still very limited [47, 77, 75, 36]. Most of these works apply a table question answering model named Sempre [83]. However, Kim et al. simplify the problem by assuming that the underlying data table is available [47] where others try to extract the data using computer vision techniques [77, 75]. One limitation with PlotQA is that after automatically extracting the data table it directly applies the Sempre [83] while answering questions which does not consider any visual features of a chart. As a result their model accuracy can be low for visual questions that makes references to graphical marks and their attributes in chart.

Moving beyond outputting a word/phrase, an open-ended question that requires explanatory sentences or paragraphs has not been supported by existing works. For example, to answer the question, *"How has the GDP of Brazil changed over time?"* from the line chart Figure 2.1 (d), the output needs to be an explanatory paragraph summarizing

the major trends. Recent works on automatic caption generation from charts (e.g., [80, 98] can serve as a starting point for generating explanatory answers (see Figure 2.2). Another way a descriptive sentence/paragraph might be helpful is by explaining how the model computes the answer to enhance the transparency of the model to the user. Kim et al. took the intermediate logical form representation of the table question answering system and then translated it into an explanation using pre-defined templates by applying a rule-based approach [47]. More recently, Kantharaj et al. also introduced a new benchmark OpenCQA containing chart-relevant questions and their explanatory answers annotated by humans [46]. Overall, generating explanatory answers to a question about charts still remains extremely under-explored.

2.1.4 Challenges and Opportunities

• **Constructing more benchmark datasets**: Most existing CQA datasets mainly provide template-based questions and synthetically plotted chart images which lack realism. To address this key limitation, it is necessary to develop human authored questions and real-world chart sources with a variety of chart types and visual styles. Human-authored questions can pose several challenges for the CQA models due to the rich semantics, language variations, informal languages and typos. Moreover, there are several types of questions from Table 2.1 that are underexplored in the existing datasets. While the PlotQA dataset [77] supported open-vocab questions, they were mostly data-related questions

without focusing on questions that refer to the visual attributes (e.g., *color*, *position*) of the chart elements. Similarly, while there have been some work on conversational interfaces for visualizations [94], there is no existing dataset that would allow us to train and evaluate conversational question answering interfaces quantitatively. Constructing new datasets on such under-explored problems could open up exciting new avenues of chart question answering.

• **Leveraging more advanced deep learning models**: There are significant rooms for improving the existing methods for chart question answering. Many natural language interfaces for charts often rely on simple heuristics that fail to understand questions that are compositional and require a deeper understanding of the syntactic, semantic, and pragmatic aspects [96, 36, 103, 116]. This is a serious limitation, as the user becomes very frustrated if the system frequently fails to understand the query correctly. In contrast, existing works that solely focus on chart question answering apply classification techniques on chart images [45, 43, 9] or table question answering on the data table of the chart [77, 47]. As we discussed in Section 2.1.3, both approaches are limited in handling many real-world questions. To address the limitation, future models need to effectively combine both the visual features and the data values of a chart in a unified representation. In this regard, adapting models from the Vision-Language domain[107, 21, 106, 13, 56], which unify visual and textual features effectively through transformer-based models can be a starting point.

• **Addressing chart data extraction challenges**: Many CQA tasks involving perceptual [29] and arithmetic reasoning with charts, which require accurate extraction of chart data and visual encodings (i.e., how data is mapped to visual attributes of graphical marks). Many natural language interfaces assume that the underlying data table and visual encodings of the charts are readily available [96, 103, 36, 97]. This assumption becomes invalid for most real-world charts on the Web which are available in bitmap image format without the underlying data. For other charts that are available in SVG format (e.g., D3 charts), there are some techniques for extracting data and visual encodings, however they are also error-prone in some situations (e.g., cannot accurately extract map chart data) [30, 35].

In general, chart data extraction from bitmap images is more challenging and requires us to apply computer vision techniques to automatically extract the underlying data. PlotQA [77] attempts to use automatically extracted data from chart images to perform question answering, however, it suffers from low accuracy due to data extraction errors. The key problem here is that existing solutions for chart data extraction are quite limited. Some of them are not fully automatic [94, 42] which is not very useful for practical purposes. Some focused on recovering color encodings from various charts [84, 118]. Others provided automatic solutions but they rely on various heuristics which do not work for many real-world charts and the performance is still not high enough [14, 64]. ChartOCR automatically extracts data from real-world charts with reasonably high accuracy [70]

but the model only predicts the raw data values of marks (e.g., bars) without associating them with their corresponding axis or legends. Overall, current approaches for automatic data extraction are usually modular and rely on assumptions which are error-prone. An end-to-end deep learning approach could help improve the performance and generalize well to different chart styles.

2.2 Chart and Document Understanding Models

Pretrained models have dominated in many vision and language tasks [22]. Building a pretrained vision-language model typically involves three steps. First, textual input is usually encoded using BERT-based encoder [68, 87, 55, 57]. Second, for the input image, some prior studies utilize Fast-RCNN [91] to encode the sequence of object regions as the image features [58, 68, 12]. However, this method may neglect some crucial regions in an image. Recent approaches favor encoding the image as a whole [39, 40, 55, 57] by using ResNet [31] or ViT [21]. Third, to fuse the textual and visual features, prior work mostly either designs a fusion encoder [107, 105, 13, 49] or a dual encoder [87, 41, 57]. Finally, multiple common cross-modal pretraining tasks have been designed such as image-text matching [12, 54], cross-modal contrastive learning [87, 41] and generation tasks such as visual question answering [13, 112].

Our work is also related to multimodal document understanding tasks that involve analyzing the textual content, layout, and visual elements of documents [115, 114, 111,

38, 48, 109]. These tasks can be addressed using encoder-only and encoder-decoder architectures. Encoder-only models rely on OCR engines to extract text from document images and use BERT-like encoders augmented with specialized embeddings to encode layout and visual features [115, 114, 111, 38]. In contrast, encoder-decoder architectures combine transformer-based encoders with autoregressive text decoders for text generation tasks related to documents [109, 48, 52]. While Tang et al. [109] incorporates an OCR tool to supplement the vision encoder, Kim et al. [48] and Lee et al. [52] operate in an end-to-end manner without external OCR engines.

More specifically, Lee et al. [52] introduced a model called Pix2struct, which focuses on visual language understanding. Pix2struct adopts an image-encoder text-decoder approach based on the Vision Transformer (ViT) architecture [21]. One key modification in Pix2struct is that it maintains the aspect ratio of input images, unlike ViT, which converts images to a fixed resolution before extracting fixed-size patches. Pix2struct instead scales the images up or down, ensuring the preservation of the width-to-height ratio, and then extracts the maximum number of patches. This approach allows the model to capture important ratio-related information in document images, which is crucial for various document understanding tasks and also enables better adaptation to downstream tasks involving higher resolution input images. To perform pretraining, Pix2struct leverages 80 million screenshots of webpages. During pretraining, the webpage's HTML code is used as a representative structure for the corresponding screenshot. Following the paradigm of

Bart [53], Pix2struct masks 50 percent of the text both in the decoder and in the screenshot. This pretraining strategy combines OCR and language modeling capabilities in Pix2struct. The unified encoder-decoder architecture of Pix2struct simplifies fine-tuning on various downstream tasks in the document understanding domain. For tasks requiring textual input, the text is rendered on top of the image and then fed into Pix2struct during fine-tuning. Pix2struct achieves state-of-the-art performance on most document understanding benchmarks.

Donut [48] is another a comprehensive encoder-decoder architecture that can be directly used for text generation tasks in document-related contexts. It comprises an image encoder based on the Swin Transformer architecture [67] and a text decoder initialized with Bart weights [53]. The main objective of Donut's pretraining phase is to extract text from over 11 million document images in a top-left to right-bottom manner, eliminating the need for additional OCR approaches during fine-tuning. For document understanding tasks like document image classification, form understanding, and visual question answering, Donut utilizes a model consisting of a transformer encoder and an autoregressive text transformer decoder. Initially, the document image is processed by a transformer-based image encoder, and then the textual input is provided as a prompt to the decoder, which generates the corresponding output based on the input. In line with the latter approach, our model adopts an end-to-end encoder-decoder architecture [48].

In general, the above work focuses on training on large image-text pairs or text corpus,

lacking focus on chart understanding. One exception is MatCha [63], a pretrained chart model based on Pix2Struct [52], which achieved SoTA on chart question answering and summarization tasks. MatCha incorporated chart rendering and data table extraction as specific objectives during the pretraining phase. To teach Pix2struct numerical reasoning, they utilized the MATH [95] and DROP [23] datasets. Since these datasets were designed for text-to-text tasks, the authors first rendered the questions and textual input, and then provided the resulting image to the Pix2struct backbone to obtain the output. The major problem with MatCha is that its pretraining tasks mainly focus on data table generation without focusing on text generation tasks. The model is also pretrained with reasoning tasks using the textual datasets which might limit its visual reasoning ability. Our model is trained on a larger corpus with chart-specific pretraining objectives, including visual reasoning and text generation, making it more versatile for various chart-related tasks.

2.3 Discussion

In general, the above work focuses on training on large image-text pairs or large text corpus without any specific focus on chart-understanding tasks. One notable exception is MatCha [63], which is a pretrained chart model based on Pix2Struct [52] and achieved SoTA on chart question answering and summarization benchmarks. However, their pretraining tasks mainly focus on data table generation from a limited number of charts without focusing on text generation tasks that are important for tasks such as OpenCQA

[46] and Chart-to-Text [98]. The model is also pretrained with reasoning tasks using the textual datasets (MATH [95] and DROP [23]) which might limit its visual reasoning ability against chart-specific questions. In contrast, our model is trained on a much larger chart corpus with chart-specific pretraining objectives including visual reasoning and text generation for charts. As such, our model is more suitable to serve as a general-purposed model with more diverse chart-specific capabilities gained in pretraining.

3 Chart Pretraining Corpus Construction

As we discussed in last chapter, one of the major drawbacks in existing CQA benchmarks is that their charts lack visual and style diversity. Moreover, the questions in these datasets are usually simple template-based queries that are related to simple tasks like retrieving a data value. More complicated questions like compositional and visual referring questions (color, position) have been relatively unexplored. As a result, we decided to provide a large chart pretraining corpus which contains various visual styles in charts and also different types of questions related to these charts in order to prepare the model for chart downstream tasks including chart question answering.

At the first step, the large corpus of charts that we curated for the pretraining phase will be introduced. In following sections, we will elaborate how we collect, generate, and filter charts from diverse sources to reach the final version of corpus. At the last step, we will have an extensive analysis about different aspects of our proposed corpus.

To build a large and diverse corpus with various styles, topics, and storage formats, we crawled charts from various online sources. Additionally, we utilized publicly available

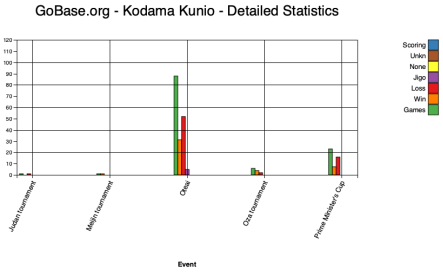
chart datasets suitable for pretraining. The collected charts can be categorized into two types: charts with underlying data tables and charts without data tables.

3.1 Charts with Data Tables

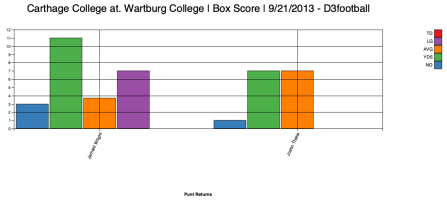
Charts with an underlying data table are collected in three ways: (*i*) utilize existing datasets, (*ii*) extract SVG charts, and (*iii*) data augmentation.

• **Utilize existing datasets** Our goal was to train the model based on real-world data, thus, we did not consider the ones that are generated from synthetic data [43, 45]. In particular, we used the following five chart datasets for which the underlying data tables were available: (*i*) Statista (statista.com) [98], (*ii*) Our World In Data or OWID (ourworldindata.org) [76], (*iii*) Organisation for Economic Co-operation and Development or OECD (oecd.org) [76], (*iv*) PlotQA [77], and (*v*) a subset of the ChartInfo [1] dataset that provides bounding box annotations for data encoding marks (e.g., bars in a bar chart).

