Doc2Chart: Intent-Driven Zero-Shot Chart Generation from Documents

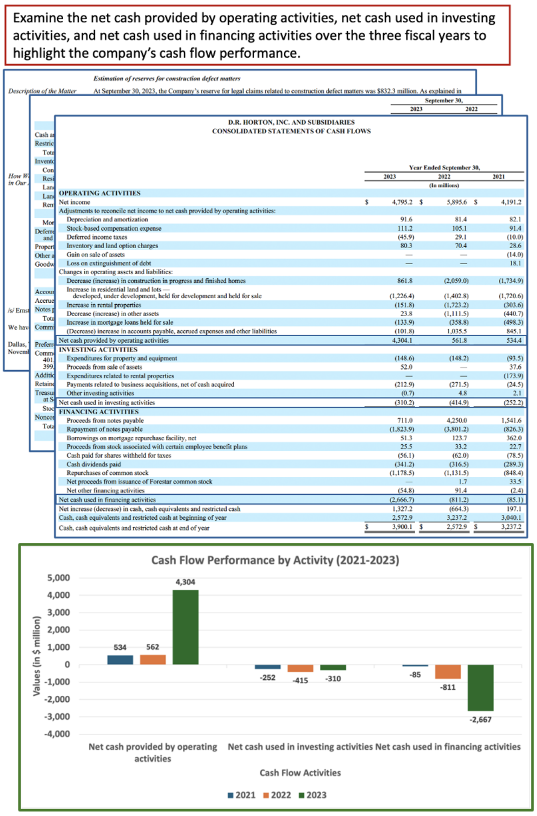
Akriti Jain, Pritika Ramu, Aparna Garimella, Apoorv Saxena
Adobe Research, India
{akritij, pramu, garimell, apoorvs}@adobe.com

Abstract

Large Language Models (LLMs) have demonstrated strong capabilities in transforming text descriptions or tables to data visualizations via instruction-tuning methods. However, it is not straightforward to apply these methods directly for a more real-world use case of visualizing data from long documents based on user-given intents, as opposed to the user pre-selecting the relevant content manually. We introduce the task of *intent-based chart generation* from documents: given a user-specified intent and document(s), the goal is to generate a chart adhering to the intent and grounded on the document(s) in a zero-shot setting. We propose an unsupervised, two-staged framework in which an LLM first extracts relevant information from the document(s) by decomposing the intent and iteratively validates and refines this data. Next, a heuristic-guided module selects an appropriate chart type before final code generation. To assess the data accuracy of the generated charts, we propose an attribution-based metric that uses a structured textual representation of charts, instead of relying on visual decoding metrics that often fail to capture the chart data effectively. To validate our approach, we curate a dataset comprising of 1,242 <intent, document, charts> tuples from two domains, finance and scientific, in contrast to the existing datasets that are largely limited to parallel text descriptions/ tables and their corresponding charts. We compare our approach with baselines using single-shot chart generation using LLMs and query-based retrieval methods; our method outperforms by upto 9 points and 17 points in terms of chart data accuracy and chart type respectively over the best baselines.

1 Introduction

Statistical charts offer an intuitive way to grasp insights from lengthy documents, such as financial reports and scientific articles, which are often dense with data, frequently presented in large tables. Automatic chart generation tailored to specific user goals can significantly enhance various document consumption and creation workflows. These include providing illustrative answers to specific questions, generating multi-modal summaries for queries, and creating visually rich presentations.



Examine the net cash provided by operating activities, net cash used in investing activities, and net cash used in financing activities over the three fiscal years to highlight the company's cash flow performance.

Estimation of reserves for construction defect matters

Description of the Matter At September 30, 2023, the Company's reserve for legal claims related to construction defect matters was $832.3 million. As explained in

	September 30,	
	2023	2022

D.R. HORTON, INC. AND SUBSIDIARIES
CONSOLIDATED STATEMENTS OF CASH FLOWS

	Year Ended September 30,		
	2023	2022	2021
		(In millions)	
OPERATING ACTIVITIES			
Net income	$ 4,795.2	$ 5,895.6	$ 4,191.2
Adjustments to reconcile net income to net cash provided by operating activities:			
Depreciation and amortization	91.6	81.4	82.1
Stock-based compensation expense	111.2	105.1	91.4
Deferred income taxes	(45.9)	29.1	(10.0)
Inventory and land option charges	80.3	70.4	28.6
Gain on sale of assets	—	—	(14.0)
Loss on extinguishment of debt	—	—	18.1
Changes in operating assets and liabilities:			
Decrease (increase) in construction in progress and finished homes	861.8	(2,059.0)	(1,734.9)
Increase in residential land and lots — developed, under development, held for development and held for sale	(1,226.4)	(1,402.8)	(1,720.6)
Increase in rental properties	(151.8)	(1,723.2)	(303.6)
Decrease (increase) in other assets	23.8	(1,111.5)	(440.7)
Increase in mortgage loans held for sale	(133.9)	(358.8)	(498.3)
(Decrease) increase in accounts payable, accrued expenses and other liabilities	(101.8)	1,035.5	845.1
Net cash provided by operating activities	4,304.1	561.8	534.4
INVESTING ACTIVITIES			
Expenditures for property and equipment	(148.6)	(148.2)	(93.5)
Proceeds from sale of assets	52.0	—	37.6
Expenditures related to rental properties	—	—	(173.9)
Payments related to business acquisitions, net of cash acquired	(212.9)	(271.5)	(24.5)
Other investing activities	(0.7)	4.8	2.1
Net cash used in investing activities	(310.2)	(414.9)	(252.2)
FINANCING ACTIVITIES			
Proceeds from notes payable	711.0	4,250.0	1,541.6
Repayment of notes payable	(1,823.9)	(3,801.2)	(826.3)
Borrowings on mortgage repurchase facility, net	51.3	123.7	362.0
Proceeds from stock associated with certain employee benefit plans	25.5	33.2	22.7
Cash paid for shares withheld for taxes	(56.1)	(62.0)	(78.5)
Cash dividends paid	(341.2)	(316.5)	(289.3)
Repurchases of common stock	(1,178.5)	(1,131.5)	(848.4)
Net proceeds from issuance of Forestar common stock	—	1.7	33.5
Net other financing activities	(54.8)	91.4	(2.4)
Net cash used in financing activities	(2,666.7)	(811.2)	(85.1)
Net increase (decrease) in cash, cash equivalents and restricted cash	1,327.2	(664.3)	197.1
Cash, cash equivalents and restricted cash at beginning of year	2,572.9	3,237.2	3,040.1
Cash, cash equivalents and restricted cash at end of year	$ 3,900.1	$ 2,572.9	$ 3,237.2

Cash Flow Performance by Activity (2021-2023)
Values (in $ million)
5,000
4,000
3,000
2,000
1,000
0
-1,000
-2,000
-3,000
-4,000
534
562
4,304
-252
-415
-310
-85
-811
-2,667
Net cash provided by operating activities
Net cash used in investing activities
Net cash used in financing activities
Cash Flow Activities
2021
2022
2023

Figure 1: Example of intent-based chart generation from documents. Input: intent, document(s); output: chart.

