

**CONNECTING TEXT AND CHARTS USING LARGE
VISION-LANGUAGE MODELS**

NAFIS TAHMID CHOWDHURY

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
FEBRUARY 2026

© NAFIS TAHMID CHOWDHURY, 2026

Abstract

Data visualizations are essential for presenting complex dataset, but the disconnect between charts and accompanying textual descriptions often leads to misinterpretation and increased cognitive effort—especially for users with limited data literacy. While prior methods attempt to bridge this gap, many depend on manual annotations or fixed chart structures, limiting scalability across diverse documents. In this thesis, we propose two large vision-language model (LVLM)-based frameworks—a single-agent baseline and a multi-agent architecture—for automatically linking textual descriptions with their corresponding chart data. Both frameworks extract structured data from chart images and use lexical, syntactic, and arithmetic reasoning to perform sentence-to-data alignment. We evaluate the performance of these frameworks on a curated dataset of Pew Research charts. Finally, we develop a browser extension that integrates this approach into Pew Research articles, enabling interactive text–chart linking for enhanced reading experiences.

Dedication

I dedicate this thesis to my father Md Abdul Monayem Chowdhury and, my mother, Nasrin Jahan. It's only because of their great care and sacrifice, I have become what I am today.

Acknowledgements

I would like to begin by expressing my deepest gratitude to my parents, whose constant support, sacrifices, and encouragement have been the greatest source of strength throughout my academic journey. Their unwavering belief in me and their continued motivation made it possible for me to pursue and complete this degree. I am profoundly thankful for everything they have done for me.

I would like to sincerely thank my supervisor, Dr. Enamul Hoque Prince, for his guidance and support throughout the course of this research. His feedback and direction were essential in shaping this thesis, and his mentorship helped me navigate both the technical and academic aspects of the work.

I also thank my thesis committee member, Dr. Sana Maqsood, for her time, feedback, and constructive suggestions, which contributed to improving the quality and clarity of this thesis.

I am grateful to be part of the Intelligent Visualization Lab at York University and would like to thank the lab members for their collaboration and discussions.

The research environment provided by the lab was instrumental to my progress.

Table of Contents

Abstract	ii
Dedication	iii
Acknowledgements	iv
Table of Contents	vi
List of Tables	x
List of Figures	xii
Preface	xvii
1 Introduction	1
1.1 Motivation	1
1.2 Research Approach	6
1.3 Contributions	8

1.4	Organization of the Thesis	9
2	Literature Review	11
2.1	Text-Visualization Pairing	12
2.1.1	Automated and Annotation-Based Approaches	12
2.1.2	Narrative Integration and Interactive Linking	13
2.1.3	Natural Language and User-Centric Interfaces	14
2.1.4	Applications in Multimedia	16
2.2	Interactive Document Reading	18
2.2.1	Contextual Glossing	19
2.2.2	Interactive Citation Reading and Summarization	20
2.2.3	Text–Visualization Linking and Guided Reading Interfaces .	21
2.3	Chart Data Extraction	25
2.4	LLM Agents	27
2.4.1	LLM Agents in Various Domains	27
2.4.2	Agentic Approaches with Charts	28
3	Methodology	30
3.1	Task Formulation	31
3.2	Dataset Construction	33
3.2.1	Description Generation	35

3.2.2	Text-Data Pairing Identification	36
3.2.3	Evaluation Dataset for Chart Object Detection	38
3.3	Chart-Text Agent Pairing Approach	38
3.4	Browser Extension Development	43
3.4.1	Content Extraction	45
3.4.2	Paragraph-Chart Association	45
3.4.3	Sentence-Chart Pairing	47
3.4.4	Interactive Chart Annotations	49
3.4.5	Persistent Chart Display	50
3.5	Summary	51
4	Evaluation and Results	52
4.1	Reference Extraction Evaluation Criteria	53
4.2	Reference Extraction Evaluation Process	55
4.2.1	Prompt Strategy Evaluation	56
4.2.2	Model Comparison	57
4.2.3	Error Analysis	58
4.3	Web Browser Extension Evaluation	59
4.4	Experiments and Results	61
4.4.1	Reference Extraction Evaluation	61

4.4.2	Web-browser extension	68
4.5	Summary	70
5	Conclusions and Future Work	72
5.1	Conclusions	72
5.2	Future Work	73
	Bibliography	75
A	Appendix	85

List of Tables

2.1	Taxonomy of Text–Visualization Pairing Literature	18
2.2	Taxonomy of Interactive Document Reading Literature	24
3.1	Distribution of chart types with sentence statistics and percentage breakdowns	33
4.2	Sentence–data pairing accuracy across models on Pew Research charts (bar, line, pie) and the Kong et al. dataset under two evaluation set- tings.	64
4.1	Comparison of pairing accuracy results across prompting strategies	64
4.3	F1-scores for sentence-data pairing on the Kong et al. dataset under pairing-only and full-pipeline evaluation settings.	65
4.4	Count and percentage of unparsable files generated by each model, per chart type.	68

4.5	Mark detection results across charts in Kong et al. comparing Pro-	
	posed Method vs. ChartOCR Baseline	69

List of Figures

- 1.1 Overview of the sentence-to-chart pairing process. (1) A Pew Research bar chart is decomposed into a structured data table. (2) The table and accompanying description are analyzed by the framework. (3) Our Agentic framework aligns sentences with bars in the chart, highlighting linked chart elements and textual references to show how descriptions are grounded in data. 3
- 1.2 An overview of the text-visual pairing process initiated upon activation of the Chrome browser extension. The extension parses a Pew Research web page (left), separating article text from chart images. The Agentic framework (center) then pairs text with chart data using vision-language models and highlights links between sentences and chart elements (right). 5