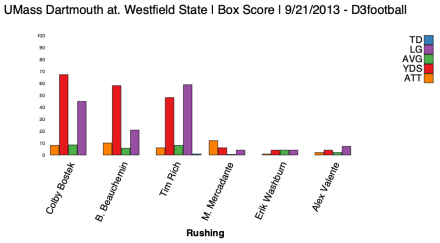
• **Extract SVG charts** We also used some charts in SVG format that do not come with data tables, but extracting the data is relatively straightforward from the SVG elements. We used the Chartblocks and Plotly datasets from the Beagle corpus [5] as they generally have better quality charts (e.g., meaningful titles and axis labels). The steps for preparing these charts are: (1) identify axis labels and legends using specific class names of html



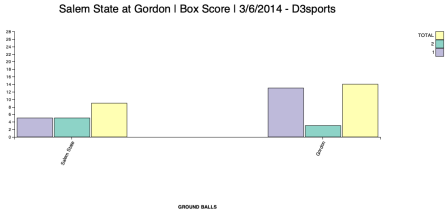
(a)



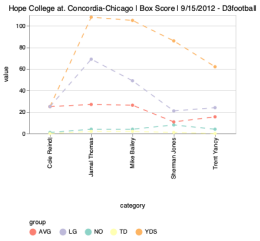
(b)



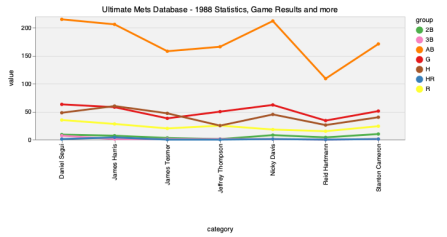
(c)



(d)



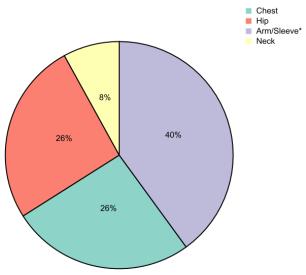
(e)



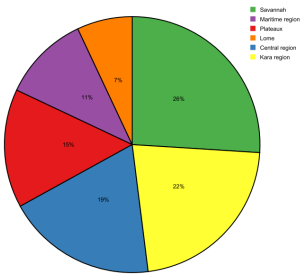
(f)

Authentic Collection Tech Fleece

How to call Togo from Malawi | NobelCom



(g)



(h)

Figure 3.1: Visually diverse charts generated by D3 and Vegalite for WDC corpus (Figure 3.1a, Figure 3.1b, Figure 3.1c, Figure 3.1d, Figure 3.1g and Figure 3.1h from D3-WDC, Figure 3.1e and Figure 3.1f from Vegalite-WDC. Visual factors like color scheme, width of bars, and existence of grids and axis labels are different among the samples.

attribute, (2) extract bounding boxes of chart elements (e.g., bars, line) using SVG attribute properties (e.g., `size` and `location` of `<rect>`, (3) construct the underlying data table by iterating through each of the `<g>` elements to find data values of each data attribute encoded in the given chart. When data labels are absent, we utilize the scale information based on the axis labels and tick marks (`height` of `<rect>`) of the chart and the bounding box information of data encoding marks to recover the data values.

• **Data augmentation** We further augmented the corpus by creating charts from publicly available data tables. We used the The Web Data Commons (WDC) [2], which used Common Crawl[2] to collect a large amount of structured data. The charts are created in the following steps:

1. *Data pre-processing:* Since many tables in WDC contain more than three columns, we decomposed so that tables are suitable for creating desired chart types (e.g., bars, lines, and pie charts). In particular, we automatically analyze the data type of each column (e.g, numeric vs. categorical) and then randomly choose one column with numeric data values and one/two column(s) with categorical data. We also limit the maximum number of rows of the table to 8 so that the corresponding chart can fit within reasonable screen space.

2. *Chart generation*: To generate visually diverse charts, we used the D3 [8] library that

[2]https://commoncrawl.org/

provides great flexibility in terms of creating diverse visualization styles. We also employed Vega-Lite [93] which creates charts based on declarative JSON syntax. We used simple heuristics for determining chart types from the data table [73]. We created four types of charts: (1) vertical simple bar charts with one numeric data column, (2) vertical grouped bar charts, (3) pie charts, and (4) line charts (both single series and multi-series).

3. *Visual diversification*: To create visually diverse charts resembling real-world variations, we manipulated the following visual style properties: (1) **Colors and shapes**: Color schemes from ColorBrewer[3] and Tableau[4] were chosen for categorical data attributes. We also varied shape properties such as bar thickness, line types (e.g., continuous vs dotted), and legend shape types (e.g., rect, circle). (2) **Position and distance**: We also varied bar positions and distances with respect to axis labels. (3) **Guides**: Charts may contain additional guides such as grids, so we generate charts with and without grids to diversify styles.

Figure 3.1 depicts a visually diverse set of charts created using this augmentation process. In total, we created a total of 189,839 charts (Table 3.1).

[3] https://colorbrewer2.org/

[4] tableau.com

3.2 Charts without Data Tables

Many charts that are available online are in image format where the corresponding data tables are not available. However, they can be still useful in large-scale pretraining as we can extract chart elements and rich textual contents (e.g., titles, surrounding texts, captions) using object detection and optical character recognition (OCR) techniques. We collected image chart datasets such as LineCap [74] and Neural Caption Generation [102] since they provide high-quality summaries. We also used the Pew dataset from [98] and further augmented it with additional 1K charts that were published between Jan 2021 to Nov 2022. Finally, we used the ExcelChart400K dataset [70] which only provides bounding boxes without underlying data tables. We also considered other existing image chart datasets such as Vis30K [10] and VisImage [19] which are scraped from scientific publications but are not very suitable as they tend to have poor resolution and lack meaningful textual content (e.g., no titles). Another visualization collection [11] provides images with multiple views which are not suitable for our intended pretraining model.

3.3 Augmentation by Knowledge Distillation for Text Generation Tasks

Chart-related downstream tasks such as and open-ended question answering [46] and chart summarization [98] require generating informative and relevant texts. However, for

most of the charts in the pretraining corpus, there are either no associated summaries or the summaries that are collected opportunistically such as the Statista dataset [98] lack quality (e.g., too short and not very informative). Training on such substandard "ground-truth" summaries can negatively affect the overall model performance as shown in text summarization [51, 17]. Indeed, Goyal et al. [28] and Liu et al. [66] have recently shown that human raters prefer summaries generated by LLMs, especially the ones that are instruction-tuned such as InstructGPT [82], compared to the reference summaries in various text summarization datasets. Consequently, the instruction-tuned LLMs have been successfully used as an annotator in several recent studies [20, 86].

Inspired by these findings, we leveraged InstructGPT to generate coherent and relevant text. Specifically, we prompted `text-davinci-003` by providing the underlying data table as input and one exemplar (i.e., 1-shot in-context learning). Since generating summaries for thousands of charts by calling OpenAI API is quite costly, we devised a knowledge distillation approach. We first used `text-davinci-003` to create a small dataset of 3700 summaries for different chart types.

We select the small dataset (3700 charts) from PlotQA and the augmented charts from WDC, since these datasets are accompanied by the underlying data tables which serve as suitable chart representation for LLMs. Also, they cover a wide range of topics which contributes to the diversity in the generated summaries (Section 3.1). Figure 3.2 shows our process for generating summaries for the charts that have underlying data tables using

InstructGPT model [82]. The input mainly consists of one demonstration (table-caption pair) followed by the desired chart data table. The output is the generated summary. Using this mechanism, we generated a small dataset of 3,700 samples. We then finetuned Flan-T5 XL [16] on this dataset. To our knowledge, Flan-T5 was the SoTA open-sourced instruction-tuned model during the development of our dataset. After finetuning on our task, we (qualitatively) observed similar performance as `text-davinci-003`. At the final step, we used the finetuned Flan-T5 model to generate summaries for all the charts that do not have an associated summary (e.g., PlotQA, augmented charts, OWID and OECD charts). In this process, we added around 454K summaries for charts in our pretraining corpus. Figure 3.4 shows some examples generated by the finetuned Flan-T5.

To benefit more from the capability of GPT models in generating high-quality summaries, we further prompt ChatGPT (`gpt-3.5-turbo`) [81] to generate summaries for the charts from Statista and Pew Research and put these in our pretraining corpus instead of the original summaries in the Chart-to-Text benchmark [98]. In most cases, we found the summaries from ChatGPT to be more elaborate and of better writing style. For the Pew Research Centre charts, the underlying data tables are not provided. However, we have observed that the underlying data values are written on the visual elements in most of these charts. Hence, we decided to use an OCR tool to extract the layout-preserving texts from the chart images, and then feed it into ChatGPT to generate the summaries as shown in Figure 3.3. We realized that ChatGPT is capable of understanding a chart from

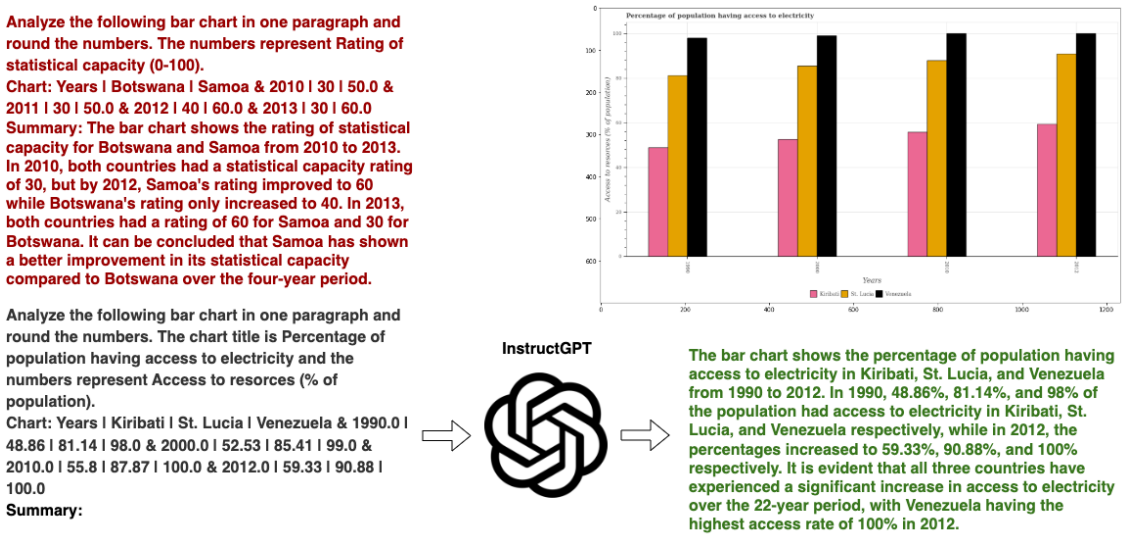


Figure 3.2: An example of the performance of *InstructGPT* in generation summaries for data tables. On the left side, the red text is a full example of a demonstration and its summary followed by the demonstration for the target chart. The paragraph in green shows the summary generated by the model.

the OCR data.