Advancements in LLMs (Brown et al., 2020; Touvron et al., 2023) have enabled recent efforts in generating high-quality statistical charts from user-given text descriptions or tables (Wang et al., 2023; Han et al., 2023; Maddigan and Susnjak, 2023; Zhang et al., 2024; Tian et al., 2024; Zadeh et al., 2024). However, these existing approaches to automatic chart generation typically assume that users will manually provide tables or textual descriptions

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 34948–34963
November 4-9, 2025 ©2025 Association for Computational Linguistics

to be transformed into charts. Consequently, these methods cannot directly generate charts from raw documents based on specific user intents or queries. Further, many previous methods rely on instruction-tuning techniques that necessitate a substantial volume of labeled data (in the order of 10s of 1000s) for effective training. Obtaining such large corpora of charts paired with their corresponding intents is a tedious and time-consuming process, particularly for long documents (spanning >5 pages), which are the primary focus of our work.

While LLMs demonstrate superior generation quality based on natural language prompts (Shao et al., 2024; Ramu et al., 2024), applying them directly for chart generation using the given intent as the NL prompt often leads to charts with (a) hallucinations or poor intent adherence in terms of the data values, and (b) chart types that are not always appropriate for the given data. To address these challenges, we propose an unsupervised, multi-stage framework. It first employs an LLM to decompose the user's intent to guide an iterative data extraction and refinement process from the document. Subsequently, it selects an appropriate chart type using predefined visualization heuristics before generating the final chart. Furthermore, evaluating the fidelity of such generated charts is non-trivial. Existing evaluation strategies either rely on human judgments, which are costly and not easily scalable (Tian et al., 2024; Zhang et al., 2024), or employ LLM- and VLM-based evaluators (Koh et al., 2024; Ford et al., 2024), which often struggle to interpret complex charts accurately (Islam et al., 2024). We propose an attribution-based chart evaluation metric that uses a structured text representation of generated charts, and uses attention-based heat map obtained from a forward LLM pass, to detect the spans of data values that are not captured in the reference charts. As no existing datasets support this specific task, we finally curate a new dataset of long documents, intents, and corresponding charts from financial reports and academic papers from *ACL conferences.

This paper makes four main contributions: **(1)** We introduce the novel task of generating charts from documents based on specific intents, to mimic real-world use cases for chart generation. **(2)** We propose a training-free, multi-stage framework that leverages LLMs to first execute an intent-guided iterative data extraction and refinement process from documents, and subsequently to select an appropriate chart type through a heuristic-guided deliberation, before constructing the final chart. **(3)** We propose a chart evaluation metric to assess the faithfulness of the generated chart data when compared to either a reference chart/ table (reference-based setting) or source document itself (reference-free setting). **(4)** We curate a dataset consisting of 1,242 <intent, document, chart> tuples from financial reports and scientific articles for this task evaluation.

We present experimental results comparing our method with baselines including naïve prompting of LLMs and query-based retrieval methods. Our approach demonstrates notable improvements in key aspects such as appropriate chart type selection and chart data accuracy. To the best of our knowledge, this is the first work to address document-to-chart generation based on intent, and can assist in furthering research in chart generation.

2 Related Work

Our work on intent-driven chart generation from documents intersects with several efforts, including approaches to generate and evaluate charts from more structured inputs, and techniques for understanding user intent and extracting information from documents.

2.1 Chart Generation from Pre-Processed Inputs

A significant body of research addresses chart generation from structured or semi-structured data. This includes text-to-chart synthesis, where systems generate charts from concise textual descriptions (Rashid et al., 2021; Zadeh et al., 2024) or employ reasoning over structured inputs like tables (Tian et al., 2024; Wang et al., 2023). Even more advanced systems that process pre-selected tabular data (Han et al., 2023) or multimodal inputs (Xia et al., 2025) primarily operate on data that is already directly present. While these methods demonstrate strong capabilities in translating well-defined inputs into charts, they typically assume the data is already extracted, curated, and directly relevant to the desired visualization. Our work differs significantly by tackling the upstream challenge of identifying the required information from voluminous and often complex documents based on a high-level user intent, before any chart can be synthesized.

2.2 Intent Understanding and Information Extraction from Documents

Understanding user intent to extract relevant information from long, complex documents is a key challenge in information retrieval and content generation. Several research directions have explored aspects of this problem. In conversational search and retrieval-augmented generation (RAG), systems such as RQ-RAG (Chan et al., 2024) aim to refine ambiguous queries through user-driven clarification dialogues (e.g., “Do you mean revenue over time or per region?”). However, these approaches typically operate on short, keyword-based inputs (e.g., “revenue growth 2023”) and assume iterative user interaction. Unlike systems that depend on external user feedback, we incorporate internal validation and refinement steps to improve data completeness and correctness before chart generation, with the goal of producing a good-quality chart in a single pass, without requiring further clarification or user intervention. Similarly, in the table retrieval literature, recent work highlights the difficulty of locating specific table content in documents when queries are abstract or underspecified (Chen et al., 2025). In such cases, simple keyword-matching or embedding retrieval approaches often underperform, especially when the query requires interpreting context across paragraphs and tables. While query decomposition techniques are widely used in multi-hop question answering and multi-table reasoning (Chen et al., 2025), they are typically designed to support inference across facts rather than to extract structured components for data visualization. These methods lack fine-grained alignment to chart-specific needs, such as identifying axis labels, categories, or values. Unlike broader intent-driven generation tasks such as story writing or document drafting (Shao et al., 2024; Ramu et al., 2024), our problem requires the precise extraction of numerical data, grounded in the document and aligned with the user’s high-level intent.

2.3 Chart Evaluation

Early chart evaluation works focus on comparing generated chart specifications or code to ground-truth references using measures like ROUGE, BLEU, and CodeBLEU (Tian et al., 2024; Zadeh et al., 2024). While these offer a measure of similarity or error, they often provide a superficial assessment, potentially overlooking semantic correctness, as data regeneration metrics alone have been found to offer a limited view of performance (Ford et al., 2024). Further, human evaluation, involving user studies or expert reviews (Tian et al., 2024; Zhang et al., 2024), provides deeper insights into chart correctness and usability but solely relying on human evaluation can be costly and not easily scalable. More recently, automated approaches using LLMs or VLMs as evaluators (Ford et al., 2024) have been proposed. However, they often exhibit factual errors and hallucinations, particularly struggling with data extraction from charts. (Islam et al., 2024). To address these limitations while evaluating the factual accuracy in the generated charts from documents, we advocate for chart attribution: tracing the generated chart data values back to the source tables in the document.