3.1	An overview of the chart types included in the dataset: (A) bar charts, (B) line charts, and (C) pie charts, sourced from Pew Research Center articles.	34
3.2	Process for generating textual chart descriptions from question–answer pairs. Question-answer pairs from ChartQA are gathered for each chart image (A). Question-answer pairs associated with each chart are then converted into assertive sentences by the large language model (B).	36
3.3	An overview of the proposed agentic framework . The chart image in 1 (A) is first processed using an LV-LM (B) that extracts structured chart data in the form of a Markdown-formatted table. The textual description associated with the chart, in addition to the data table (C), is given as an input to a second prompt (D), which identifies matching sentences and aligns them with corresponding rows of the data. The output (E) returns the sentence–data pairings in JSON format, where each sentence is linked to specific data tuples from the table.	39
3.4	An outline of the Chart Deconstruction prompt strategy, showcasing chart image as input, processes followed for extracting data, and generated output.	40

3.5	An outline of the prompt strategy followed for pairing textual descriptions with data, showing structured table data and textual description, steps followed for the pairing process, and paired data as output.	42
3.6	Overview of the sentence-to-chart pairing and overlay generation pipeline. Paragraph text, chart metadata, and data tables (A) are used to associate sentences with candidate charts using prompts (B). Sentence-data pairs (C) are grounded to visual chart marks (D) to generate overlay regions that highlight referenced data points (E). The final output (F) presents an augmented reading interface with synchronized text highlights and chart overlays.	44
3.7	An outline of the prompt strategy followed for pairing paragraphs in Pew Research article with their corresponding embedded charts. .	46
3.8	An outline of the prompt strategy followed for aligning textual descriptions paired with structured data to their corresponding visual marks in embedded chart images. In this strategy, the visual groundings of chart marks is given to the prompt, which then follows the outlined steps for associating with their corresponding marks	48

4.1	Illustration of the browser extension’s deconstruction of a Pew Research webpage, where the original article view (A) is parsed to separate text from embedded charts and reorganized into a processed layout (B) with sentences linked to corresponding visual elements. Selecting a linked sentence highlights the relevant chart marks and renders the full chart in a persistent display panel (C), keeping it visible during scrolling.	71
A.1	Chart data extraction prompt used for automatic chart-to-table generation.	86
A.2	Zero Shot Sentence-data pairing prompt used for linking textual descriptions to chart data.	87
A.3	Few Shot Sentence-data pairing prompt template used in the structured chart understanding pipeline.	90

A.4	An overview of sentence–data pairing outcomes produced by the sentence-data pairing framework. Example outputs illustrate different alignment cases: a correct sentence–data match (true positive), a missed data tuple despite a correct sentence (false negative), an incorrectly inserted data tuple (false positive), and a combined error where the paired sentence does not match the description, resulting in both an incorrect pairing and a missed correct alignment (false positive + false negative)	90
-----	--	----

Preface

This thesis is submitted to the Faculty of Graduate Studies in partial fulfillment of the requirements for a Master of Science Degree in Computer Science. The entire work presented here is done by the author **Nafis Tahmid Chowdhury** under the supervision of **Dr. Enamul Hoque Prince**.

1 Introduction

1.1 Motivation

Data visualization is the graphical representation of information through using visual elements such as charts and graphs. They help in conveying trends, comparisons and statistical findings, leading to their widespread adoption across news media, research articles, textbooks, business analytics, and software tools [6]. Many research institutions and media outlets such as Pew Research Center [59] frequently publish articles on topics such as politics, religion and technology that include charts to summarize key information. However, despite the usefulness of charts, many readers struggle with effectively connecting textual descriptions and their corresponding visualizations. In such reports, sentences that describe the data of the charts tend to be positioned separate from the charts themselves, with the descriptions being either above, below or on a different section of the page in relation to the charts.

This layout requires readers to manually cross-reference the charts with the cor-

responding textual descriptions that are in differing locations, resulting in them shifting their focus between both the charts and the text. Such effort causes the attention of the readers to be divided across the two sources of information, resulting in an extra cognitive burden that is termed as the split-attention effect [9]. The lack of clear connections between the textual descriptions and the corresponding charts results in a cognitive load in the readers from retaining comprehension of one source of information while manually locating another corresponding source of information in the data report. This can lead to readers misinterpreting the patterns, comparisons and statistical findings present in the report, resulting in the readers developing a skewed understanding of the data, especially among readers with limited data literacy [55]. The increasing use of data visualizations in research articles highlights the growing need for effective methods that help readers integrate textual explanations with graphical data representations, which could improve comprehension, reduce reader effort and make data-driven reports more accessible [7].

Various research has been done on linking texts with their corresponding visualizations. Zhi et al. [86] demonstrated that explicitly connecting descriptive text with visual elements improves readability, comprehension, and retention. On approaches to linking texts with visualizations, both manual and automated methods have been explored. Kong et al. [40] introduced a crowdsourcing approach