3.4 Datasets Analysis

Our chart pretraining corpus has over 611K charts covering a diverse range of bar charts, line charts, and pie charts (Table 3.1). Data tables of *Simple* charts have two columns (simple bar charts or single-series line charts), whereas *Complex* charts involve at least three columns (e.g., stacked or group bar charts, line charts with multiple lines). The first two chart groups in Table 3.1 come with an underlying data table which cover over 80% of the corpus. The bottom group contains five datasets which only provide charts

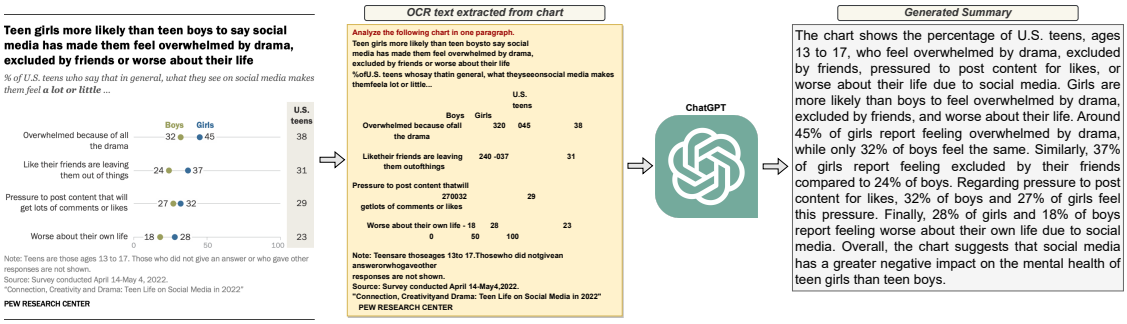
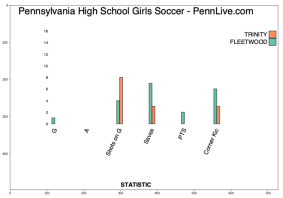


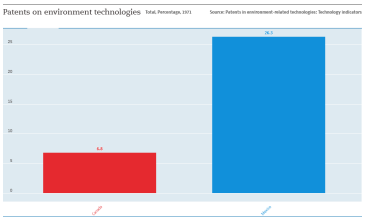
Figure 3.3: An example of the layout-preserved OCR-extracted text for a PewResearch chart image where the underlying data table is not available. The extracted text is then given to ChatGPT to generate a summary. ChatGPT can still extract and comprehend important information and insights from the layout-preserving text of the chart image.

Table 3.1: Chart type distribution and linguistic statistics of the chart pertaining corpus. The charts in the last group (magenta) do not come with an underlying data table. The charts generated by the data augmentation process are shown in blue.

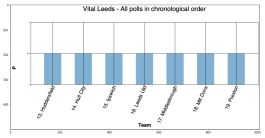
Datasets	Chart Type						Linguistic Statistics		
	Bar		Line		Pie	#Charts	#Vocab	Avg. Character	Avg. Token
	Two-Col	Multi-Col	Two-Col	Multi-Col					
Statista	71.9%	15.3%	8.3%	2%	2%	**19,143**	24,392	111.37	21.88
OWID	51.9%	0.0%	9%	38.9%	0.0%	**60,624**	3,721	85.89	16.96
OECD	49.1%	0.0%	3.1%	47.7%	0.0%	**64,380**	1,606	65.47	14.67
PlotQA	11.2%	55.6%	6.7%	26.2%	0.0%	**157,070**	2,230	155.32	33.21
Beagle	29.8%	27.3%	24.7%	17.9%	0.0%	**3,972**	11,361	78.76	20.55
ChartInfo	31.7%	51.0%	8.6%	8.6%	0.0%	**1,796**	13,329	120.75	26.11
Data Augmentation	13.3%	49.3%	11.7%	11.1%	14.3%	**189,836**	117,244	85.62	21.16
ExcelChart400K	11.5%	32.7%	12.0%	22.3%	27.7%	**106,897**	515,922	138.68	27.72
PewResearch	11.4%	55.5%	4.4%	21.9%	6.5%	**5,295**	38,165	477.33	98.08
LineCap	0.0%	0.0%	15.9%	84.0%	0.0%	**2,821**	16,570	102.11	24.62
Neural Caption	0.0%	0.0%	100%	0.0%	0.0%	**100**	389	117.56	27.43
Total	**21.95%**	**36.56%**	**9.23%**	**23.71%**	**9.39%**	**611,934**	**888,522**	**114.70**	**25.01**



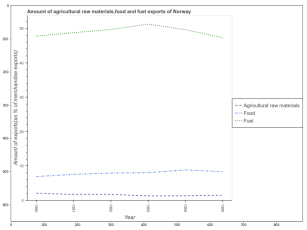
The bar chart compares the performance of two high school girls soccer teams, Fleetwood and Trinity, in a game. Both teams had a goal difference of 1.0, with Fleetwood scoring 1.0 goal and Trinity scoring 0.0. Both teams had 0.0 in shots on goal, saves, points, and corner kicks. However, Fleetwood had 4.0 shots on goal, 7.0 saves, 2.0 points, and 6.0 corner kicks, while Trinity had 8.0 shots on goal, 3.0 saves, 2.0 points, and 3.0 corner kicks. Overall, the chart shows that both teams had similar performances in the game, with Fleetwood scoring 1.0 goal and Trinity scoring 0.0.



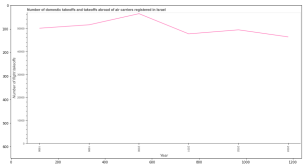
The bar chart compares the number of patents on environment technologies between Canada and Mexico. Canada has 6.8 patents on environment technologies, while Mexico has 26.3. This indicates that Mexico has more than double the number of patents on environment technologies compared to Canada. This suggests that Mexico is more active in the field of environment technologies.



The bar chart shows the results of the Vital Leeds poll in chronological order for seven teams: Huddersfield, Hull City, Ipswich, Leeds United, Middlesbrough, MK Dons, and Preston. All teams had a score of 0.0 in the poll, indicating that they had not been voted on yet. This suggests that the team was not yet considered to be a clear favorite in the poll, and that the results of the poll may be skewed by the outcome of the next few games.

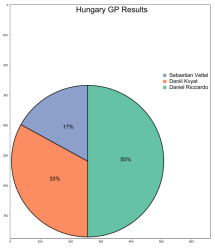


The line chart shows the amount of agricultural raw materials, food, and fuel exports of Norway from 1990 to 1995. Agricultural raw materials exports decreased from 2.18% in 1990 to 1.32% in 1993. Food exports increased from 6.94% in 1990 to 8.85% in 1994. Fuel exports increased from 47.78% in 1990 to 51.26% in 1993. Overall, the trend shows a decrease in the amount of agricultural raw materials, food, and fuel exports of Norway from 1990 to 1995.



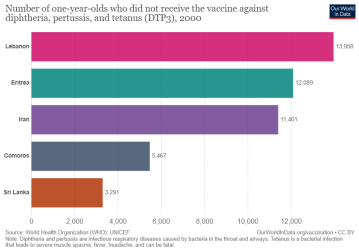
The line chart displays the number of domestic takeoffs and takeoffs abroad of air carriers registered in Israel over a six-year period from 1998 to 2003. The number of flight takeoffs started at 50,100 in 1998, increased to 51,010 in 1999, then decreased to 56,427 in 2000, decreased further to 47,648 in 2001, increased to 49,404 in 2002, and decreased again to 46,334 in 2003. Therefore, the chart depicts a fluctuating trend of the number of flight takeoffs of air carriers registered in Israel during the six-year period, with an overall decreasing trend from 1998 to 2003.

The line chart shows the statistical performance of individual players from Delaware Valley and Widener in a football game on November 10, 2012. The categories measured are number of carries (NO), yards gained (YDS), average yards per carry (AVG), longest gain (LG), and touchdowns scored (TD). Among the players, Lewis Vincent had the most carries (5.09) and yards gained (64.09), with an impressive average of 13.09 yards per carry. Chris Ruiz also had a strong performance, with an average of 8.42 yards per carry and 0.42 touchdowns scored. Meanwhile, Mike Anusky had the fewest carries (1.42) and yards gained (6.09), but still managed to score a touchdown. Overall, the chart shows that the players from both teams had varying levels of performance in the game.



The pie chart shows the results of the Hungarian Grand Prix motor race in 2020, categorized by the winner of the race. The winner of the race was Sebastian Vettel, who accounted for 17% of the total results, followed by Daniil Kvyat at 33%, and Daniel Ricciardo at 50%. This chart indicates that the race was closely contested, with all three drivers finishing within a close margin of each other.

The line chart displays the number of non-fatal injuries from terrorist attacks over a nine-year period from 1970 to 1982. The x-axis shows the number of non-fatal injuries, while the y-axis shows the number of fatalities. The number of non-fatal injuries increased from 4.23 in 1970 to 271.09 in 1980, before decreasing to 42.18 in 1981. Overall, the chart shows an increasing trend in the number of non-fatal injuries from terrorist attacks during the nine-year period, with a slight decrease in 1978.



The bar chart shows the number of one-year-olds who did not receive the vaccine against diphtheria, pertussis, and tetanus (DTP3) in 2000 for five countries: Lebanon, Eritrea, Iran, Comoros, and Sri Lanka. Lebanon had the highest number of one-year-olds who did not receive the vaccine with 13,958 while Eritrea had 12,089 and Iran had 11,401. Comoros had 5,467 and Sri Lanka had 3,291. The data suggests that there is a significant variation in the number of one-year-olds who did not receive the vaccine against diphtheria, pertussis, and tetanus across these countries.

Figure 3.4: Examples of summaries generated by Flan-T5 XL model after fine-tuning.

Table 3.2: Statistics about the captions of the datasets used in LM pretraining.

Datasets	#Vocab	Avg. Char	Avg. Token	Avg. Sentence
Statista	72,725	450.28	106.68	4.46
OWID	58,212	463.48	105.99	4.47
OECD	24,752	414.97	95.08	4.78
PlotQA	112,394	666.09	149.84	5.20
Data Aug.	162,239	468.41	113.46	4.49
PewResearch	13,449	604.04	133.01	4.66
LineCap	2018	110.82	26.24	1.87
Neural Caption	1338	262.84	53.28	3.58

in image format without a data table[5] and cover about 20% of the corpus. Bar charts make up the majority portion (58.51%), followed by line charts (33.94%) and pie charts (9.39%). About 60.27% of the charts have multiple columns in their data tables, while 39.73% of the charts have only two columns.[6] The corpus also covers a diverse range of topics including technology, economy, politics, health, and society. The linguistic characteristics of the textual elements vary across different datasets, with charts from PlotQA and PewResearch often having longer text elements (e.g., axis labels, legends, titles), while augmented data and Beagle datasets contain shorter text (Table 3.1, right). In Table 3.2, we further provide linguistic statistics for the summaries of the datasets used in the summary generation task at pretraining.

[5]The ExcelChart400K dataset only provides bounding box annotations of chart elements and we used this dataset for data value estimation task during pretraining.

[6]Since we do not have access to the chart types of Pew dataset, we manually tagged random 200 samples from each of these datasets to estimate the chart type distribution.

3.5 Summary

In this chapter, we provide detailed explanations of our process for creating the pretraining corpus. We categorized various types of sources, made modifications to them, and conducted a extensive analysis of the final version of the corpus. Given that charts are generally rare data types on the web, we believe that our augmentation of real-world data and the generation of well-written summaries using the knowledge distillation approach have greatly contributed to the development of a comprehensive and adaptable corpus for the task of chart understanding.

4 Chart Pretraining Methodoloy

In this chapter, we will outline our approach to pretraining. Initially, we will provide a precise definition of the pretraining problem, followed by a discussion of the model architecture and pretraining objectives employed to train our model. Subsequently, we will introduce the downstream tasks and delve into the comprehensive evaluation methods, including both automated and extensive human assessments, that we utilized to thoroughly evaluate the performance of our model on various benchmarks.

4.1 Problem Definition

In this section, we will first discuss the pretraining problem and then explain how it will be applied to Chart Question Answering (CQA) and other related downstream tasks such as OpenCQA, Chart-to-Table, and Chart-to-Text. The pretraining corpus consists of samples denoted as $S = \{C, P, L\}$, where C is a chart image, P is a prompt or question about C, and L is the label that should follow the prompt. In the pretraining process the model predicts L based on C and P as inputs. Figure 4.1 shows the paradigm of the

pretraining. This mechanism allows our model to function in an end-to-end manner and be more appropriate for real-world scenarios where most of the time chart image is the only available information about the chart.

After the pretraining phase, the pretrained model will be utilized for Chart Question Answering and other major downstream tasks in chart domain. There are two different scenarios for the CQA problem [76]. In the first scenario, the data table behind the chart image is available, while in the second scenario, the image is the only information available for the chart and the data table cannot be accessed. In this study, we will concentrate on the second scenario, which will provide an end-to-end solution for the CQA problem. By adapting the CQA problem to our pretraining setup, a question can play the role of P and the answer to the question can be represented by L. In this case, other chart downstreams tasks can be seen simply in the same format where the answer would be L and the prompt would be P in the problem definition concept. In open-ended question answering problem the question is P and L is the explanatory answer of the question. Gold summary and data table would be L for Chart-to-Text and Chart-to-Table tasks as well.