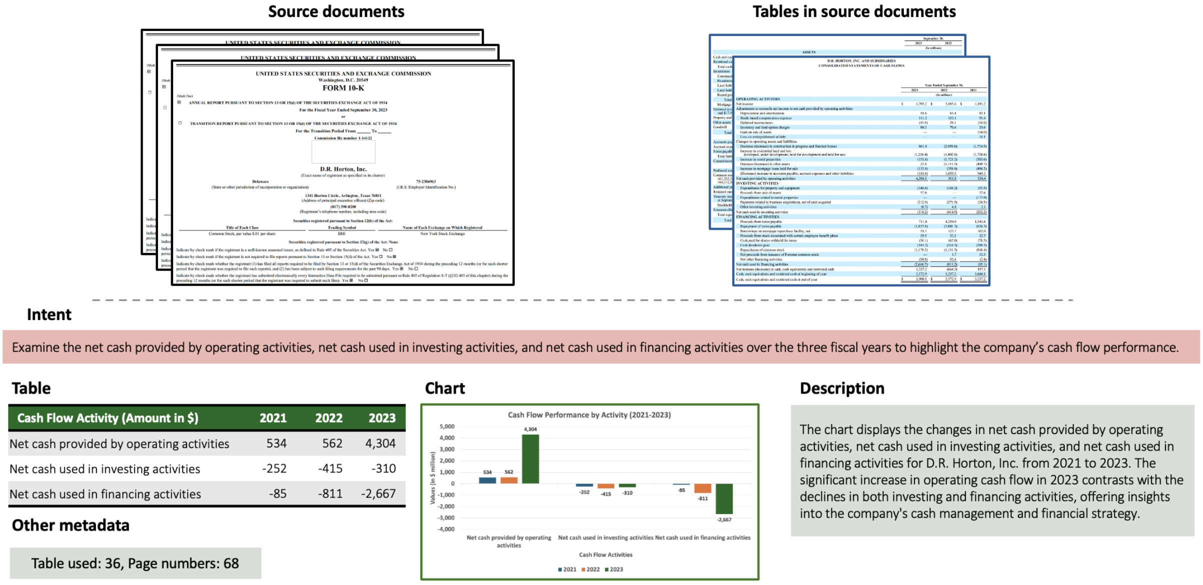
3 Task Setup & Dataset

The task of intent-driven chart generation from documents is characterized by two primary challenges: first, the precise extraction of relevant data; and second, the selection of an appropriate chart type to effectively communicate this information in line with the user’s intent. Consequently, for a given document D and a user-specified intent I, the objective is to produce a statistical chart C that is both grounded in D and directly addresses I, as depicted in Figure 1.

3.1 Dataset Curation Overview

Existing chart generation datasets primarily focus on text-to-chart (Rashid et al., 2021; Zadeh et al., 2024) and table-to-chart (Han et al., 2023) tasks, where the input text or tables contain exactly the information represented in the charts—nothing more. In contrast, our task involves generating charts from potentially long and complex documents that include significantly more content than what appears in the final visualization. To support this setting, we curate a new dataset by providing annotators with source documents (e.g., financial reports, scientific articles) and instructing them to: (a) formulate relevant user intents, and (b) create corresponding charts grounded in these documents. However, scaling such annotation is time-consuming and costly, as it requires deep ingestion of lengthy documents (often spanning tens of pages) to derive meaningful intents. This challenge motivates our adoption of a zero-shot methodology, which requires no task-specific fine-tuning.

For the source documents, we consider two do-



Examine the net cash provided by operating activities, net cash used in investing activities, and net cash used in financing activities over the three fiscal years to highlight the company's cash flow performance.

Cash Flow Activity (Amount in $)	2021	2022	2023
Net cash provided by operating activities	534	562	4,304
Net cash used in investing activities	-252	-415	-310
Net cash used in financing activities	-85	-811	-2,667

Table used: 36, Page numbers: 68

The chart displays the changes in net cash provided by operating activities, net cash used in investing activities, and net cash used in financing activities for D.R. Horton, Inc. from 2021 to 2023. The significant increase in operating cash flow in 2023 contrasts with the declines in both investing and financing activities, offering insights into the company's cash management and financial strategy.

Figure 2: Sample annotations for a given SEC document and its constituent data tables. An intent and a corresponding chart are provided, along with the data table supporting the chart and a description summarizing the chart key takeaways. Other metadata such as page number and tables used from the source document are also provided. We use the page numbers to truncate the documents to include ±5 pages while using the source documents with the chart generation methods, to avoid context length issues.

	SEC	ACADEMIC
# docs	73	106
Avg. # pages	103	11
Avg. # tables	24	6
# intents/ doc	10	5
Avg. table size	10 x 10	6 x 6

Table 1: Source dataset statistics.

Dataset	Intent	Input type	# Fig.	Desc.	Code
ChartLlama	✗	Table	11K	✓	✓
ChartX	✗	Table	6K	✓	✓
Text2Chart31-v2	✗	Table	28.2K	✓	✓
Ours	✓	**Document**	**2.2K**	✓	✓

Table 2: Comparative analysis with various chart generation datasets: ChartLlama (Han et al., 2023), ChartX (Xia et al., 2025), Text2Chart31-v2 (Zadeh et al., 2024).

mains, namely finance and scientific. We use the U.S. Securities and Exchange Commission (SEC) 10-K filings[1] that are publicly available on the EDGAR website, and academic papers from *ACL conferences from the SciDuet dataset (Sun et al., 2021) respectively. We scrape the HTML documents for 1,000 SEC 10-K filings, and consider all the 1,088 papers from SciDuet in the PDF form. We parse these HTML and PDF documents[2] to obtain the tables in them separately in spreadsheets. From among the SEC filings, we first filter documents that contain table(s) with size >7x7, and then pick the top 100 ones that have the maximum number of tables in them. As most of the numeric content that can be visualized is present in tables in these documents, we use those that densely contain them. Since academic papers do not always contain very large tables, we take the top 120 documents that have the maximum number of tables in them (and relax the size constraints). Since these documents contain multiple tables, and each table can convey several insights, we obtain multiple intents from each of them, to reduce the number of documents that are to be ingested and the overall cognitive load. Table 1 provides the source document details for the two domains. Figure 2 provides an overview of the data curation process.

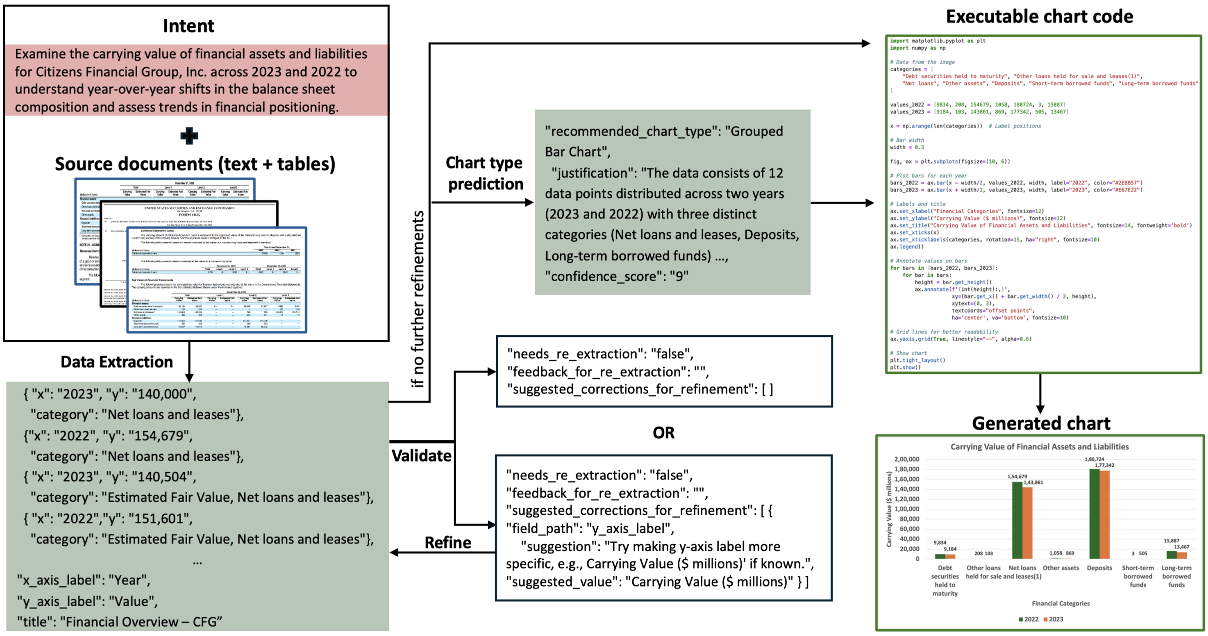
3.2 Data Annotation Setup

To obtain intent-chart annotations, we recruited three annotators from a freelancing platform,[3] who were compensated at $15/hour. All were proficient in content creation with similar demographics (nationality, graduate education, age 20-30). After pilot studies (5 documents each), they were instructed to provide: (a) 5 creative intents; (b) corresponding chart images and tables with appropriate data