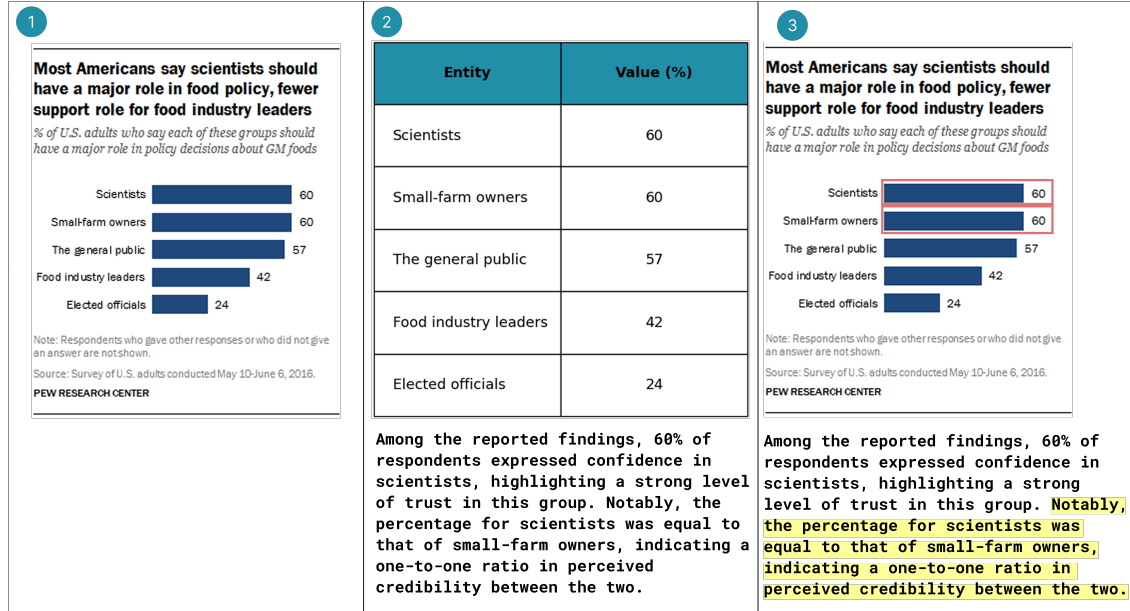


Figure 1.1: Overview of the sentence-to-chart pairing process. (1) A Pew Research bar chart is decomposed into a structured data table. (2) The table and accompanying description are analyzed by the framework. (3) Our Agentic framework aligns sentences with bars in the chart, highlighting linked chart elements and textual references to show how descriptions are grounded in data.

for generating high-quality text-to-chart mappings, where human annotators manually paired phrases from the text with corresponding visual marks in the charts. While accurate, such manual annotations lack scalability. In contrast, automated methods have attempted to address this issue by linking texts to chart elements through natural language processing techniques and integrating them within interactive reading system to keeps chart visualizations contextually linked to nearby textual explanations, enhancing user comprehension and engagement [60, 75, 44].

Beyond visual charts, research has also been done on linking texts with tabu-

lar data. Kim et al. [38] introduced a pipeline for automatically linking textual references to table data, improving document readability and reducing the split-attention effect. Zhu et al. [88] further explored maintaining persistent links between textual descriptions and datasets, allowing users to interactively manipulate and update text-based data references.

However, most existing systems that link textual descriptions with visualizations either rely on manual crowdsourcing pipelines [40] or use automated rule-based natural language processing methods [60, 38]. While these approaches work well in controlled settings where the underlying data is readily available, they are difficult to apply to data-driven reports that are published by Pew Research, which consist of multiple charts and tend not to expose the underlying data used in the charts, thereby making it challenging for such systems to function reliably. In addition, traditional natural language processing systems tend to rely on fixed templates or heuristic rules for detecting references made in textual descriptions [60], which can make them less robust when evaluating textual descriptions present in data-driven reports, which can use complex sentence structures. Moreover, they are not designed to work directly within web browsing environments, which limits their ability to handle the scale and variability of content found across Pew Research publications. Recent advances in large vision-language models, which are trained on large datasets of images and text, makes them well-suited for understanding and

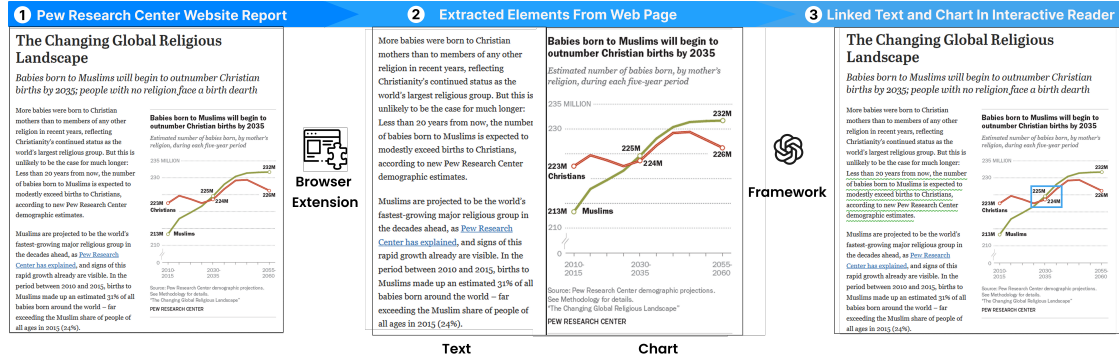


Figure 1.2: An overview of the text-visual pairing process initiated upon activation of the Chrome browser extension. The extension parses a Pew Research web page (left), separating article text from chart images. The Agentic framework (center) then pairs text with chart data using vision-language models and highlights links between sentences and chart elements (right).

aligning information across charts and their associated descriptions, even when the structure of the accompanying text is complex .

In this work, the central goal is to develop a robust system that not only links textual descriptions in Pew Research articles with their corresponding chart data, but also presents these connections through an interactive reading interface that persistently displays the relevant chart. This design directly addresses the split attention effect by keeping visual context visible as readers move through the article, thereby reducing cognitive effort and improving comprehension.

1.2 Research Approach

In this work, the central goal is to present a system that can assist readers in better interpreting data-driven articles published by Pew Research through linking textual descriptions with their corresponding chart elements and presenting the connections through an interactive reading interface that persistently displays the relevant chart with annotated chart elements. This design addresses the split attention effect by keeping the annotated chart in view as readers engage with the corresponding text, thereby lowering cognitive effort and enhancing comprehension. To achieve this, we divide the approach into two parts. First, we develop a sentence-to-data pairing framework using large vision language models (LVLM), which are artificial intelligence systems trained on large amounts of textual and image data that make them capable of understanding and generating natural language to perform a wide range of tasks. LVLMs are also capable of reasoning tasks, such as making comparisons, performing arithmetic operations or interpreting descriptions. This enables LVLMs to identify individual sentences in a paragraph and determine the corresponding data values that it references for a chart. For instance, if a sentence says, "Among the reported findings, 60% of respondents expressed confidence in scientists, highlighting a strong level of trust in this group", the LVLM will identify the visual element in the chart (e.g axes, legends, bars, point) corresponding to the

data and for a link between the two.