4.2 Model Architecture

We propose UniChart, a unified pretrained model for chart comprehension and reasoning. UniChart consists of two main modules: a chart image encoder and a text decoder as

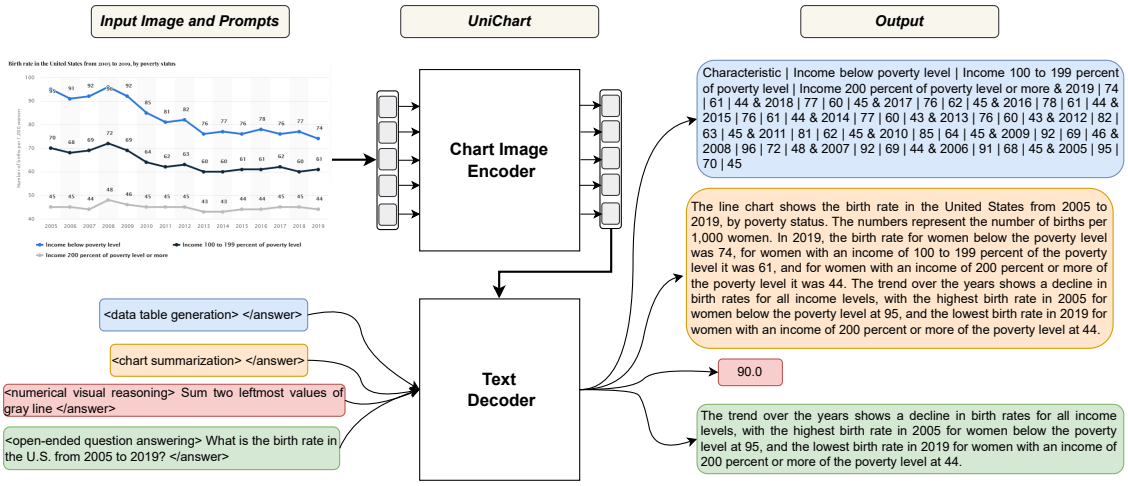


Figure 4.1: Our UniChart model with different pretraining objectives. The model consists of two main modules: Chart Image Encoder, and Text Decoder. Four different pretraining objectives are specified in different colors; *data table generation*, *chart summarization*, *numerical and visual reasoning*, and *open-ended question answering*.

shown in Figure 4.1.

• **Chart Image Encoder** In order to effectively encode a chart image, an encoder needs to identify and interpret three different types of chart components: (1) textual elements (axis labels and legends), (2) visual elements (e.g., bars, lines), and (3) the layout that arranges textual and visual elements within a chart. Since this has a similarity with document

Table 4.1: Training details for pretraining and finetuning experiments.

Experiment	# Epochs/Steps	Learning Rate	Batch Size	GPUs	Saving Mechanism
Pretraining					
First-stage/ablations	300K steps	1e-4	160	4xA100 82GB	each 50K steps
Second-stage	100K steps	1e-4	80	4xA100 82GB	each 50K steps
Finetuning (main 960x960 model)					
ChartQA	20 epochs	5e-5	24	4xV100 32GB	each 1 epoch
Chart-to-text Pew	200 epochs	5e-5	48	4xA100 40GB	each 5 epochs
Chart-to-text Statista	100 epochs	5e-5	48	4xA100 40GB	each 5 epochs
OpenCQA	200 epochs	5e-5	24	4xV100 32GB	each 5 epochs

image (e.g., receipts) understanding, our chart image encoder builds upon the encoder of one of the recent state-of-the-art document image understanding models, Donut [48].

Donut offers an OCR-free architecture. The model is pretrained using an OCR-pseudo task, where it sequentially generates the encoded text in a document image, following the order from the top-left corner to the bottom-right corner of the image. As a result, we did not have to run an external OCR module like CRAFT [4] and Parseq [6], which improved time and memory efficiency throughout our training pipeline. Donut employs Swin Transformer [67] architecture as the image encoder. To encode the chart image features, the images are split into non-overlapping patches, which are then processed using shifted window-based multi-headed self-attention and MLP layers to produce the image embeddings.

• **Text Decoder** Similar to Donut [48], we use the BART [53] decoder for generating the output. The textual (task-specific) prompts are fed to the decoder and the decoder has to generate the output by conditioning on the prompted context (see Figure 4.1).

Further details about the setup and hyper-parameters for the pretraining experiments are provided in Table 4.1

4.3 Pretraining Objectives

Our pretraining objectives include low-level tasks that are more focused on retrieving the underlying data from the chart images and high-level tasks that align closely with the

Table 4.2: Number of examples for each task in pretraining.

Dataset	Data Table Generation	Numerical & Visual Reasoning	Open-ended Question Answering	Chart Summarization
Pew	0	0	5,295	5,295
Statista, OECD, OWID	144,147	679,420	126,009	126,009
PlotQA	155,082	2,414,359	157,070	157,070
LineCap	0	0	2,821	2,821
Neural Caption	0	0	100	306
Beagle	3,972	51	0	0
ChartInfo	1,796	21,949	0	0
Data Aug.	189,792	2,218,468	189,802	189,802
ExcelChart	106,897	0	0	0
Total	**601,686**	**5,334,247**	**481,097**	**481,303**

downstream tasks.

• **Data Table Generation** A chart creates a visual representation of a data table by mapping each data attribute (e.g., 'country', 'population') to corresponding visual attributes (e.g., `x-positions`, `height`) of graphical marks (e.g, bars). An effective chart comprehension and reasoning model should be able to deconstruct the structured underlying data table by recovering such mappings. To this end, we propose the data table generation task in which we ask the model to generate the flattened data table given a chart image.

A vast amount of charts available online are stored in bitmap image format with no access to the underlying data table. It is important to learn how to recover data values when the chart data is not available. To this end, we also designed the task of data value estimation, in which the model is asked to generate the scale of the graphical marks (e.g., bars, line points) as a percentage of the chart plot area. We obtain these scales by dividing the bars or line points heights (bounding boxes) by the height of the chart plot area and

rounding the result to two decimal places. At the final stage, we use charts for which both data tables and object bounding boxes are available as well as charts for which at least the bounding box annotations are available, e.g., ExcelCharts from [70].

• **Numerical & Visual Reasoning** Many downstream applications over charts may involve numerical and visual reasoning with the chart elements such as chart QA and summarization. For example, the model may need to apply a series of mathematical and logical operations such as addition, subtraction and comparisons to answer a question. To inject such reasoning skills into the model, we design template-based numerical reasoning tasks where the model is trained to execute/perform the most common mathematical operations over the chart data values. We manually analyzed the existing task datasets (e.g., ChartQA) to find the most common operations (*e.g.,*sum, average, difference, etc.) and constructed 90 templates that we utilize to generate synthetic question and answer pairs. All the templates are provided in Table 4.7.

• **Open-ended Question Answering** It is very common for users to ask open-ended questions over charts [46]. Such questions often ask for answers that require high-level reasoning and explanations. To improve the capability of the model in answering open-ended questions, we follow previous work [99] to generate synthetic open-ended QA pairs. Specifically, a T5 model [88] pretrained on SQuAD [89] is employed to generate an open-ended question for each summary. The sentence containing the answer in the summary then serves as the answer to its generated question.

• **Chart Summarization** Image captioning is a fundamental problem in AI in which the machines need to summarize the main content of the image in the textual form. This task has been studied extensively [110, 33, 37, 57]. We follow previous work [110, 113] to pretrain our model on this task to further enhance the model's capability in generating textual descriptions from the chart image. As discussed in Section 3.3, we used mostly the summaries generated from GPT models provided by OpenAI either directly or through a knowledge distillation step.

Detailed statistics of each pretraining objective among all the datasets in pretraining corpus is shown in Table 4.2.

4.4 Downstream Tasks

In addition to zero-shot evaluation, we also adapt UniChart by finetuning it on a downstream task. We consider four downstream tasks: *(1) Factoid Chart Question Answering:* we use ChartQA [76], which is a benchmark consisting of factoid question-answer pairs for charts with a particular focus on visual and logical reasoning questions; *(2) Complex Chart Question Answering:* we consider OpenCQA [46], another QA benchmark in which answers are explanatory descriptions; *(3) Chart Summarization:* we use Chart-to-Text [98], a large-scale benchmark for chart summarization in case the user asks for a summary about the chart; *(4) Chart-to-Table:* Another downstream task in which the model is asked to generate underlying data table of a chart. Significance of accessing the gold data table in

chart question answering has been studied previously [76]. Consequently, we believe the model's ability to comprehend the underlying data table in the chart would significantly enhance its capacity to address chart-related questions as well. We use ChartQA for both finetuning and evaluation. Moreover, we evaluate the pretrained model in a zero-shot setup on the WebCharts dataset [14], a collection of 300 charts obtained from the web. Our experimental setups and hyperparameters for each downstream task are provided in Table 4.1.

4.5 Evaluation

4.5.1 Baselines

We compare our model against five baselines:

- *T5* [88], a unified seq2seq Transformer model that achieved state-of-the-art (SoTA) results on various text-to-text tasks, including question answering and summarization.
- *VL-T5* [13], a T5-based model that unifies Vision-Language (VL) tasks as text generation conditioned on multimodal inputs and achieved SoTA results on OpenCQA [46].
- *VisionTapas* [76], an extension of TaPas [34], a BERT-based SoTA table encoder,

adapted for QA over charts utilizing visual features extracted from visual transformer architecture.

- *Pix2Struct* [52], a pretrained image-to-text model for visual language understanding, which can be finetuned on tasks involving visually-situated language, and achieved SoTA results on document understanding tasks.

- *MatCha* [63], an adapted version of Pix2Struct for charts that is further pretrained on math reasoning and chart data extraction tasks, achieving SoTA results on Chart-to-Text [98] and ChartQA [76].

4.5.2 Evaluation Metrics

To evaluate our approach, we follow previous works [52, 98, 76, 46, 63] and utilize Relaxed Accuracy for ChartQA and BLEU [85] for text-generation tasks (Chart-to-Text and OpenCQA). However, the BLEU score has limitations as it primarily focuses on n-gram matching between the generated and reference texts, overlooking important factors such as semantic similarity, informativeness, and factual correctness [28]. Therefore, we conduct a human evaluation and ChatGPT-driven study to assess and compare these crucial aspects in the outputs of different models (Section 4.5.5). Finally, we use Relative Number Set Similarity (RNSS) [76] and Relative Mapping Similarity (RMS) [62] metrics to evaluate the Chart-to-Table task.

Table 4.3: Evaluation results on four public benchmarks: ChartQA, Chart-to-Text, OpenCQA, and Chart-to-Table. All the results are calculated after finetuning UniChart pretrained checkpoint except for WebCharts (zero-shot).

	ChartQA (RA)			OpenCQA ($BLEU$)	Chart-to-Text ($BLEU$)		Chart-to-Table ($RNSS$ | RMS_{F1})	
Model	aug.	human	avg.	OpenCQA	Pew	Statista	ChartQA	WebCharts
VisionTaPas [76]	61.44	29.60	45.52	-	-	-	-	-
T5 [76]	56.96	25.12	41.04	9.28	10.49	35.29	-	-
VL-T5 [76]	56.88	26.24	41.56	14.73	-	-	-	-
Pix2Struct [52]	81.6	30.5	56.0	-	10.3	38.0	-	-
MatCha [63]	**90.2**	38.2	64.2	-	12.2	**39.4**	85.21 | 83.49	44.37 | 17.94
UniChart	88.56	**43.92**	**66.24**	**14.88**	**12.48**	38.21	**94.01 | 91.10**	**60.73 | 43.21**

4.5.3 Main Results

4.5.3.1 Chart Question Answering

Table 4.3 shows that UniChart outperforms previous state-of-the-art models, MatCha and Pix2Struct, on the ChartQA benchmark [76]. First, there is a noticeable difference between the performance of UniChart and the baselines reported by Masry et al. [76] (T5, VL-T5, and VisionTapas), which underscores the importance of our relevant pretraining objectives (numerical/visual reasoning and data table extraction) in enhancing the model capability in factoid question answering task. Furthermore, UniChart outperforms more recent SoTA models which employed pretraining approach, Pix2Struct and MatCha. The performance gap is more prominent on the challenging human-written questions in the ChartQA benchmark, where our model's pretraining objectives tailored to visual and numerical reasoning give it a significant advantage.