[1] https://www.sec.gov/edgar.shtml

[2] BeautifulSoup, https://developer.adobe.com/document-services/docs/overview/pdf-extract-api/

[3] https://www.upwork.com


Intent
Examine the carrying value of financial assets and liabilities for Citizens Financial Group, Inc. across 2023 and 2022 to understand year-over-year shifts in the balance sheet composition and assess trends in financial positioning.
✚
Source documents (text + tables)
Data Extraction
if no further refinements
Chart type prediction
Validate
Refine
OR
Executable chart code
Generated chart
Carrying Value of Financial Assets and Liabilities
Carrying Value ($ millions)
2,00,000
1,80,000
1,60,000
1,40,000
1,20,000
1,00,000
80,000
60,000
40,000
20,000
0
9,834
9,184
208 103
1,54,679
1,43,861
1,058 869
1,80,724
1,77,342
3 505
15,887
13,467
Debt securities held to maturity
Other loans held for sale and leases(1)
Net loans
Other assets
Deposits
Short-term borrowed funds
Long-term borrowed funds
Financial Categories
■ 2022 ■ 2023

{ "x": "2023", "y": "140,000",
"category": "Net loans and leases"},
{"x": "2022", "y": "154,679",
"category": "Net loans and leases"},
{ "x": "2023", "y": "140,504",
"category": "Estimated Fair Value, Net loans and leases"},
{ "x": "2022","y": "151,601",
"category": "Estimated Fair Value, Net loans and leases"},
...
"x_axis_label": "Year",
"y_axis_label": "Value",
"title": "Financial Overview – CFG"

"recommended_chart_type": "Grouped Bar Chart",
"justification": "The data consists of 12 data points distributed across two years (2023 and 2022) with three distinct categories (Net loans and leases, Deposits, Long-term borrowed funds) ...,
"confidence_score": "9"

"needs_re_extraction": "false",
"feedback_for_re_extraction": "",
"suggested_corrections_for_refinement": [ ]

"needs_re_extraction": "false",
"feedback_for_re_extraction": "",
"suggested_corrections_for_refinement": [{
"field_path": "y_axis_label",
"suggestion": "Try making y-axis label more specific, e.g., Carrying Value ($ millions)' if known.",
"suggested_value": "Carrying Value ($ millions)" }]

```
import matplotlib.pyplot as plt
import numpy as np

# Data from the image
categories = [
    "Debt securities held to maturity", "Other loans held for sale and leases(1)",
    "Net loans", "Other assets", "Deposits", "Short-term borrowed funds", "Long-term borrowed funds"
]

values_2022 = [9834, 208, 154679, 1058, 180724, 3, 15887]
values_2023 = [9184, 103, 143861, 869, 177342, 505, 13467]

x = np.arange(len(categories))  # Label positions

# Bar width
width = 0.3

fig, ax = plt.subplots(figsize=(10, 6))

# Plot bars for each year
bars_2022 = ax.bar(x - width/2, values_2022, width, label="2022", color="#2E8B57")
bars_2023 = ax.bar(x + width/2, values_2023, width, label="2023", color="#E67E22")

# Labels and title
ax.set_xlabel("Financial Categories", fontsize=12)
ax.set_ylabel("Carrying Value ($ millions)", fontsize=12)
ax.set_title("Carrying Value of Financial Assets and Liabilities", fontsize=14, fontweight='bold')
ax.set_xticks(x)
ax.set_xticklabels(categories, rotation=15, ha="right", fontsize=10)
ax.legend()

# Annotate values on bars
for bars in [bars_2022, bars_2023]:
    for bar in bars:
        height = bar.get_height()
        ax.annotate(f'{int(height):,}',
                    xy=(bar.get_x() + bar.get_width() / 2, height),
                    xytext=(0, 3),
                    textcoords="offset points",
                    ha='center', va='bottom', fontsize=10)

# Grid lines for better readability
ax.yaxis.grid(True, linestyle="--", alpha=0.6)

# Show chart
plt.tight_layout()
plt.show()
```

Figure 3: The proposed pipeline, illustrating: tterative data extraction, where misalignment with intent or data incompleteness facilitates a re-extraction. A refinement stage, where minor issues in otherwise suitable data are corrected. Chart type prediction based on the finalized data, followed by code generation

and chart types; (c) supporting source content (e.g., tables, page numbers); and (d) text descriptions of the charts. To account for subjectivity, annotators could list up to three chart type variants per intent, ordered by preference. For large tables, they were encouraged to combine data from multiple tables or use subsets, critically considering insightful visualizations. The complexity of this task is reflected in the resulting annotations; while over 95% of charts sourced data from tables (due to the dense numerical information required for statistical charts), their creation was often non-trivial. For instance, approximately 15% of charts required composing data from multiple tables. Even for the 80% of charts that used a single table, annotators had to perform complex selections and subsetting of the data to create insightful visualizations supporting the intent. A smaller portion (5%) required integrating data from both textual and tabular content. Following 2-3 feedback rounds focused on ensuring intents were neither too generic nor specific, the main task involved generating 5 intent-chart pairs for academic articles and 10 for SEC filings, initially yielding 1,275 data points. The authors, as expert reviewers, then filtered samples with ambiguous intents or inaccurate charts, resulting in 1,242 <intent, document, chart> tuples. Table 2 compares our dataset with three existing chart generation datasets. Including plausible chart type variants for some samples increased the total chart count to 2.2K charts.

4 DOC2CHART: Methodology

Generating charts directly from documents using user's intent as a simple prompt for a Large Language Model (LLM) often yields suboptimal results. Such charts can suffer from poor intent adherence, include hallucinated or irrelevant data, or omit crucial information, especially when data spans multiple segments or requires subsetting from large tables in long documents. To address these issues, we propose DOC2CHART, an unsupervised, multi-stage framework (illustrated in Figure 3). Our pipeline systematically processes the document and intent to produce an accurate and appropriate chart.

Iterative Data Extraction and Refinement. The first stage involves decomposing the user intent and identifying relevant content from the document. To ensure the highest fidelity of the extracted data when dealing with the complexities of long-form content, this initial extraction undergoes a critical validation and refinement phase. This step systematically verifies the accuracy, completeness, and relevance of the information, applying corrections or guiding re-extraction as needed. The outcome of this validation dictates the next step. If critical issues such as significant data omissions or fundamental misinterpretations of the intent are identified, the validation module generates specific feedback. This feedback then guides a subsequent extraction attempt, repeating the initial extraction step. If no corrections are deemed necessary, the validated data moves forward directly. This iterative cycle of extraction, validation, and conditional re-extraction or refinement ensures a high degree of alignment between the extracted data, the source document, and the user's intent.