Leveraging the capabilities of large vision-language models (LVLMs), we develop an agentic framework for pairing textual descriptions with chart data. The framework employs an LVLM agent to process a chart image by extracting its underlying structure and converting it into a structured data table. Using this representation, the agent analyzes the associated textual description and matches each sentence to its corresponding structural data by identifying key values, comparative expressions, and relationships that may involve arithmetic reasoning. To evaluate the performance of both frameworks, we construct a dataset of 350 charts from Pew Research articles, each paired with a cleaned data table and a descriptive paragraph derived from the ChartQA dataset [49].

Second, we extend the sentence-to-data pairing framework into a web-based browser extension that allows users to read Pew Research articles in an interactive manner. The extension is designed to function directly on the Pew Research website. Once activated, the extension scans the web page to extract both the chart images and the article text. These extracted elements are then processed through the LVLM-based pairing framework to pair sentences corresponding with specific data points in the chart.

Sentences identified and connected to chart data are visually highlighted within the article to denote the pairing. When a user hovers or clicks on a highlighted

sentence, the corresponding part of the chart is annotated to emphasize the relevant chart data. To further reduce the cognitive effort associated with switching between texts and charts, the matched chart is displayed persistently in a pinned sidebar as the reader scrolls through the article. This persistent display ensures that the chart remains visible during the reading process, allowing the reader to interpret the text and chart together, thereby helping to mitigate the cognitive burden that comes from the split attention effect.

1.3 Contributions

To support the goal of improving the readability of data-driven articles, our contributions are as follows:

- (i) We curate a benchmark dataset comprising 350 charts published by Pew Research. Each chart is annotated with a cleaned Markdown-formatted data table and a descriptive paragraph, resulting in 541 manually verified sentence-to-data pairings. This dataset serves as the foundation for both framework development and evaluation.
- (ii) We develop a large vision language model-based framework for linking descriptive sentences to their corresponding chart data. The framework is designed to identify and pair textual explanations with rows in a structured

data table extracted from chart images, using lexical, syntactic, arithmetic, and semantic reasoning.

- (iii) We implement a browser extension that integrates our pairing framework into a real-world reading environment. This extension automatically highlights sentences matched to chart data and persistently displays the corresponding chart with annotated elements, reducing the reader’s cognitive effort and supporting comprehension through synchronized visual context.

1.4 Organization of the Thesis

In this section, we outline the structure of the remaining chapters of this thesis. Chapter 2 reviews prior research across four main areas: (i) text–visualization pairing, (ii) interactive document reading systems, (iii) chart data extraction techniques, and (iv) large vision language model–based agent frameworks applied to data visualization. Chapter 3 presents the methodology of our work, beginning with the task formulation for sentence-to-data pairing, followed by the design of the proposed agentic framework, the development of the browser extension, and the construction and validation of the benchmark dataset. Chapter 4 describes the evaluation methodology, experimental setup, and results, which include the criteria for measuring pairing accuracy, a comparative analysis across models, error

analysis for hallucination detection, and results from both sentence-to-data pairing experiments and the browser extension deployment. Finally, Chapter 5 concludes the thesis with a summary of key findings, limitations of the current study, and directions for future work.

2 Literature Review

This chapter focuses on two core domains central to our work: text–visualization linking and interactive document reading. We begin by reviewing research that addresses the problem of aligning textual descriptions with chart components. We then examine interactive document reading systems that aim to improve comprehension through techniques such as contextual glossing, citation tracking, and integrated text-visual interfaces. In addition to these main areas, we explore techniques to extract data from charts, an auxiliary domain that provides necessary groundwork for understanding and interpreting chart images, especially when raw data is unavailable. Finally, we discuss recent work on LVLMM-based agentic frameworks for data visualization, outlining how agent-driven approaches have been applied to tasks such as visualization generation, insight extraction, and chart editing, and examining how these frameworks can be leveraged to enhance the performance of sentence-to-chart alignment.

2.1 Text-Visualization Pairing

2.1.1 Automated and Annotation-Based Approaches

Several approaches have been developed to automatically detect chart elements and link textual references to them. ChartText [60] introduces a two-stage approach for linking textual phrases with chart elements through application of various syntactic matching and comparison techniques. Kori [45] proposes a mixed initiative interface that suggests potential chart references through matching text segments to chart features and allows users to manually select and link text phrases to corresponding chart elements. VizFlow [75] explores text-to-chart linking in data driven articles through using a CNN-based chart parsing module to offer automated suggestions for common chart elements and offering side-by-side text-chart views so that the visualizations remain contextually linked to the textual content. Other systems focus on enhancing interpretability through visual explanations. Voder [74] enhances user understanding of visualizations by generating and linking system-suggested statistical insights that users can interact with to trigger alternative visualizations or visual embellishments. Latif et al. [44] introduces a system that parses textual markups in web documents and links them with generated, interactive visualizations such as sparklines and figures.

Research has explored emphasizing key features and annotations. Kim et al. [39]

introduces *EmphasisChecker*, a system that identifies prominent features in time-series charts, such as peaks and trends, using the Ramer-Douglas-Peucker algorithm and compares them with emphasized textual references to detect discrepancies and guide authors in refining their descriptions. Lai et al. [42] uses object detection and natural language processing modules for identifying and linking key visual elements in charts to their textual references, after which the system automates annotations that highlight relevant data points during reading. Ren et al. [64] enables authors to attach text directly to chart data marks, producing annotations that move with the underlying data and ensuring textual highlights remain visually grounded. Chen et al [12, 13] similarly connects user-authored textual notes with chart insights, such as clusters and outliers, enabling annotations with rich semantic meanings to be semi-automatically generated and embedded within the visualization, thereby lower typing effort and improving the quality of the generated annotations.