In terms of open-ended chart question answering task, UniChart achieves a higher BLUE score compared to the SoTA VL-T5 model on OpenCQA benchmark [46], which demonstrates our model's capability in generating explanatory answers for questions about charts. Overall, these results establish UniChart as the SoTA model for chart question answering domain.

4.5.3.2 Chart-to-Table

Data Extraction from charts is another downstream task which is highly correlated with chart question answering domain [76]. To evaluate our model, we choose two ChartQA, and WebCharts [15] datasets, which contain diverse visual styles and chart types, making the data extraction task challenging for the model. We also take MatCha as the SoTA baseline in chart understanding domain. As shown in Table 4.3, UniChart surpasses MatCha's performance on both datasets based on both $RNSS$ and RMS_{F1} metrics. Better performance of UniChart on WebCharts demonstrates its generalizability across diverse visual styles, even in a zero-shot setup on unseen charts.

4.5.3.3 Chart-to-Text

Finally, we evaluate UniChart on generating comprehensive summaries about charts based on Chart-to-Text benchmark [98]. Table 4.3 shows UniChart outperforms T5, best baseline reported by Shankar et al. [98] and Pix2Struct in both Pew and Statista splits.

Table 4.4: UniChart ablations on ChartQA benchmark.

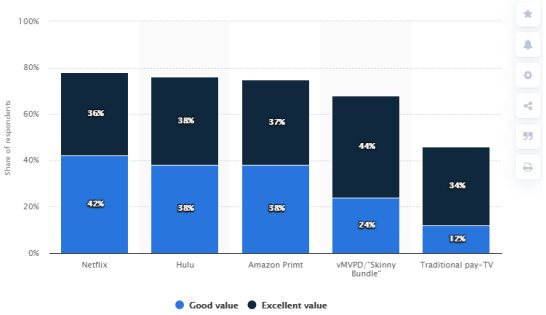
	ChartQA		
Model	aug.	human	avg.
UniChart (512x512)	85.84	43.60	64.72
No Chart Summarization	84.96	42.72	63.84
No Open-ended Question Answering	85.52	42.96	64.24
No Numerical & Visual Reasoning	84.08	35.44	59.76
No Data Table Generation	83.84	42.24	63.04

Compared to SoTA MatCha, UniChart demonstrates a higher performance based on BLUE score on Pew split, although it shows slightly lower performance on Chart-to-Text Statista. We will do a more extensive evaluation approach for Chart-to-Text task in future sections by utilizing both human and model-based methods.

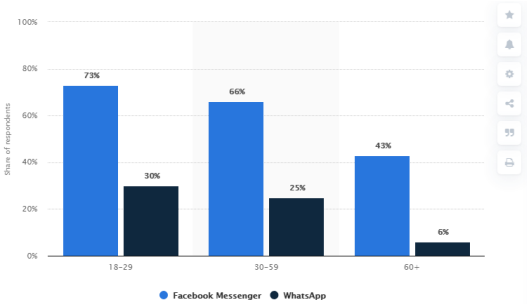
Figure 4.2 shows the performance of UniChart on four different downstream tasks on which we evaluate our pretraining approach.

4.5.4 Ablation Studies

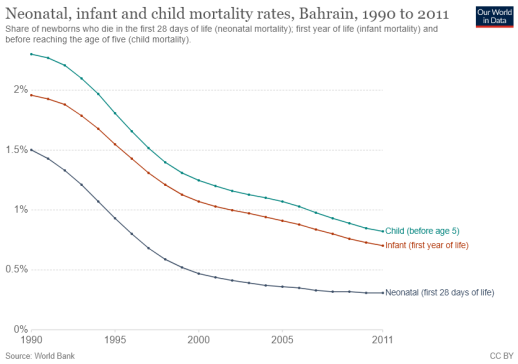
To further assess the impact of our different pretraining objectives on our model's performance, we conducted a series of ablation studies. Due to computational limitations, we focused on pretraining the model only on the lower image size (512x512) and compared it against the corresponding main model (512x512). From Table 4.4, we observe that removing the Chart Summarization or Open-ended Question Answering objectives resulted in a minor decrease in performance on ChartQA. We attributed this to the abun-



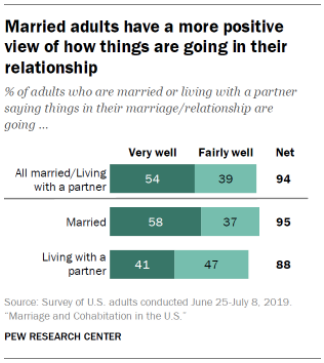
Q1: What is the lowest value in blue bar?
UniChart: 12



Q2: Extract underlying data table.
UniChart: Characteristic | Facebook Messenger | WhatsApp & 18-29 | 73% | 30% & 30-59 | 66% | 25% & 60+ | 43% | 6%



Q3: Summarize the chart.
UniChart ZeroShot: The line chart shows the neonatal, infant and child mortality rates in Bahrain from 1990 to 2011. The child (before age 5) mortality rate decreased from 2.34 in 1990 to 0.86 in 2011, while the infant (first year of life) mortality rate decreased from 2.0 in 1990 to 0.74 in 2011. The neonatal (first 28 days of life) mortality rate decreased from 1.54 in 1990 to 0.34 in 2011. Overall, the trend shows a decrease in the neonatal, infant and child mortality rates in Bahrain from 1990 to 2011.



Q4: How do married adults conceive their relationship satisfaction in comparison to adults living with a partner?
UniChart: Among adults who are married, 54% say they are very well married or living with a partner, compared with 41% of those who are living with a partner.

Figure 4.2: Some examples from different downstream tasks where UniChart demonstrates appropriate performance.

Table 4.5: Average Informativeness scores from Human and ChatGPT-based evaluation.

Summary	Human	ChatGPT	p-value
UniChart ZeroShot	**3.97**	**3.18**	1.70e-10
UniChart Finetuned	2.86	2.37	2.32e-8
MatCha [63]	2.50	2.18	0.0020
Gold [98]	3.19	2.73	2.13e-6

Table 4.6: Human evaluation on summaries for 150 random samples from Chat2text Statista test split.

Criteria	ZeroShot	Finetuned	MatCha	Gold
Factually incorrect sents	13.45%	9.63%	21.97%	**3.59%**
Elemental/encoded sents	19.42%	26.06%	**29.61%**	21.07%
Statistical/relational sents	**57.41%**	33.42%	34.07%	34.70%
Perceptual/cognitive sents	**6.98%**	1.41%	0.31%	5.39%
Contextual/domain-specific sents	1.36%	14.44%	7.32%	**20.56%**

dance of numerical reasoning examples in pretraining. On the other hand, removing the Numerical Reasoning pertaining task led to a substantial decrease in performance on ChartQA, highlighting the importance of this task in imbuing numerical abilities into our model. Pretraining the model without the Data Table Generation objective resulted in a relatively weak performance in the ChartQA benchmark, underscoring the significance of understanding underlying data tables of charts in answering reasoning questions.

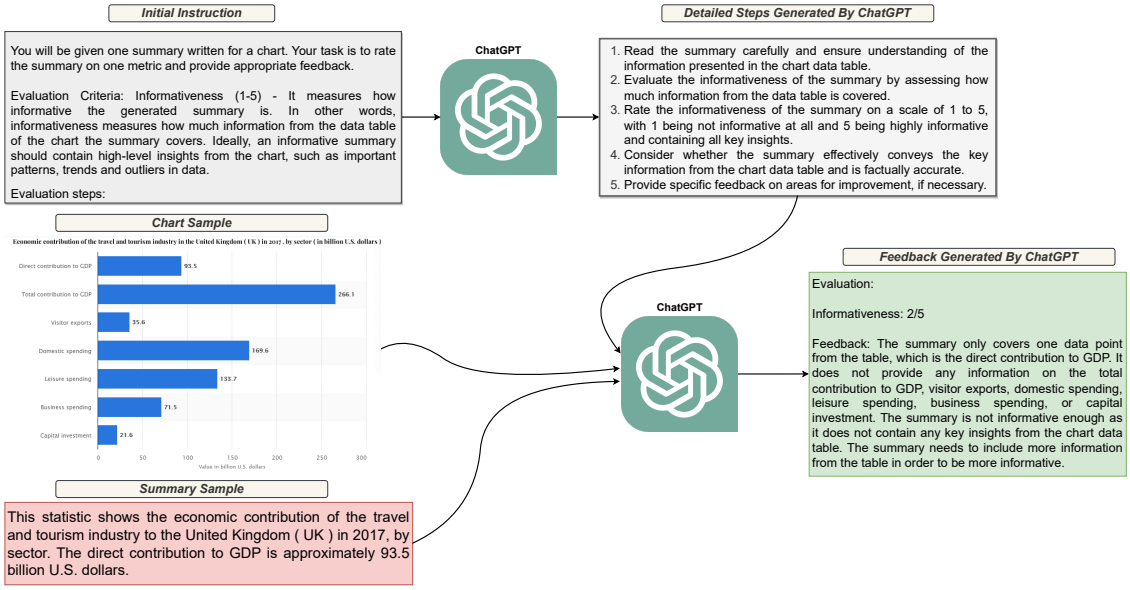
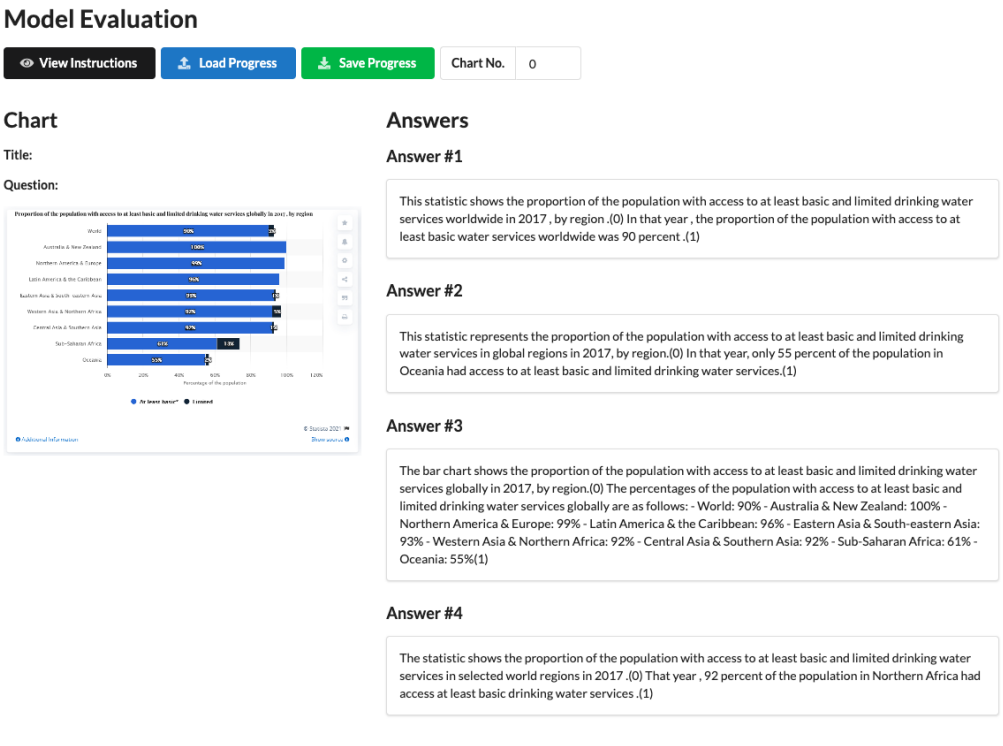


Figure 4.3: The pipeline designed for the ChatGPT Evaluation Experiment. First, we feed the task description followed by our desired criteria into ChatGPT in order to get detailed grading instructions. Then, the chart (underlying data table representation) and a sample summary are appended to the prompt which is fed again into ChatGPT to receive the feedback.