Chart Type Prediction. After verifying and refining the data, the next step is to determine the most suitable chart type. This choice can be non-trivial, as multiple chart types may fit a given dataset and intent. We adaopt a heuristic-guided approach where an LLM analyzes the structure of the data—such as the types of values and the number of data points or categories, in conjunction with the user's intent and recommends a chart type accordingly. For example, line charts are typically preferred for time-series data to highlight trends, while bar charts may be used when the data contains only a few points. Simple categorical comparisons suit standard bar charts, whereas grouped or stacked bar charts are better for subcategory comparisons. Pie charts can be effective for part-to-whole relationships, but only when the number of segments is small enough to remain clear. These also help the model avoid common visualization pitfalls, such as cluttered visuals or misleading representations. Rather than relying on rigid rules, the LLM combines data characteristics with these heuristics to recommend a chart type, along with a justification and a confidence score for its choice.

Code Generation. Finally, we generate an executable chart code with `Matplotlib` based on the extracted data and the selected chart type for rendering the chart.

5 Experimental Setup

We conduct experiments using four LLMs, namely GPT-4o (OpenAI, 2024), Gemini-2.0 (Google, 2024), LLaMA-3.1-8B-Instruct (Meta, 2024) and Claude-3.5-Sonnet (Anthropic).

5.1 Baselines

We compare our approach against four baselines.

Single-Step Generation serves as the most straightforward approach, directly generating the chart from the input document and user intent without any intermediate retrieval.

Embedding-Based Retrieval incorporates a retrieval step as no parallel data tables are available for the intents. The document is segmented based on headings, and SBERT embeddings (all-MiniLM-L6-v2) (Reimers and Gurevych, 2019) are used to retrieve sections most relevant to the intent which are then used for chart code generation.

LLM-Based Retrieval builds on recent advances in LLM-powered retrieval (Zhu et al., 2024), which have demonstrated that LLMs can outperform embedding-based approaches by capturing richer contextual relationships between queries and documents. In this baseline, an LLM is used as a retriever before generating the chart code.

Query Decomposition for Table Retrieval takes advantage of the fact that most chart content comes from tables. Inspired by Chen et al. (2025), this method decomposes intents into (concept, attribute) pairs to enhance retrieval accuracy. Query decomposition is first applied, followed by LLM-based retrieval to extract relevant tabular data before generating the chart code.

In all the baselines, the chart type prediction is fixed to a naïve LLM instruction for a suitable type.

5.2 Evaluation Metrics

The generated charts should accurately reflect the underlying data from the documents and use appropriate chart types to convey the intended information. To evaluate these aspects, we use: (a) *chart data accuracy*, which includes completeness, correctness, and overall data quality with respect to the reference chart and (b) *chart type validation*, to compare the predicted type with the ground-truth chart types. For data accuracy, recent works either use n-gram text similarity metrics for the generated code (Han et al., 2023; Zadeh et al., 2024) or Visual Language Models (VLMs) to compare the generated and reference charts. However, such n-gram-based measures are known to be limited to surface-level aspects and fail to capture the nuanced error cases in the data values. VLMs often struggle with faithfully interpreting complex charts, especially when dealing with multiple data points or subtle variations in values (Huang et al., 2024), and are inherently limited by their visual perception accuracy (Ford et al., 2024).

We take inspiration in attribution as a strategy to validate specific spans of text by grounding them in the source context, both in textual space

Model	Method	Chart Data	Chart Type	
			Best	Out-of-3
GPT-4o	Single-step	67.38	71.79	75.63
	Embedding retrieval	38.98	35.04	38.97
	LLM retrieval	59.97	62.90	68.11
	LLM retrieval (w/ ques decomp)	57.09	68.45	73.44
	DOC2CHART	75.18	79.49	82.62
Gemini-2.0	Single-step	62.51	49.36	52.28
	Embedding retrieval	38.92	14.71	14.86
	LLM retrieval	59.21	45.55	48.63
	LLM retrieval (w/ ques decomp)	50.01	59.72	63.48
	DOC2CHART	71.53	74.12	79.41
Claude-3.5-Sonnet	Single-step	63.75	64.80	67.92
	Embedding retrieval	43.69	27.29	28.28
	LLM retrieval	58.60	61.50	66.33
	LLM retrieval (w/ ques decomp)	64.48	61.27	64.74
	DOC2CHART	69.45	82.01	84.13
LLaMA-3.1-8B-Instruct	Single-step	39.20	54.47	58.30
	Embedding retrieval	24.09	15.66	17.78
	LLM retrieval	27.11	40.67	43.31
	LLM retrieval (w/ ques decomp)	39.74	48.98	52.48
	DOC2CHART	42.17	71.75	78.05

Table 3: Performance comparison of various methods on chart data accuracy and chart type selection. Green highlights best performance; Red highlights second-best. Our methods DOC2CHART consistently outperform baselines.

and more recently for VLMs (Jiang et al., 2025; Phukan et al., 2025) We propose **CHARTEVAL**, an attribution-based metric that uses structured textual representations for charts, and avoids reliance on visual decoding altogether. We trace the intermediate representations of the generated charts, extracted as structured JSONs before code generation, back to their source tables in the document(s). Each chart is represented as a collection of tuples $\langle x\text{-axis}, y\text{-axis}, \text{value}\rangle$ in the JSON, each of which is individually validated against the values in the corresponding tables in ground truth. For this validation, we use a modified version of the attribution algorithm from (Phukan et al., 2024; Cohen-Wang et al., 2025): *(i)* We construct a prompt using the reference table[4] as the "document", and generated chart JSON as the "output". *(ii)* We forward pass this prompt through the Llama-3.1-8b-Instruct model and aggregate the cross attention scores between the "output" and "document" tokens to get a token-level heatmap of <output tokens, document tokens> size. *(iii)* For each data value token in the "output", we identify the best matching span in the "document" tokens using Kadane's algorithm (Kadane, 2023) on the obtained heatmap. While this formulation of CHARTEVAL does a reference-based evaluation with the ground truth tables, it can be extended to a reference-free variant as well, where the "document" would be the entire source document context in the LLM forward pass.

5.3 Human Evaluation for Metric Quality

To qualitatively validate our approach, we conduct human surveys to compare the model-generated charts against references. Each evaluation instance consists of an intent, ground truth (GT) chart, its corresponding GT table, along with three AI-generated charts (from our approach and two baselines), which are anonymized and randomized to eliminate any biases. We hire three expert annotators and provide them with 300 instances (150 from each domain) to rate them on six criteria: *Chart Data Correctness* to measure whether all values in the AI-generated chart match the reference data exactly; *Chart Data Completeness* to evaluate whether the chart includes all relevant values from the reference table; *Overall Chart Data Quality* to determine how accurately the chart conveys the reference data; *Chart Type Validation* to check if the selected chart type aligns with the given intent

[4] This can be used for reference chart as well by obtaining its JSON representation, if table references are not available.

Metric	**Single-Step**	**LLM-R w/ QD**	**Ours**
Chart Data Correctness	2.98	2.59	**3.52**
Chart Data Completeness	3.12	2.66	**3.63**
Overall Data Quality	2.98	2.59	**3.53**
Chart Type Validation	3.72	3.33	**4.13**
Insightfulness	2.77	2.43	**3.36**
Overall Quality	2.74	2.41	**3.34**

Table 4: Comparison of different methods against human ratings across evaluation axes. LLM-R w/ QD: LLM retrieval with query decomposition.

and data; *Chart Insightfulness* to assess how well the chart highlights key insights, adheres to intent, and whether visual encoding (e.g., colors, labels, legends) aids understanding; finally, *Overall Chart Quality* to assess the clarity, accuracy, and usefulness of the generated chart. Each criterion is rated on a four-point scale: *Minimal*, *Partial*, *Most*, and *Full*, where *Minimal* indicates the chart does not meet expectations at all, and *Full* represents an ideal chart. If a chart only partially contains correct values—either missing some or including incorrect ones—the rating is to be adjusted accordingly. We compute correlations between CHARTEVAL ratings and human ratings on these 900 samples, and find a strong alignment with human judgments (Pearson's $r = 0.71$). Using a simpler LLM-based metric, on the other hand, where the generated chart tuples and reference table values are given to an LLM which is then instructed to provide a rating for data accuracy, we note a much lower $r = 0.39$.