2.1.2 Narrative Integration and Interactive Linking

Integration of visualizations with text is prominent within data-driven storytelling. *ChartStory* [84] automatically partitions collections of charts into narrative segments by grouping related visualizations based on shared data characteristics and generates captions so that each visual segment is explained with contextual text. Bryan et al [8] combines time-series plots with annotated textual snapshots, where

significant events or user-selected points are highlighted and paired with text so that written descriptions are anchored at precise moments in the data. Similarly, Hullman et al [32] extracts textual events from news articles and embedding them as annotations at corresponding points on stock charts, directly linking narrative context to visual fluctuations.

Contrasting these automated linking methods, Kong et al.[40] presents a crowdsourcing pipeline where workers pair texts with their chart elements, which are collected and merged to generate visual links that are integrated into an interactive document viewer to assist readers identifying relationships between text and charts. Metoyer et al [51] also integrates bidirectional interaction between textual descriptions and charts in the field of sports journalism by extracting entities and actions from sports narratives and linking them to game statistics and charts, allowing readers to move seamlessly between textual play descriptions and visual evidence.

2.1.3 Natural Language and User-Centric Interfaces

Following the concept of assisting, Kim et al. [38] introduces a three-stage pipeline for automatically extracting and linking textual references to their corresponding cells using natural language processing techniques. CrossData [88], uses a text-data connection engine that distinguishes between independent and derived data phrases within text to maintain persistent links with the underlying dataset and

ensure synchronization when textual references or datasets are updated. Gao et al. [20] uses a mixed-initiative natural language interface that surfaces ambiguities in text queries and connects them to multiple possible visual interpretations, which users can refine through interactive widgets. Guo et al. [24] decomposes complex natural questions into sub-questions and linking each fragment of text to corresponding visual answers through annotated charts. Setlur et al. [69, 68] enable natural language interaction with visualizations by interpreting subjective textual modifiers and conversational queries, linking them to corresponding data attributes and visual outputs, and providing interactive widgets to resolve ambiguity, while exposing this interpretation through interactive text so users can see and refine how language has been translated into visual filters. Badam et al. [5] proposes a dynamic linking framework that parses tables to extract structured relationships, identifies meaningful row-column associations, and automatically highlights relevant visualizations when users highlight textual descriptions, thereby enhancing data-driven document reading experiences.

User-centric and authoring-based approaches have also been explored. Charagraph [50] uses a user-centric approach where it dynamically generates visualizations based on the numerical values that the readers manually highlight in a paragraph, allowing for better understanding in part of the users. Xu et al [82] introduces a semi-automated authoring system that extracts statistical features from

Datawrapper charts to generate interactive, editable sentences linked with chart elements, which highlight the related part of the chart. Setlur et al [70] integrates autocompletion with inline data previews so that partially written text queries are immediately connected with suggested visualizations, thereby allowing users to refine queries by directly interacting with the previews and thereby reducing the cognitive effort of bridging text with charts. Latif et al [43] pair automatically generated textual profiles of researchers with inline sparklines and trend charts, collaborations, and topics, linking narrative descriptions of productivity, topics, and collaborations to visual timelines and networks. Mumtaz et al [54] pairs automatically generated textual explanations of code quality with embedded graphics and visualizations, allowing statements about code quality to be cross-checked against visualizations of software metrics.

2.1.4 Applications in Multimedia

Applying text–visualization linking to multimedia, Shen et al. [71] automate the generation of data videos by aligning narration segments with their corresponding visual elements using a combination of large language models and constraint-based animation. VisJockey [41] enables scripted storytelling by allowing authors to preload visualization interactions that are embedded within the text, using an author-driven system where predefined interactive plays, triggered by text, animate

visualizations to align dynamically with narrative flow. Han et al. [25] introduce a framework for documenting remote visualization discussions by capturing referential gestures, speech, and visualization metadata to automatically link transcribed text segments to corresponding visual elements, supporting interactive meeting documentation and enhancing collaborative knowledge retention.

Nevertheless, most of these approaches are based on having access to the structured data such as data tables or metadata encodings[38, 82], reliance on manual processes [40], or depend on traditional NLP pipelines that rely on heuristic rules to detect textual references to data [60]. As such, these approaches are difficult to apply to data-driven reports that incorporate charts without making the underlying data available. Furthermore, the sentence structures found in these reports are often complex, making them less robust when aligning descriptions with their corresponding chart elements using traditional natural language processing methods. In contrast, our work adopts a vision-language model based approach, making them well suited for interpreting visual elements such as charts linking sentences even when the structure of the text is complex and the underlying chart data is not explicitly provided.

Table 2.1: Taxonomy of Text–Visualization Pairing Literature

Category	Representative Works
Automated Approaches	ChartText [60], Kori [45], VizFlow [75], Voder [74], Latif et al. [44]
Annotation and Emphasis-Based Methods	EmphasisChecker [39], Lai et al. [42], ChartAccent [64], Click2Annotate [12], Touch2Annotate [13]
Narrative Storytelling	ChartStory [84], Bryan et al. [8], Hullman et al. [32]
Interactive and Bidirectional Linking	Kong et al. [40], Metoyer et al. [51]
Natural Language Interfaces	Kim et al. [38], CrossData [88], DataTone [20], Talk2Data [24], Eviza [69], Sentifiers [68], Elastic Documents [5]
User-Centric and Authoring Tools	Charagraph [50], Xu et al. [82], Sneak Pique[70], Latif et al. [43], Mumtaz et al. [54]
Applications in Multimedia	Shen et al. [71], VisJockey [41], Han et al. [25]

2.2 Interactive Document Reading

Interactive document reading systems have enabled readers to engage with texts by leveraging various techniques through connecting textual contents with various forms of supplemental information. We have grouped prior research in this space broadly into the categories of (i) *contextual glossing* [27, 18] (ii) citation-mapping and summarization systems for scholarly content [10, 63, 4], (iii) guided reading interfaces and text–visualization linking approaches for integrating descriptions with data representations [40, 38]. The following subsections outline representative work across these categories.