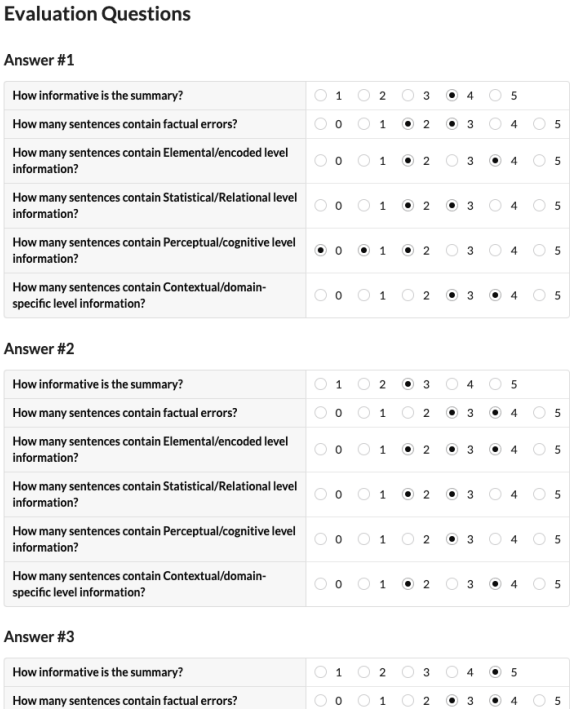
4.5.5 Human and ChatGPT evaluation

As discussed in Section 4.5.2, reference-based metrics like BLEU have major weaknesses in terms of evaluating text-generation tasks have relatively low correlations with human judgments [7, 108, 65], and generated texts with very high such scores can be of a very poor quality [101]. Therefore, we decided to conduct a human evaluation to measure the quality of summaries generated by different models. We focus on three major criteria in the chart summarization task including:

1. *(1) Informativeness* which measures how much information from the chart the summary covers. Ideally, an informative summary should contain high-level insights



(a)



(b)

Figure 4.4: Human evaluation interface. Figure 4.4a shows a chart sample from Chart-to-Text Statista test split with four different versions of UniChart ZeroShot, UniChart fine-tuned, MatCha, and Ground True summaries in a random order. Figure 4.4b shows the questions an annotator should answer regarding each summary.

from the chart, such as important patterns, trends, and outliers in data.

2. *(2) Factual Correctness* which considers how accurate the summary is. A factually correct summary should only contain information (e.g. numbers, events, entities) that is true and/or supported by the chart.

3. *(3) Semantic Levels* defined by Lundgard and Satyanarayan [69] which categorize the content of summaries across four levels: visual encoding (e.g., axis, legends, color), statistical/relational (e.g., min, max, avg.), perceptual/cognitive (e.g., describing overall trends, complex patterns, or outliers), and context/domain specific information.

We randomly picked 150 sample charts from Chart2text Statista test split and asked 3 human annotators to rate four summaries for each chart based on *informativeness* out of 1 to 5. The order of exposure of summaries to the annotator was randomized to avoid any potential bias. Summaries for each chart were rated by one annotator except for the first 100 charts for which we had two annotators to measure the agreement. We computed Krippendorff's alpha [50] to measure inter-annotator agreement and found a moderate level of agreement with an alpha coefficient of 0.54. For factual correctness and semantic level measures, the annotator goes through each sentence of the summary to determine whether the sentence contains any factual error and what levels of semantic content are present for that sentence. We further utilize ChatGPT for evaluating the same 150 samples,

as LLMs have demonstrated their effectiveness as evaluators for text generation tasks [72, 65, 26, 24]. We define the informativeness criteria and rating scheme to ChatGPT and then employ ChatGPT to generate evaluation steps. We then send these evaluation steps along with the data table of the chart and the summary to ChatGPT to obtain ratings. Figure 4.3 shows an overview of the paradigm we use in our ChatGPT-driven evaluation study. Figure 4.4 depicts the interface we used in our human evaluation study. The annotator is required to evaluate each summary by responding to six different questions. The initial question involves providing a score for informativeness, where only one option can be chosen. Subsequently, the annotator must identify the sentence numbers that contain factual errors. The following four questions focus on determining the sentence numbers that fall into specific semantic categories.

Table 4.5 shows the result of human evaluation on chart summarization based on informativeness criteria. We notice that annotators preferred ZeroShot version of our model which generates summaries that are more similar to those generated by GPT, rather than gold summaries. The finetuned version of UniChart was also rated higher compared to SoTA MatCha [63]. The finetuned UniChart model also produces fewer factual errors compared to Matcha and the ZeroShot version (Table 4.6). We observe that the ratings provided by ChatGPT are roughly consistent with the human annotators' scores in terms of informativeness criteria. In terms of different semantic contents, the ZeroShot model tends to contain more sentences with high-level visual patterns and trends. A previous

study finds that such high-level insights lead to more reader takeaways compared to the text describing low-level visual encodings like axes and colors [104]. Overall, the results above suggest that UniChart model's summaries are more informative with high-level insights and factually accurate than the SoTA (MatCha).

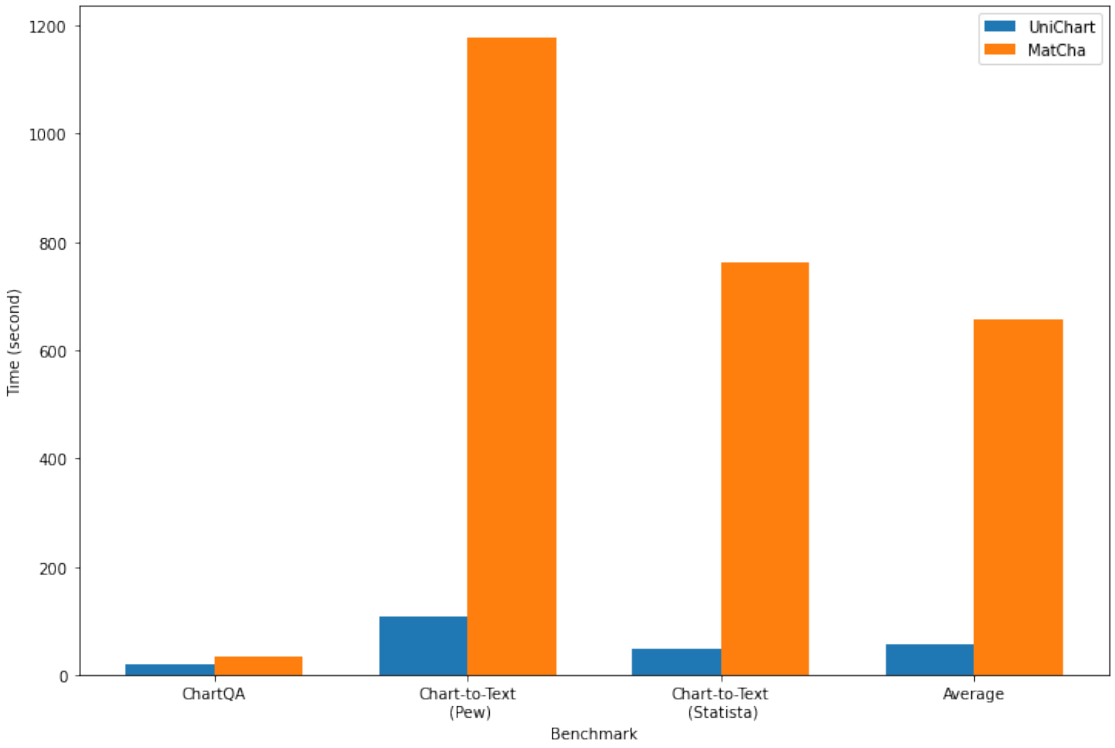


Figure 4.5: Average inference time for 10 random samples from three major benchmarks in chart understanding domain for UniChart and MatCha models

4.5.6 Time and Memory Efficiency

UniChart exhibits significant time efficiency compared to MatCha, as shown in Figure 4.5. To compare the time efficiency, we measure the average inference time of the models on

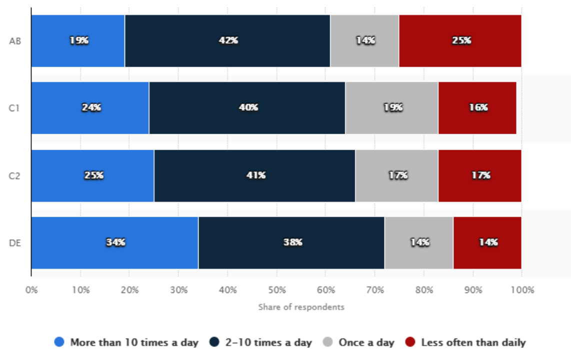
three benchmarks: ChartQA, Chart-to-Text (Pew), and Chart-to-Text (Statista) using 10 random samples from each benchmark. The experiments were conducted on Google's Colab platform with cpu type. The gap in speed is more evident on tasks that require the generation of long output sequences (e.g., Chart-to-Text). This difference in speed can be attributed to MatCha's use of a long input sequence (4K) with a quadratic increase in complexity while UniChart's vision encoder relies on sliding windows with a local attention mechanism that scales linearly with the input image size. Moreover, UniChart boasts a smaller parameter count (201M) compared to MatCha (282M), further contributing to its efficiency. As a result, UniChart is highly suitable for real-world applications that prioritize fast inference speeds.

4.6 Error Analysis and Challenges

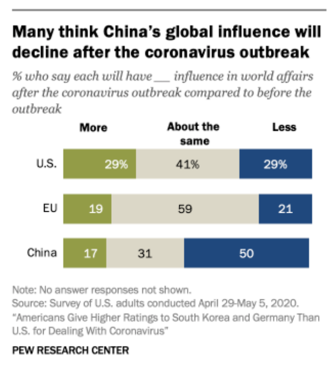
We conducted a manual analysis of our model's outputs to identify key challenges faced by existing models.

• **Densely populated charts:** Our model struggles with extracting insights from chart images that contain numerous data elements densely packed in a limited area. This is evident in Figure Figure 4.6 (Q3) where our model generates a hallucinated summary due to the complexity of the chart. Increasing model parameters and input image resolution could potentially improve performance in these cases.

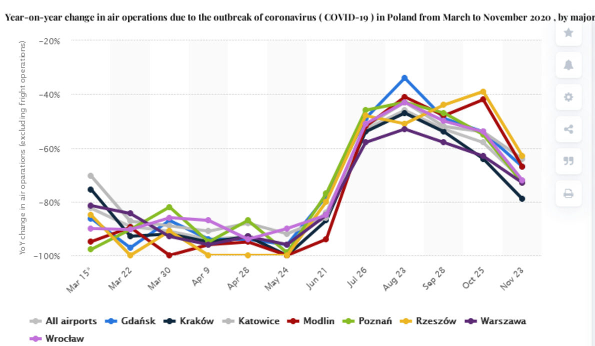
• **Numerical reasoning:** Despite efforts to incorporate mathematical skills, our model



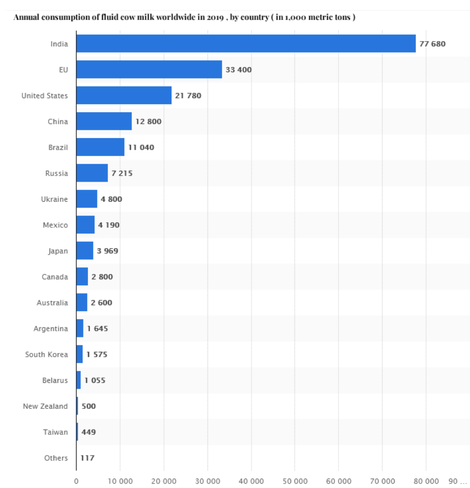
Q1: What is the largest value of red bar?
Answer: 25 ***UniChart***: 25 **MatCha**: 40



Q2: What's the average of all the values in the green bars (round to one decimal)?
Answer: 21.6 ***UniChart***: 0.4 **MatCha**:0.51



Q3: Summarize the chart.
UniChart ZeroShot*:* The line chart shows the year-on-year change in air operations due to the outbreak of COVID-19 in Poland from March to November 2020, broken down by major occupations. The occupations are measured in air operations (excluding Fight operations), in wholes (excluding Fight operations), and in wholes (excluding Fight operations).The occupations are measured in air operations (excluding Fight operations), in wholes (excluding Fight operations), and in wholes (excluding Fight operations). The occupations are measured in air operations (excluding Fight operations), in wholes (excluding Fight operations), and in wholes (excluding Fight operations). The occupations are measured in air operations (excluding Fight operations), in wholes (excluding Fight operations), ... revehivehivehivehivehivehivehivehivehivehivehivehivehivehivehiveh.