6 Results & Discussion

We evaluate the performance of DOC2CHART pipeline using four LLMs: GPT-4o, Gemini-2.0, Claude-3.5-Sonnet, and LLaMA-3.1-8B-Instruct (Table 3). In terms of chart data accuracy, DOC2CHART significantly outperforms all baselines. For instance, with GPT-4o, DOC2CHART achieves 75.18% accuracy, a notable improvement over that for the single-step baseline (67.38%) and other retrieval-augmented approaches such as LLM retrieval (w/ ques decomp) (57.09%). Similar trends are observed for other models; with Gemini-2.0, DOC2CHART (71.53%) substantially surpasses LLM retrieval (w/ ques decomp) (50.01%). This underscores the benefit of the iterative refinement process in improving the factual correctness of the extracted data. Similar gains are observed in chart-type prediction across models using the heuristic-based analysis using LLM, compared to naïve prompting. The results consistently show that improving chart data accuracy through validation and refinement directly translates to better chart type recommendations as well.

Among the baselines, the single-step one performs the best in most cases for data accuracy. Embedding-based retrieval consistently underperforms, lacking the necessary contextual depth for accurate chart data extraction. While LLM-based retrieval, especially when combined with query decomposition, shows some improvement over simple embeddings, it still lags behind. The query decomposition baseline struggles as it only breaks down the intent into broad topics and attributes rather than structured data tuples. For example, when given the intent "*Assess the hotel revenues for the top 5 highest performing regions from 2021 to 2023, focusing on the trends in revenue growth and regional performance,*" it outputs generic components like `<sub_c>hotels:revenue</sub_c>` and `<sub_c>revenue:trend</sub_c>`. However, these lack the necessary structure to retrieve precise data. Table 4 shows the human ratings for the generations using our approach and two other baselines (taking majority rating for each sample and averaging across samples). Our method's outputs are consistently rated higher than those generated by the baselines, and the single-step baselines are rated as the next best.

7 Conclusions & Future Work

We present the task of intent-based chart generation from documents, where the objective is to generate charts that not only align with a user-specified intent but are also grounded in the source document. In contrast to prior datasets that focus on table-to-chart or plain text-to-chart generation, our dataset includes 1,242 (intent, document, chart) tuples, reflecting more realistic and open-ended scenarios. Our unsupervised, multi-stage framework decomposes the user intent to iteratively extract and refine relevant data,selected appropriate chart type using visualization heuristics, and generates executable chart code in a zero-shot manner. While ideal data accuracy would be close to 1—especially to assist users in bypassing the need to manually navigate long documents—our method consistently outperforms strong baselines, including single-shot generation and retrieval-based methods, across both data accuracy and chart type selection. To increase trust in AI-generated charts, we advocate for chart attribution: tracing chart values back to their textual sources. Future work can extend this by attributing data values not just to tables but to specific text spans. Attribution failures may also serve as useful feedback signals to guide data refinement. While human evaluations reflect similar trends, it is important to note that chart type selection can be inherently subjective—multiple chart types may be valid depending on the user's analytic goal—so future evaluation strategies should account for this flexibility. We hope that our work paves the way towards developing more accessible, intent-driven document visualizations, with potential applications in domains like finance, science, and public policy.

8 Limitations

User intents can often be vague or underspecified, leading to multiple valid interpretations. While our framework performs intent decomposition and iterative refinement to approximate the user's needs, it does not incorporate dynamic user feedback or interactive clarification. Although interactive mechanisms—where users confirm or adjust extracted information—could further improve alignment with user expectations, our focus is on generating a high-quality first draft without requiring user intervention. Additionally, due to the limited context window of current LLMs, scaling the approach to handle multiple long documents remains a challenge. Lastly, while our chart attribution metric helps evaluate factual grounding, it is currently implemented in a reference-based setting—comparing chart data directly against source tables. This metric can be extended to a reference-free setup, where the attribution model takes the generated table and raw document markdown(s) as input. While this improves scalability, it may occasionally attribute values to the wrong context if the same value appears elsewhere in the document—a challenge we leave for future work.

9 Ethical Statement

Automated chart generation carries the risk of producing misleading visualizations, especially in high-stakes domains such as finance, science, and policy. To mitigate these risks, our work emphasizes faithfulness and transparency by introducing chart attribution—a method that traces chart content back to its source tables—alongside quantitative evaluation grounded in document data. We also avoid generating speculative content and restrict chart construction to source-supported values only. Nonetheless, we acknowledge that LLMs may still produce flawed outputs (hallucinations). Future work should explore mechanisms for uncertainty estimation, user-in-the-loop validation, and better safeguards to ensure responsible deployment of automatic charting systems.

References

Anthropic. Introducing Claude 3.5 Sonnet.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation.

Peter Baile Chen, Yi Zhang, and Dan Roth. 2025. Is table retrieval a solved problem? exploring join-aware multi-table retrieval.

Benjamin Cohen-Wang, Yung-Sung Chuang, and Aleksander Madry. 2025. Learning to attribute with attention.

James Ford, Xingmeng Zhao, Dan Schumacher, and Anthony Rios. 2024. Charting the future: Using chart question-answering for scalable evaluation of llm-driven data visualizations.

Google. 2024. Gemini 2.0 flash.

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

Kung-Hsiang Huang, Mingyang Zhou, Hou Pong Chan, Yi Fung, Zhenhailong Wang, Lingyu Zhang, Shih-Fu Chang, and Heng Ji. 2024. Do LVLMs understand charts? analyzing and correcting factual errors in chart captioning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 730–749, Bangkok, Thailand. Association for Computational Linguistics.

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

Nick Jiang, Anish Kachinthaya, Suzie Petryk, and Yossi Gandelsman. 2025. Interpreting and editing vision-language representations to mitigate hallucinations.

Joseph B. Kadane. 2023. Two kadane algorithms for the maximum sum subarray problem. *Algorithms*, 16(11):519.

Woosung Koh, Jang Han Yoon, MinHyung Lee, Youngjin Song, Jaegwan Cho, Jaehyun Kang, Taehyeon Kim, Se young Yun, Youngjae Yu, and Bongshin Lee. 2024. c^2: Scalable auto-feedback for llm-based chart generation.

Paula Maddigan and Teo Susnjak. 2023. Chat2VIS: Generating Data Visualizations via Natural Language Using ChatGPT, Codex and GPT-3 Large Language Models. *IEEE Access*, 11:45181–45193.

Meta. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

OpenAI. 2024. Gpt-4 technical report.

Anirudh Phukan, Divyansh, Harshit Kumar Morj, Vaishnavi, Apoorv Saxena, and Koustava Goswami. 2025. Beyond logit lens: Contextual embeddings for robust hallucination detection & grounding in vlms.