2.2.1 Contextual Glossing

Several approaches have been researched in providing contextual definitions to support readers in understanding domain-specific language. ScholarPhi [27] augments scientific papers by automatically annotating key terms and symbols in order to help readers understand the scientific content without interrupting their reading flow. The system applies symbol recognition and named entity recognition to extract relevant terms, matches them with contextually derived definitions, and presents them interactively through hover-over tooltips and sidebar explanations. Similarly, Qlarify [18] enables interactive engagement with scientific text through embedding popups, contextual links and explanations for implicit references. Readers can interact with the scientific text to automatically receive additional context for implicit or ambiguous references on request, thereby allowing for greater understanding of scientific texts. Living Papers [28] also augments scientific papers by embedding equations with interactive terms definitions, dynamic figures, parameterized simulations and enhanced previews for citations and cross-referenced materials. While these approaches focus on understanding key terminologies of domain specific language through semantic overlays, our work targets sentence-level alignment between descriptive statements and quantitative data, focusing on linking larger textual units to specific chart elements.

2.2.2 Interactive Citation Reading and Summarization

Beyond the realm of contextual glossing, interactive citation reading has been researched to help readers trace references and explore related work of scientific texts. CiteSee [10] introduces a citation-aware reading interface which allows readers to see how cited works are used within and across the scientific paper. It also provides contextual summaries of cited works as well as citation graphs, thereby enhancing reader awareness of how the reference works are used as well as navigating through related works. Similarly, [63] also introduces a citation-aware reading interface that integrates contextual information from follow-on works into the reference paper, thereby allowing readers to switch between the referenced paper and the contextual sentences. Threddy [35] provides an interactive system that enables users to curate and manage research threads by automatically detecting in-text citations in highlighted text and linking them to their referenced papers. Similarly, Synergi [36] enables collaborative literature synthesis by clustering citation networks and generating topic-based summaries using large language models.

Parallel to citation reading, other systems focus on improving document accessibility through skimming and summarization. Scim [19] applies intelligent skimming by categorizing and highlighting essential components of a scientific paper, such as research objectives, methodology and findings, thereby allowing readers to quickly

extract critical insights without reading the entire document. Khan et al. [37] presents an audio skimming app with a visual reading interface to support document skimming in situationally impairing environments. Paper Plain [4] uses natural language summarization techniques to simplify academic content, which makes academic materials easier to understand for non-expert readers. Marvista [11] supports the reader by summarization of articles through three interactive stages. Before reading, users can apply filters based on time constraints or specific questions. During reading, it dims unrelated text and prompts users with AI-generated questions to reflect on their understanding. After reading, it composes a personalized summary by integrating AI-generated content with data about which parts of the article the user spent time on or highlighted. In comparison to these systems that augment document comprehension through summarization or contextual tracing, our approach is to align textual content with chart data and highlight such alignments, thereby allowing greater readability of data-driven reports, particularly for readers with limited data literacy.

2.2.3 Text–Visualization Linking and Guided Reading Interfaces

Research has also been done to support readers’ comprehension of documents through guided attention, collaborative workflows, multimodal annotation and text-visualization integration. To stimulate critical thinking, Peng et al. [58] introduces

a conversational assistant that prompts the reader with section-specific questions when reading a scientific text. Relatedly [56] enhances literature review generation by presenting readers with paragraph-level segments from related work sections across papers, which are ranked dynamically to highlight unread information, thus maintaining user focus on novel content.

Research has also been done on interactive systems in collaborative and multimodal settings. Passages [26] enables users to search, collect, annotate, and organize text segments from multiple documents for sharing across multiple applications, streamlining multi-document exploration and text reuse. Romat et al [65] presents a multimodal system that allows users to annotate documents and visualizations with digital ink to support active reading behaviour. Focusing on accessibility, SciA11y [79] focuses on enhancing accessibility for blind and low vision readers by transforming static scientific PDFs into interactive HTML documents that support keyboard-based navigation, integration of alt-text for figures and tables, thereby offering an inclusive reading experience for users with visual impairments.

Several works have also explored the integration of textual descriptions with tables, charts, and visualizations to reduce cognitive load and improve comprehension. Kong et al. [40] introduced a crowdsourcing pipeline where annotators selected textual references to charts in research papers, news articles, and reports, which were merged and clustered into structured text-visual links that were then

integrated into interactive document viewers. Kim et al. [38] introduced a three-stage pipeline that automatically links textual descriptions with their corresponding table cells and displays the table in a persistent side panel as the reader scrolls through the article, thereby reducing the split-attention effect. Our work extends the approach from tables to linking textual descriptions with a broader range of visualizations, specifically bar, line, and pie charts commonly found in data-driven reports. Similarly, we implement a persistent panel displaying the charts to mitigate the split-attention effect, thereby increasing the readability of the reports, particularly for readers with limited data literacy. Furthermore, we implement the interactive reading interface directly within web browsers through the use of browser extension, enabling dynamic, real-time sentence-to-chart linking across a large scale of content.

Table 2.2: Taxonomy of Interactive Document Reading Literature

Category	Representative Works
Contextual Glossing	ScholarPhi [27], Qlarify [18], Living Papers [28]
Citation-Aware Reading	CiteSee [10], CiteRead [63], Threddy [35], Synergi [36]
Summarization and Skimming Interfaces	Scim [19], Audio Skimming App [37], Paper Plain [4], Marvista [11]
Guided Reading and Attention Support	Crebot [58], Relatedly [56]
Collaborative and Multimodal Interfaces	Passages [26], ActiveInk [65]
Accessibility-Focused Systems	SciA11y [79]
Text-Visualization Linking Interfaces	Kong et al. [40], Kim et al. [38]

2.3 Chart Data Extraction

Extracting data from chart images requires the interpretation of both graphical and textual elements, which includes axes, labels, and visual marks. Early research relied on rule-based and model-driven approaches, often combining edge detection with template matching. Zhou et al. [87] proposed a Hough transform algorithm for detecting bar charts, while Huang et al. [30, 31] developed model-based frameworks for vectorizing chart images. Mishchenko et al. [52] introduced a system that classified chart images using edge features and applied template matching to extract structured data.