Q4: Summarize the chart.
UniChart Finetuned: This statistic illustrates the annual consumption of fluid cow milk worldwide in 2019, by country. In that year, India's fluid cow milk amounted to 77.7 million metric tons, making it the world's second largest consumer of milk.

Figure 4.6: Some challenging examples from the ChartQA and Chart-to-text benchmarks.

still encounters difficulties with complex arithmetic calculations (Q2 in Figure 4.6). Addressing this challenge involves decoupling arithmetic calculations and reasoning steps by employing external program executors that perform the calculations using the equations generated by our model [25].

• **Factual correctness in generated summaries:** Factual correctness still poses a challenge for autoregressive language models [60, 81, 119]. Although our finetuned UniChart model produced fewer factual errors compared to MatCha (see Table 4.6), it still generates some incorrect statements (see Q4 in Figure 4.6). This issue can be attributed to factual errors in the pretraining captions generated by ChatGPT.

4.7 Discussion

In this chapter we presented our pretraining methodology followed by pretraining objectives, each focusing on distinct skills in chart understanding domain. Then we introduced downstream tasks, baselines and evaluation metrics we used to measure UniChart performance against the baselines. We applied an extensive evaluation procedure using both automatic metrics and also human and model-based approaches to compare UniChart and SoTA performance. We also compared UniChart with SoTA based on time and memory complexity. Finally, we briefly explained current challenges and errors that we observed after evaluation procedure.

Table 4.7: Numerical & Visual reasoning templates.

Template-based Questions
1) First bar from the top/left in the second group from the top/left
2) First bar from the bottom/right in the second group from the bottom/right
3) Second bar from the bottom/right in the first group from the bottom/right
4) Second bar from the right/bottom in the first group from the left/top
5) Topmost/Leftmost bar
6) Bottommost/Rightmost bar
7) Second bar from the top/left
8) Second bar from the right/bottom
9) Leftmost topmost bar
10) Leftmost bottommost bar
11) Rightmost topmost bar
12) Rightmost bottommost bar
13) Leftmost <color> data
14) Rightmost <color> data
15) Second from the left <color> data
16) Second from the right <color> data
17) Which legend represented by <color>?
18) What is the color of <legend>?
19) Which one is greater, <x1> or <x2>?
20) Divide the sum of largest and lowest values by <n>
21) When did line <legend - label> peak?
22) What is the difference between maximum and minimum of <legend - label>?
23) Sum pie segments above <value>
24) What is the sum of top three values?
25) What is the median/mode of <legend - label>?
26) What is the negative peak of <legend - label>?
27) What is the largest/smallest value of <legend - label>?
28) Which two x-axis labels of <legend - label> sums up to <value>?
29) What is the sum of the second highest and second lowest value of <legend - label>?
30) Which x-axis label is second highest for <legend - label>?
31) What is the sum of two middle values of <legend - label>?
32) Which two x-axis labels of <legend - label> have a difference of <value> ?
33) What is the average of <legend - label> from <x - label - 1> to <x - label - 2>?
34) What is the average of the highest and lowest value of <legend - label - 1>?
35) What is the sum of the average of <legend - label - 1> and average of <legend - label - 2>?
36) What is the sum/difference of the maximum of <legend - label - 1> and minimum of <legend - label - 2>?
37) Which x-axis label has the maximum/minimum difference between <legend - label - 1> and minimum of <legend - label - 2>?
38) Which x-axis label witnessed the smallest value of <legend - label>?
39) Which label contains largest/smallest values across all labels?
40) Sum up the medians of all the data series in this chart
41) What is the average of all values above <value>?
42) What is the sum of the largest and smallest difference between <legend - label - 1> and <legend - label - 2>?
43) What is the maximum/minimum difference between <legend - label - 1> and <legend - label - 2>?
44) What is the ratio of the largest to the smallest pie segment?
45) What is the ratio of the two largest/smallest segments?
46) What is the difference between the leftmost and rightmost bars?
47) What is the sum of the bars in the second group from the left?
48) What is the sum of the bars in the first group from the right?
49) What is the ratio between the two leftmost bars?
50) What is the difference between the rightmost <color - 1> bar and leftmost <color - 2> bar?
51) What is the average of <color> bars values?
52) How many <color>bars are larger than <N> ?
53) What is the average of the bars in the second group from the right?
54) How many bars in the leftmost group have a value over <N>?
55) What does the <color> represent?
56) What is the median value of the <color> bars/line?
57) What is the average of the <color - 1> sum and <color - 2> sum?
58) What is the average of the <color - 1> median and <color - 2> median?
59) What is the least difference between the <color - 1> and <color - 2> bars/line?
60) What is the ratio between the leftmost and rightmost bar in the first group from the left?
61) What is the maximum value in the <color> bars/line?
62) What is the minimum value in the <color> bars/line?
63) What is the sum of <color> bars/line?
64) What is the difference between the maximum values of the two leftmost bar groups?
65) Sum of the first <color - 1> and last <color - 2> bars/line points
66) Difference between the two lowest <color> bars
67) Add largest and smallest <color> line/bar values and divide by 2
68) What is the value of <color> line/bars in <x - axis - label>?
69) Sum/Average of <color - 1> and <color - 2> values in <x - axis - label>?
70) Sum of highest points in <color - 1> and <color - 2> lines/bars
71) Which color has the highest/smallest values?
72) How many values are equal in <color - 1> line/bar?
73) Sum two rightmost values of <color> graph
74) Product of two smallest values in the graph
75) Sum of lowest and median values of <color> graph/bars
76) When did <color> line reached the peak?
77) What is the average of the rightmost three points of <color> line?
78) How many <color> data points are above <value>?
79) What's the ratio of the largest and the third/second-largest <color> bar?
80) Is the sum of lowest value of <color - 1> and <color - 2> bar greater than largest value of <color - 3> bar?
81) Is the median value of <color - 1> bars greater than the median value of <color - 2> bars?
82) Is the median of all the <color - 1> bars greater than the largest value of <color - 2> bar?
83) What's the product of <color> bars in India and Japan?
84) Is the sum of the two middle bars greater than the sum of top and bottom bars?
85) What's the ratio of the <x - axis - label - 1> <color - 1> bar and the <x - axis - 2> <color - 2> bar?
86) Is the total of all <color - 1> bars greater than the total of all <color - 2> bars?
87) Take the sum of the two smallest <color - 1> bars and smallest <color - 2> bars, deduct the smaller value from the larger value, what's the result?
88) What is the sum/average of two smallest/largest <color> bars?
89) What is the ratio of <color - 1> and <color - 2> segments?
90) What segment is represented by <color>?

5 Conclusions and Future Work

Here we will outline the final conclusion of this research and also provide some directions for further future work on this domain.

5.1 Conclusion

We present UniChart, a general purpose pretrained model designed for chart question answering and a broad range of other chart-related tasks. Our model incorporates chart-specific pretraining tasks and is trained on a large and diverse collection of charts and corresponding summaries collected opportunistically using LLMs. We conducted both human and ChatGPT evaluations to show the superiority of our method. While our model sets the state-of-the-art record on four different downstream tasks and showed improved time and memory efficiency, the evaluation also reveals opportunities for improvement. We believe that our model and pretraining data will be valuable resources for future research and encourage further exploration in this relatively new area.

Overall, our major contributions in this work are *(i)* Curating a large pretraining corpus containing diverse visual and topic styles, which makes our model generalizable on unseen

charts; *(ii)* Introducing an pretraining approach which utilizes both low-level (data table extraction) and high-level (numerical/visual reasoning, summarization, and open-ended question answering) objectives, which causes our model adaptable on other downstream tasks in chart understanding domain; *(iii)* Applying different evaluation methods including automatic metrics, model-based and human evaluation to demonstrate our model SoTA performance on different benchmarks.

5.2 Future Work

While UniChart exhibits state-of-the-art performance on several benchmarks, it suffers from several limitations. Despite the remarkable abilities on the ChartQA dataset, the model still struggles to answer questions that involve compositional mathematical operations. Moreover, we have noticed that the model may hallucinate and produce factually incorrect statements on the text generation tasks such as Chart-to-Text and OpenCQA.

Despite the generalizability of our model on unseen chart image styles (WebCharts), there's still a noticeable drop in performance compared to the performance on the tasks on which the model is finetuned (e.g., ChartQA). Hence, there's still a need for better generalizable chart models for the diverse charts on the Web. One direction is to enlarge our pretraining datasets by crawling millions of chart images from the Web. Since most charts on the Web do not provide high-quality captions or the underlying data table, self-supervised pretraining objectives are needed to benefit from these charts.

Second, improving current deficiencies in the performance of SoTA models like complicated numerical reasoning, factual correctness in generated summaries, and hallucination in over-populated charts can be a focus in future. Current SoTA demonstrates very well performance on simple numerical reasoning or less populated charts, but making the performance robust against all kinds of input charts and questions requires extra effort.

Another domain that can be explored is employing current large language models to generate more human-like and coherent summaries for complicated charts like PewResearch. Currently, SoTA models are capable of generating well-written summaries for charts with more regular style like Statista, but when it comes to the charts with unique visual style multiple data series, the model is inclined to hallucinate and even generate low-quality text. One possible solution for this issue is to utilize large language models enjoys great ability in generating high-quality text in order to generate coherent and human-like summaries for complicated charts, just like what we observed from ChatGPT performance in summary generation for PewResearch charts during curating pretraining corpus Section 3.3.

Lastly, there remains an insufficient exploration of a crucial aspect within the chart domain, namely the provision of a model that is well-suited for a wide range of potential downstream tasks. UniChart employs four objectives that are confined to a textual/visual format as input and a textual format as output. However, there are other scenarios to consider, such as processing multiple views (where multiple charts are provided as input)

or visual storytelling (where the generation of chart visualizations is the desired output). These scenarios necessitate novel architectures capable of effectively processing different modalities both in input and output.

Due to the limited computing resources, we did not investigate the effect hyperparameter tuning might have on the performance on the different downstream tasks. Also, although we have noticed the convergence of UniChart at the end of the second stage pretraining, we can not confirm whether further pretraining may improve the performance of our model.

Bibliography

[1] Competition on harvesting raw tables from infographics. https://chartinfo.github.io/index_2022.html. 2022.

[2] Web data commons, extracting structured data from the common crawl. https://webdatacommons.org/.

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

[4] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. *CoRR*, abs/1904.01941, 2019. URL http://arxiv.org/abs/1904.01941.

[5] Leilani Battle, Peitong Duan, Zachery Miranda, Dana Mukusheva, Remco Chang, and Michael Stonebraker. Beagle: Automated extraction and interpretation of visualizations from the web. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2018.

[6] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *European Conference on Computer Vision*, pages 178–196, Cham, 10 2022. Springer Nature Switzerland. doi: 10.1007/978-3-031-19815-1_11. URL https://doi.org/10.1007/978-3-031-19815-1_11.

[7] Anja Belz and Ehud Reiter. Comparing automatic and human evaluation of NLG systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 313–320, Trento, Italy, April 2006. Association for Computational Linguistics. URL https://aclanthology.org/E06-1040.

[8] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-driven documents. 2011. URL http://vis.stanford.edu/papers/d3.

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

[10] Jian Chen, Meng Ling, Rui Li, Petra Isenberg, Tobias Isenberg, Michael Sedlmair, Torsten Moller, Robert S Laramee, Han-Wei Shen, Katharina Wunsche, et al. Vis30k: A collection of figures and tables from ieee visualization conference publications. *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[11] Xi Chen, Wei Zeng, Yanna Lin, Hayder Mahdi Al-Maneea, Jonathan C. Roberts, and Remco Chang. Composition and configuration patterns in multiple-view visualizations. *CoRR*, abs/2007.15407, 2020. URL `https://arxiv.org/abs/2007.15407`.

[12] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020.