Anirudh Phukan, Shwetha Somasundaram, Apoorv Saxena, Koustava Goswami, and Balaji Vasan Srinivasan. 2024. Peering into the mind of language models: An approach for attribution in contextual question answering.

Pritika Ramu, Pranshu Gaur, Rishita Emandi, Himanshu Maheshwari, Danish Javed, and Aparna Garimella. 2024. Zooming in on zero-shot intent-guided and grounded document generation using LLMs. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 676–694, Tokyo, Japan. Association for Computational Linguistics.

Md. Mahinur Rashid, Hasin Kawsar Jahan, Annysha Huzzat, Riyasaat Ahmed Rahul, Tamim Bin Zakir, Farhana Meem, Md. Saddam Hossain Mukta, and Swakkhar Shatabda. 2021. Text2chart: A multi-staged chart generator from natural language text.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. 2024. Assisting in writing Wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6252–6278, Mexico City, Mexico. Association for Computational Linguistics.

Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy X. R. Wang. 2021. D2S: Document-to-slide generation via query-based text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1405–1418, Online. Association for Computational Linguistics.

Yuan Tian, Weiwei Cui, Dazhen Deng, Xinjing Yi, Yurun Yang, Haidong Zhang, and Yingcai Wu. 2024. Chartgpt: Leveraging llms to generate charts from abstract natural language. *IEEE Transactions on Visualization and Computer Graphics*, page 1–15.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Lei Wang, Songheng Zhang, Yun Wang, Ee-Peng Lim, and Yong Wang. 2023. Llm4vis: Explainable visualization recommendation using chatgpt.

Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Min Dou, Botian Shi, Junchi Yan, and Yu Qiao. 2025. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning.

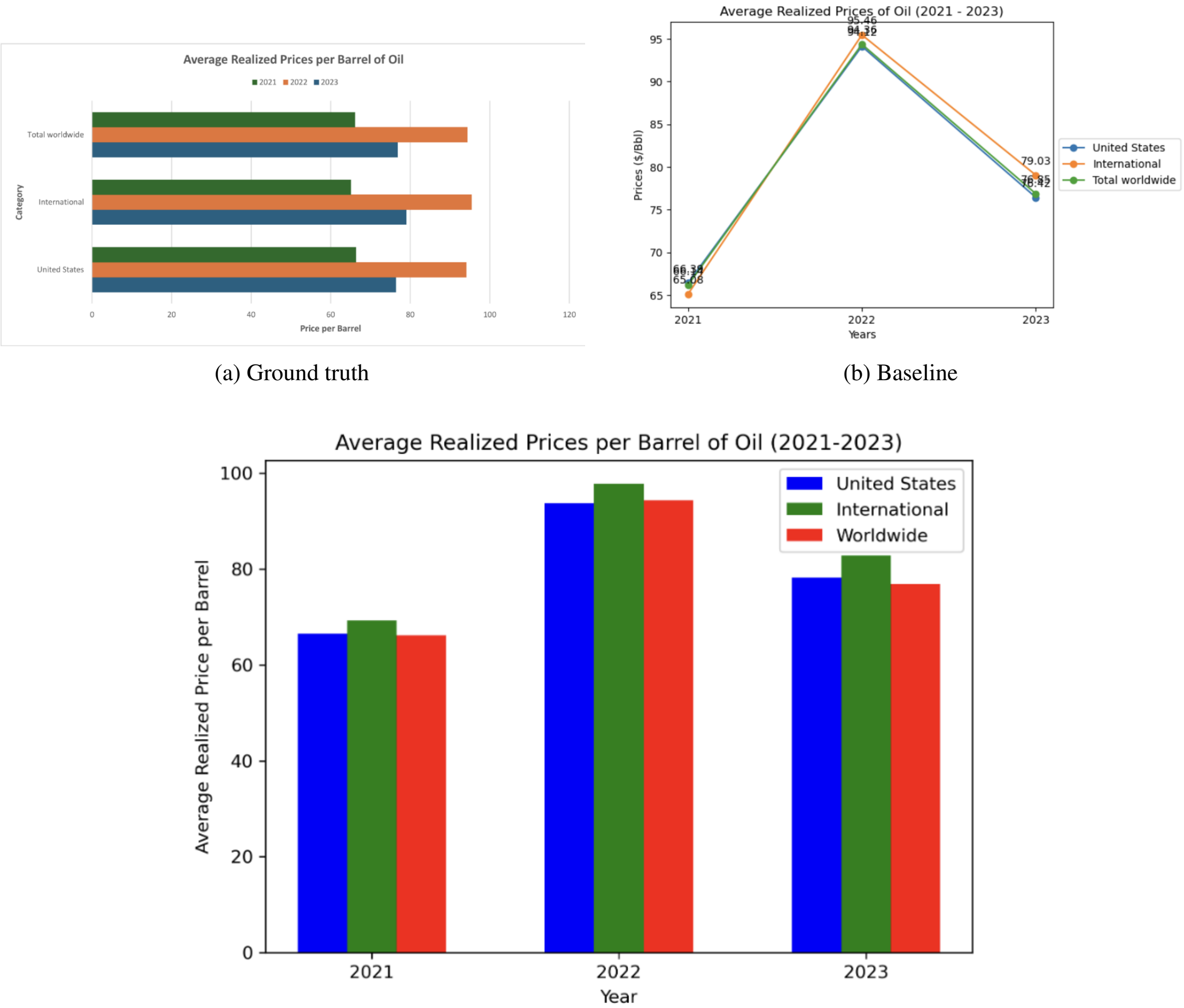
Fatemeh Pesaran Zadeh, Juyeon Kim, Jin-Hwa Kim, and Gunhee Kim. 2024. Text2chart31: Instruction tuning for chart generation with automatic feedback.

Songheng Zhang, Lei Wang, Toby Jia-Jun Li, Qiaomu Shen, Yixin Cao, and Yong Wang. 2024. Chartifytext: Automated chart generation from data-involved texts via llm.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2024. Large language models for information retrieval: A survey.