Learning-based approaches advanced these pipelines by introducing automated classification stages. ReVision [66] employed an SVM classifier to categorize charts, followed by heuristic rules for extracting elements specific to each chart type. Al-Zaidy et al. [2] applied connected component labeling in Lab color space to detect bar regions, combined with Tesseract OCR for textual labels and a rule-based inference step to map bar heights to data values. Choudhury et al. [14] proposed scalable algorithms for scholarly figure mining, applying Random Forests and Support Vector Machines for detecting figures, classifying chart types, and identifying text roles within figures.

Alongside fully automated systems, semi-automatic tools such as PlotDigitizer

[3] and Auto-Trace [48] allow users to manually define chart elements, after which automated tracing techniques are applied to recover numerical values. In contrast to learning-based pipelines, Sreevalsan-Nair et al. [73] presented a geometry-driven alternative using tensor fields to extract primitives such as bars and scatter points through topological analysis, bypassing the need for training data.

Building on earlier work, later approaches incorporated Convolutional Neural Networks to improve chart classification and feature detection by learning spatial patterns [72, 15, 76]. Poco et al. [61] proposed an end-to-end pipeline for recovering visual encodings from bitmap images, combining visual analysis with convolutional networks for mark classification. Dai et al. [17] proposed an end-to-end pipeline, which generated textual and numerical summaries from extracted chart content using a sequence-to-sequence neural model. Jung et al. [34] combined CNN-based chart classification with user corrections to extend automated extraction to complex chart types such as radar and area charts. Luo et al. [46] integrated CNN-based object detection with rule-based alignment to jointly detect bars, pie segments, lines, axes, and text, and convert them into structured data. Ma et al. [47] proposed a multimodal framework that combines visual features with OCR-extracted text embeddings to improve the alignment of graphical marks and textual labels.

2.4 LLM Agents

2.4.1 LLM Agents in Various Domains

Recent research has demonstrated the applicability of LLM-based agents, both single and multi-agent, across a diverse range of domains. Nakano et al. introduced WebGPT, which augments an LLM with web-browsing capabilities to retrieve information, cite sources, and answer open-ended questions with improved factual grounding. Schick et al. presented Toolformer [67], which trains LLMs to autonomously decide when and how to call external APIs, allowing agents to leverage tools such as calculators, translators, and search engines without additional supervision. In interactive and simulated environments, AutoGPT [77] demonstrated the use of autonomous goal decomposition and iterative execution to complete complex tasks with minimal supervision. Wang et al. introduced Voyager [78], an autonomous agent in Minecraft that continually learns new skills through a dynamic curriculum and a self-refining code library, achieving effective exploration and task-solving efficiency. Park et al. presented Generative Agents [57], which simulated socially coherent characters by combining episodic memory with reflection, enabling long-term consistent behaviors in interactive settings.

In software engineering, Qian et al. developed ChatDev [62], a role-based multi-agent framework where LLM agents assume the responsibilities of product

managers, architects, coders, and testers to collaboratively complete the software development lifecycle. Hong et al. proposed MetaGPT [29], which similarly organizes agents into structured roles to supervise product design and code generation, reducing costs and improving coordination.

Recent research has also introduced general frameworks that extend LLM agents with memory and collaborative capabilities. Modarressi et al. introduced Ret-LLM [53], a memory-augmented framework that enables agents to retrieve structured knowledge for answering complex and time-sensitive questions. Wu et al. introduced AutoGen [81], a multi-agent communication framework in which LLM agents collaborate through scripted conversations to solve tasks requiring reasoning, planning, and tool use.

2.4.2 Agentic Approaches with Charts

Recent work has explored agentic frameworks for automating and enhancing data visualization tasks by enabling LLM-based systems to plan, reason, and iteratively refine visual outputs. PlotGen [23] introduces an agentic system that supports the generation of scientific visualizations through structured query planning, code synthesis, and visual feedback, allowing the system to progressively improve visualization outputs. MatPlotAgent [83] similarly uses an agentic workflow in generating scientific visualizations by decomposing the process into query understanding,

iterative code execution, and visual self-assessments. LightVA [85] introduces a framework with planning, execution, and control agents that recursively decompose high-level analytical goals into analysis and visualization subtasks.

Agentic approaches have also been applied to chart editing and modification tasks. PlotEdit [22] enables editing embedded chart images through a multi-agent framework where agents deconstruct input charts, extract underlying data and code, and modify them based on natural language instructions. ChartCitor [21] introduces an agentic framework designed to improve factual grounding in chart-based question answering systems by providing fine-grained visual citations.

Other agentic frameworks focus on managing and communicating insights during analysis. InsightLens [80] uses agents to automatically extract, organize, and rank insights as users interact with data, providing a structured overview of analytical findings. DataNarrative [33] focuses on storytelling by pairing a generator agent with a critic agent to produce multi-paragraph narratives and accompanying visualizations. To the best of our knowledge, however, no known agentic approaches have been utilized in an interactive document reading environment for pairing textual descriptions with their corresponding data.

3 Methodology

In this chapter, we outline the methodology used to develop and evaluate our framework for linking textual descriptions with visual chart elements in data-driven articles. Our approach is divided into two major stages. The first stage focuses on the design and assessment of a large vision-language model-based framework for sentence-to-data pairing using chart images and textual descriptions. This framework is evaluated using a curated benchmark dataset consisting of charts from Pew Research articles. The second stage involves the implementation of a browser extension that integrates the pairing system into web pages, enabling real-time sentence-to-chart linkage and interactive annotations to support augmented document reading. In the sections that follow, we describe the structure of the proposed framework, the implementation of the browser extension, and the dataset construction and validation process used for evaluation.