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

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

[15] Yejin Choi, Eric Breck, and Claire Cardie. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 431–439, Sydney, Australia, 2006. ACL. ISBN 1-932432-73-6.

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

[17] Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.565. URL `https://aclanthology.org/2021.acl-long.565`.

[18] Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. Summarizing information graphics textually. *Computational Linguistics*, 38(3):527–574, 2012. doi: 10.1162/COLI_a_00091. URL https://www.aclweb.org/anthology/J12-3004.

[19] Dazhen Deng, Yihong Wu, Xinhuan Shu, Jiang Wu, Mengye Xu, Siwei Fu, Weiwei Cui, and Yingcai Wu. Visimages: a corpus of visualizations in the images of visualization publications. *arXiv preprint arXiv:2007.04584*, 2020.

[20] BOSHENG DING, Chengwei Qin, Linlin Liu, Yew Ken Chia, Lidong Bing, Boyang Li, and Shafiq Joty. Is gpt-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, ACL'23, Toronto, Canada, 2023. ACL.

[21] 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. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[22] Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. A survey of vision-language pre-trained models. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5436–5443. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/762. URL https://doi.org/10.24963/ijcai.2022/762. Survey Track.

[23] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246.

[24] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire, 2023.

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

[26] Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. Human-like summarization evaluation with chatgpt, 2023.

[27] Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology*, UIST 2015, pages 489–500, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3779-3.

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

[29] Daniel Haehn, James Tompkin, and Hanspeter Pfister. Evaluating 'graphical perception'with cnns. *IEEE transactions on visualization and computer graphics*, 25(1):641–650, 2018.

[30] Jonathan Harper and Maneesh Agrawala. Converting basic d3 charts into reusable style templates. *IEEE transactions on visualization and computer graphics*, 24(3): 1274–1286, 2017.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[32] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[33] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. *Advances in Neural Information Processing Systems*, 32, 2019.

[34] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. TAPAS: weakly supervised table parsing via pre-training. *CoRR*, abs/2004.02349, 2020. URL `https://arxiv.org/abs/2004.02349`.

[35] Enamul Hoque and Maneesh Agrawala. Searching the visual style and structure of d3 visualizations. In *IEEE Transactions on Visualization and Computer Graphics (Proc IEEE InfoVis 2019)*, volume 26, pages 1236–1245. IEEE, 2019.

[36] Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. Applying pragmatics principles for interaction with visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):309–318, 2017.

[37] Xiaowei Hu, Xi Yin, Kevin Lin, Lei Zhang, Jianfeng Gao, Lijuan Wang, and Zicheng Liu. Vivo: Visual vocabulary pre-training for novel object captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1575–1583, 2021.

[38] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4083–4091, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548112. URL https://doi.org/10.1145/3503161.3548112.

[39] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

[40] Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. Seeing out of the box: End-to-end pre-training for vision-language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12976–12985, 2021.

[41] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[42] Daekyoung Jung, Wonjae Kim, Hyunjoo Song, Jeong in Hwang, Bongshin Lee, Bohyoung Kim, and Jinwook Seo. Chartsense: Interactive data extraction from chart images. ACM, 5 2017. URL https://www.microsoft.com/en-us/research/publication/chartsense-interactive-data-extraction-chart-images/.

[43] Kushal Kafle, Scott Cohen, Brian L. Price, and Christopher Kanan. DVQA: understanding data visualizations via question answering. *CoRR*, abs/1801.08163, 2018. URL http://arxiv.org/abs/1801.08163.

[44] Kushal Kafle, Robik Shrestha, Scott Cohen, Brian Price, and Christopher Kanan. Answering questions about data visualizations using efficient bimodal fusion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1498–1507, 2020.

[45] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*, 2017.

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

[47] Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. Answering questions about charts and generating visual explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[48] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer, 2022.

[49] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR, 18–24 Jul 2021. URL `http://proceedings.mlr.press/v139/kim21k.html`.

[50] Klaus Krippendorff. Computing krippendorff's alpha-reliability, 2011.

[51] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1051. URL `https://aclanthology.org/D19-1051`.

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

[53] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and

comprehension. *CoRR*, abs/1910.13461, 2019. URL http://arxiv.org/abs/1910.13461.

[54] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11336–11344, 2020.

[55] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021.

[56] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=OJLaKwiXSbx.

[57] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

[58] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019.

[59] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. What does BERT with vision look at? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5265–5275, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.469. URL https://aclanthology.org/2020.acl-main.469.

[60] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229.

[61] Can Liu, Yun Han, Ruike Jiang, and Xiaoru Yuan. Advisor: Automatic visualization answer for natural-language question on tabular data. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*, pages 11–20, 2021. doi: 10.1109/PacificVis52677.2021.00010.

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

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

[64] Xiaoyi Liu, Diego Klabjan, and Patrick N. Bless. Data extraction from charts via single deep neural network. *ArXiv*, abs/1906.11906, 2019.

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

[66] Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Jason Wu, Caiming Xiong, and Dragomir Radev. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, ACL'23, Toronto, Canada, 2023. ACL. URL `https://arxiv.org/abs/2212.07981`.

[67] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[68] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

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

[70] Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. Chartocr: Data extraction from charts images via a deep hybrid framework. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1916–1924, 2021.

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

[72] Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. Chatgpt as a factual inconsistency evaluator for text summarization, 2023.

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

[74] Anita Mahinpei, Zona Kostic, and Chris Tanner. Linecap: Line charts for data visualization captioning models. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pages 35–39. IEEE, 2022.

[75] Ahmed Masry and Enamul Hoque. 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, 2021.

[76] Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 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, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177.

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

[78] Tamara Munzner. *Visualization analysis and design*. CRC press, 2014.

[79] Arpit Narechania, Arjun Srinivasan, and John Stasko. Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):369–379, 2020.

[80] Jason Obeid and Enamul Hoque. 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, 2020. URL https://www.aclweb.org/anthology/2020.inlg-1.20.

[81] OpenAI. Chatgpt: Optimizing language models for dialogue, 2022. URL https://openai.com/blog/chatgpt/.

[82] 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. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

[83] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1142. URL https://www.aclweb.org/anthology/P15-1142.

[84] Jorge Poco, Angela Mayhua, and Jeffrey Heer. Extracting and retargeting color mappings from bitmap images of visualizations. *IEEE transactions on visualization and computer graphics*, 24(1):637–646, 2017.

[85] Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

[86] Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. Is chatgpt a general-purpose natural language processing task solver?, 2023.

[87] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[88] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[89] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016. URL http://arxiv.org/abs/1606.05250.

[90] Revanth Reddy, Rahul Ramesh, Ameet Deshpande, and Mitesh M. Khapra. Figurenet : A deep learning model for question-answering on scientific plots. *Proceedings of the International Joint Conference on Neural Networks*, 2019-July, 2019. doi: 10.1109/IJCNN.2019.8851830.

[91] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[92] Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427*, 2017.

[93] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics*, 23(1):341–350, 2016.

[94] M. Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and J. Heer. Revision: automated classification, analysis and redesign of chart images. *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011.

[95] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models, 2019.

[96] Vidya Setlur, Sarah E. Battersby, Melanie Tory, Rich Gossweiler, and Angel X. Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST 2016, pages 365–377, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4189-9.

[97] Vidya Setlur, Enamul Hoque, Dae Hyun Kim, and Angel X. Chang. Sneak pique: Exploring autocompletion as a data discovery scaffold for supporting visual analysis. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 966–978, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375146. doi: 10.1145/3379337.3415813. URL https://doi.org/10.1145/3379337.3415813.

[98] Kantharaj Shankar, Leong Rixie Tiffany Ko, Lin Xiang, Masry Ahmed, Thakkar Megh, Hoque Enamul, and Joty Shafiq. 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*, 2022.

[99] Peng Shi, Patrick Ng, Feng Nan, Henghui Zhu, Jun Wang, Jiarong Jiang, Alexander Hanbo Li, Rishav Chakravarti, Donald Weidner, Bing Xiang, and Zhiguo Wang. Generation-focused table-based intermediate pre-training for free-form question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11312–11320, Jun. 2022. doi: 10.1609/aaai.v36i10.21382. URL https://ojs.aaai.org/index.php/AAAI/article/view/21382.

[100] Hrituraj Singh and Sumit Shekhar. STL-CQA: Structure-based transformers with localization and encoding for chart question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3275–3284, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.264. URL https://www.aclweb.org/anthology/2020.emnlp-main.264.

[101] Aaron Smith, Christian Hardmeier, and Joerg Tiedemann. 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, 2016. URL https://aclanthology.org/W16-3414.

[102] Andrea Spreafico and Giuseppe Carenini. Neural data-driven captioning of time-series line charts. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375351. doi: 10.1145/3399715.3399829. URL https://doi.org/10.1145/3399715.3399829.

[103] Arjun Srinivasan and John Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE transactions on visualization and computer graphics*, 24(1):511–521, 2018.

[104] Chase Stokes, Vidya Setlur, Bridget Cogley, Arvind Satyanarayan, and Marti A Hearst. Striking a balance: Reader takeaways and preferences when integrating text and charts. *IEEE Transactions on Visualization and Computer Graphics*, 29 (1):1233–1243, 2022.

[105] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International*

Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=SygXPaEYvH.

[106] Chen Sun, Austin Myers, Carl Vondrick, Kevin P. Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7463–7472, 2019.

[107] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[108] Liling Tan, Jon Dehdari, and Josef van Genabith. An awkward disparity between BLEU / RIBES scores and human judgements in machine translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 74–81, Kyoto, Japan, October 2015. Workshop on Asian Translation. URL https://aclanthology.org/W15-5009.

[109] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. *arXiv preprint arXiv:2212.02623*, 2022.

[110] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[111] Jiapeng Wang, Lianwen Jin, and Kai Ding. Lilt: A simple yet effective language-independent layout transformer for structured document understanding. *arXiv preprint arXiv:2202.13669*, 2022.

[112] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

[113] Qiaolin Xia, Haoyang Huang, Nan Duan, Dongdong Zhang, Lei Ji, Zhifang Sui, Edward Cui, Taroon Bharti, and Ming Zhou. Xgpt: Cross-modal generative pre-training for image captioning. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 786–797. Springer, 2021.

[114] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020.

[115] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1192–1200, 2020. doi: 10.1145/3394486.3403172.

[116] Bowen Yu and Cláudio T. Silva. Flowsense: A natural language interface for visual data exploration within a dataflow system. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1–11, 2020. doi: 10.1109/TVCG.2019.2934668.

[117] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1425. URL `https://aclanthology.org/D18-1425`.

[118] Lin-Ping Yuan, Wei Zeng, Siwei Fu, Zhiliang Zeng, Haotian Li, Chi-Wing Fu, and Huamin Qu. Deep colormap extraction from visualizations. *arXiv preprint arXiv:2103.00741*, 2021.

[119] Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. *arXiv preprint arXiv:2305.03268*, 2023.

[120] Jialong Zou, Guoli Wu, Taofeng Xue, and Qingfeng Wu. An affinity-driven relation network for figure question answering. *Proceedings - IEEE International Conference on Multimedia and Expo*, 2020-July, 2020. ISSN 1945788X. doi: 10.1109/ICME46284.2020.9102911.

A Ethical Considerations

During the dataset collection process, we made sure to comply with the terms and conditions of the different websites we used to crawl our data. Statista[7] provide a permissive license to use their publicly available data for scientific purposes. Pew Research Centre [8] also provide a permissive license to use their data with the condition that we attribute it to the Centre. OECD[9] allows the users to download and publish their data as long as they give appropriate credit to the OECD website. For OWID[10], all their data are provided under the Creative Commons BY license which gives the permission for downloading and publication. Web Data Commons[11], which we used in the data augmentation process, allows the usage of their data under the conditions of the Apache License Software which gives the right to download and publish. Finally, all the remaining datasets (PlotQA, Beagle, ChartInfo, ExcelChart400K, LineCap, and Neural Captions)

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

[8] https://www.pewresearch.org/about/terms-and-conditions/

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

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

[11] https://webdatacommons.org/

are publicly available datasets which were released in earlier scientific publications.

Due to the generative nature of our models, they may be abused for misinforming the public by generating factually incorrect responses. Moreover, we can not guarantee that our models may not produce texts that may contain hate speech or harmful content.