A Appendix

A.1 Qualitative Examples



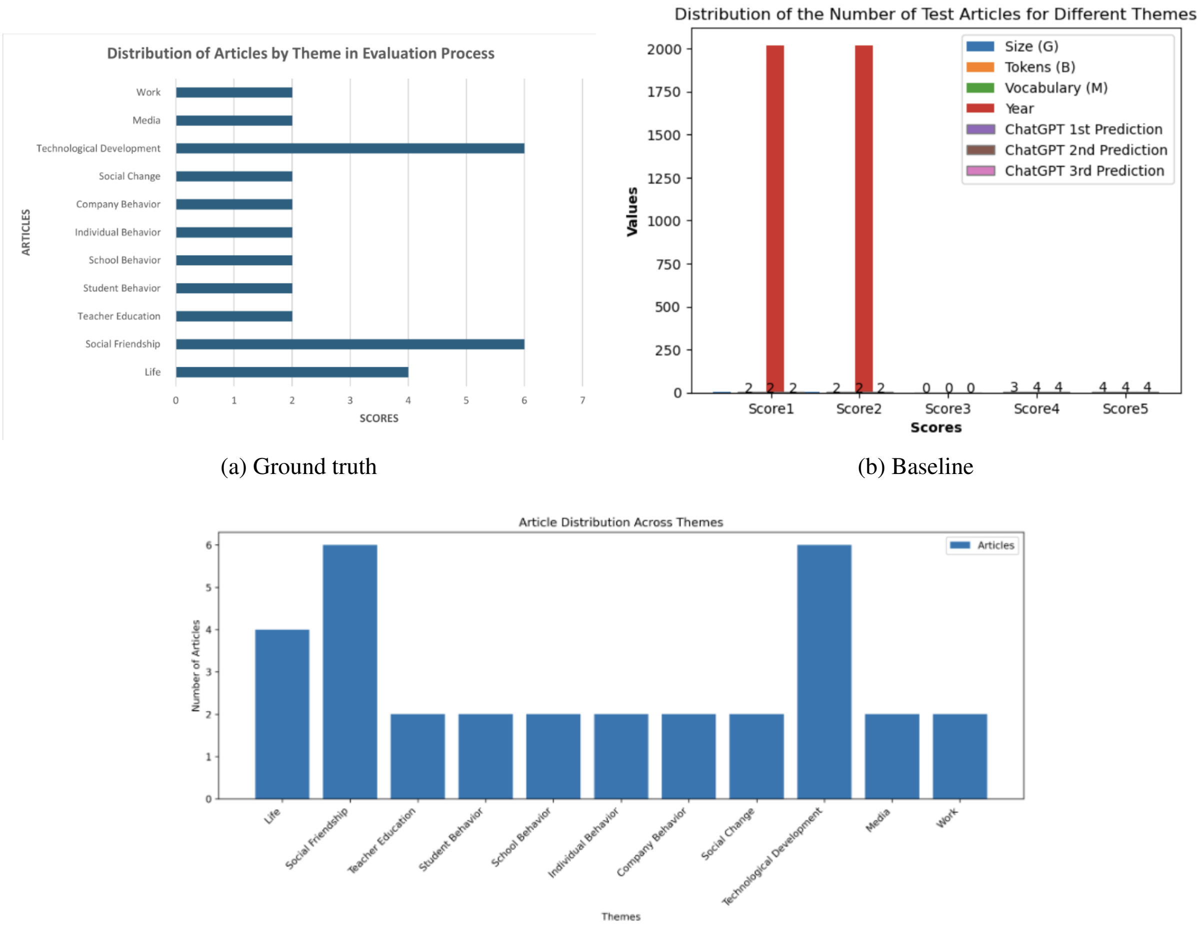
(a) Ground truth

(b) Baseline

(c) Ours

Figure 4: Qualitative example for intent-driven chart generation on a financial document. **Intent:** Examine the average realized prices per barrel of oil in the United States, International markets, and globally for the years 2021–2023.

A.2 Prompt Templates



(a) Ground truth

(b) Baseline

(c) Ours

Figure 5: Qualitative example for intent-driven chart generation on a research paper. **Intent:** Analyze the distribution of articles across different themes in the evaluation process, focusing on thematic representation and balance.

Task:
Extract structured chart data from the provided content based on the user's intent, adhering to the specified JSON format.
Input:

- User Intent: `{intent}`
- Content: `{content}`
- Optional Feedback (if available): `{optional_feedback_section}`
- Output Format Schema: `{output_format}`

Instructions:

1. Carefully read the User Intent.
2. **Internal Thought Process (Mentally follow these steps):**
   - **Decompose:** Break down the intent into specific data points, labels, categories, and title.
   - **Locate:** Scan the content for exact data matching the above.
   - **Extract & Structure:** Collect and format data strictly according to the schema.
3. Extract relevant data points: $(x, y, \text{category})$, axis labels, and chart title.
4. **If feedback is provided:** Focus on fixing issues like missing elements or ignored sections. Adjust your decomposition and extraction accordingly.
5. Output must follow the JSON schema exactly. Keep numeric formats consistent.
6. Output only the JSON object. **Do not** include explanations or markdown like “‘`json`.

Example Output Format:

```
{
  "values": [
    {
      "x": "[string or number]",
      "y": "[number or string representing number]",
      "category": "[string, optional]"
    }
  ],
  "x_axis_label": "[string]",
  "y_axis_label": "[string]",
  "title": "[string]"
}
```

Figure 6: Chart Data Extraction

Task: Validate the extracted chart data against the source content and user intent. Determine if re-extraction is necessary or if only minor refinements are needed.
Input:

- Original Intent: `{intent}`
- Source Content: `{content}`
- Extracted Chart Data: `{extracted_data}` *// JSON object from the extraction step*
- Expected Schema: `{output_format}`

Validation Checks to Perform:

1. **Intent Fulfillment & Source Coverage:** Does the `extracted_data` capture the key information requested in the `intent` that is present in the `Source Content`? Are there critical omissions?
2. **Data Accuracy:** Are the values (`x`, `y`, `category`) and labels/title in `extracted_data` accurately reflecting the `Source Content`?

Response Format:

```
{
  "needs_re_extraction": "[true/false]",
  "feedback_for_re_extraction": "[string]",
  "suggested_corrections_for_refinement": [
    {
      "field_path": "[JSON path, e.g., values[0].y or title]",
      "suggestion": "[Brief description of the fix, e.g., 'Convert to number']",
      "suggested_value": "[Optional: The corrected value if easily determined]"
    }
  ],
  "confidence_score": "[0-10 score reflecting confidence in the data]"
}
```

Focus on the primary decision: re-extract or refine/accept. Keep feedback concise.
Output only a valid JSON and no other text. Do not add prefix like “‘`json...`

Figure 7: Chart Data Validation

Task: Apply the suggested minor corrections to the extracted chart data.
Input:

- Original Intent: `{intent}`
- Source Content: `{content}`
- Extracted Data (Pre-Refinement): `{extracted_data}`
- Suggested Corrections: `{suggested_corrections}`
- Expected Schema: `{output_format}`

Instructions:

1. Iterate through the `Suggested Corrections`.
2. Apply each correction to the corresponding `field_path` in the `Extracted Data`. Use `suggested_value` if provided, otherwise interpret the `suggestion`.
3. Ensure the final `refined_data` strictly follows the `Expected Schema` provided in the input.
4. Do **not** add new data or make changes beyond the `Suggested Corrections`.

Response Format:

```
{
  "refined_data": [The data structure with corrections applied, adhering to the Expected Schema],
  "refinement_summary": {
    "changes_applied_count": [number],
    "issues_applying_corrections": [List any suggestions that could not be applied and why]
  }
}
```

Output only a valid JSON and no other text. Do not add prefix like “‘json...“‘

Figure 8: Chart Data Refinement

Task:
Compare a Ground Truth Table and a Predicted Chart JSON in terms of data accuracy by:

- Ensuring consistent numeric formatting.
- Directly comparing values, accounting for slight paraphrasing of attributes and entities.
- Providing a structured output showing discrepancies and an overall accuracy score.

Input:

- **Ground Truth Table:** A table with rows containing values for entities and attributes (e.g., years, categories, months).
- **Predicted Chart JSON:** Contains a `final_output` with `x_labels`, `y_labels`, and detailed values.

Instructions:

1. **Number Formatting Consistency:**
    - Scan both datasets to identify the common number format (e.g., decimal precision, thousands separators).
    - Ensure all values follow the same formatting rules (e.g., convert ‘1,234.5’, ensure uniform decimal precision, remove trailing zeros).
2. **Compare Ground Truth and Predicted Data:**
    - Directly compare values, ensuring entities and attributes match, even if paraphrased (e.g., “Sales Revenue” vs. “Revenue from Sales”).
    - Identify and list discrepancies where values differ.

Figure 9: Prompt for LLM-based Data Accuracy Evaluation