3.1 Task Formulation

Given a chart image C embedded in a data-driven article, a structured data table D is first generated through chart decomposition using a large vision-language model. Combined with a textual description of one or more sentences $S = \{s_1, s_2, \dots, s_n\}$, the goal is to produce a mapping that matches each sentence $s_i \in S$ with the relevant entries in D . Following this mapping, the corresponding entries in D are then associated with their visual marks in C . This requires resolving lexical anchors such as category labels or dates, interpreting syntactic or semantic cues such as comparative expressions (“higher than,” “increased from”), and performing arithmetic reasoning when sentences describe derived values rather than directly stated ones.

The expected output is a set of sentence-data pairings, $\Pi = \{(s_i, d_j)\}$, where each s_i is linked to one or more tuples $d_j \in D$, where each tuple $d_j = (a_{j1}, a_{j2}, \dots, a_{jk})$ corresponds to a row of the data table consisting of categorical labels and numerical values. For instance, in a bar chart comparing political affiliation by year, a tuple might take the form $d_j = (\text{“2018”}, \text{“Republicans”}, 45\%)$, where the first two entries represent categorical attributes and the last entry is the numerical value.

To address this task, a large vision–language model–based pipeline is employed

in which the model first analyzes the chart image to infer its underlying structure and generate a structured data table. Using this table alongside the associated textual description, the model then examines each sentence to determine its correspondence with specific data tuples by leveraging lexical cues, syntactic structure, semantic context, and arithmetic reasoning. The framework operates using the base knowledge of large vision–language models, without requiring task-specific fine-tuning.

In addition to the task of pairing textual descriptions with their corresponding data entries, we extend this task into a deployment setting through the development of a browser extension. In this context, the input consists of a full Pew Research article A , represented as a sequence of paragraphs $P = \{S_1, S_2, \dots, S_m\}$ and a set of embedded chart images $\{C_1, C_2, \dots, C_k\}$. The system must first associate each paragraph with the most relevant chart, then apply the sentence-to-data pairing framework to produce mappings. Sentences that are matched to chart data are highlighted within the article, and selecting a highlighted sentence accentuates the corresponding marks in the chart. To further reduce the split-attention effect, the matched chart remains persistently displayed in a pinned sidebar as the user scrolls, ensuring that the visual context is continuously available.

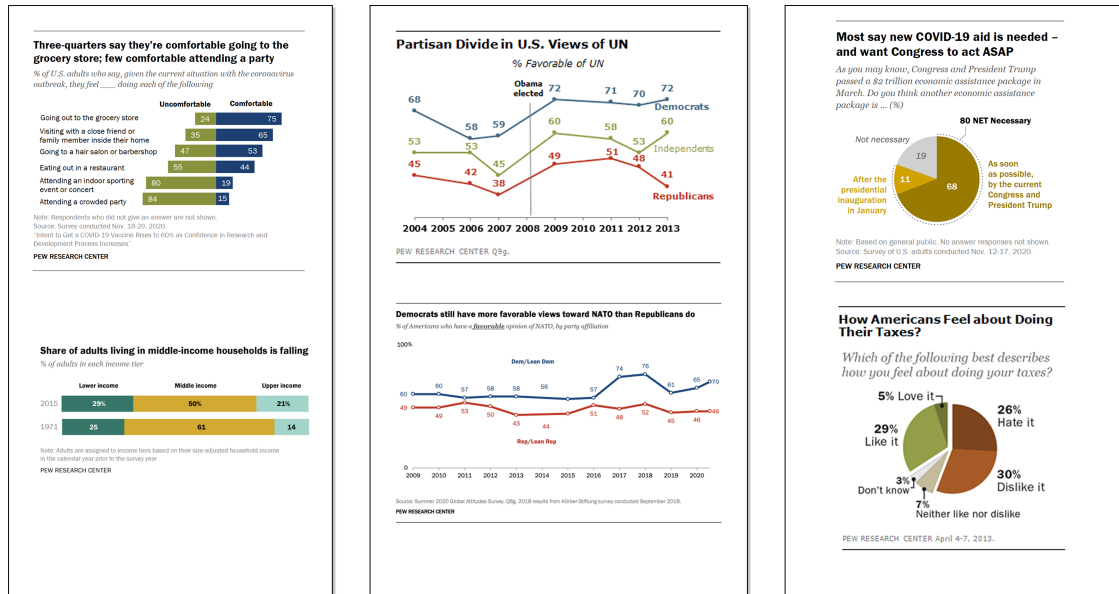
3.2 Dataset Construction

To evaluate our proposed text-visualization pairing frameworks, we curated a benchmark dataset of charts sourced from Pew Research Center. The charts were obtained from the ChartQA dataset [49], which consists of 20,882 charts collected from four different online platforms across a diverse range of topics. From this corpus, we selected 350 charts specifically attributed to Pew Research. The dataset was designed to cover different chart types that are used prevalently, which are primarily bar, line and pie charts. Our final dataset contains 125 bar charts, 125 line charts, and 100 pie charts, maintaining a balanced distribution across chart types.

Chart Type	Chart Count	Total Sentences	Paired Sentences	Chart %	Sentence %
Bar Charts	125	195	195	35.71%	35.78%
Line Charts	125	199	195	35.71%	36.51%
Pie Charts	100	151	151	28.57%	27.71%
Total	350	545	541	100.00%	100.00%

Table 3.1: Distribution of chart types with sentence statistics and percentage breakdowns

Each chart sampled was associated with a Markdown-formatted data table provided by the ChartQA dataset. However, several of these tables contained inconsistencies or omissions such as missing values, incorrect row entries, or mismatches between axis values and visual encodings shown in the corresponding chart images. To ensure the accuracy and reliability of our dataset, we manually reviewed each



A Bar Chart

B Line Chart

C Pie Chart

Figure 3.1: An overview of the chart types included in the dataset: (A) bar charts, (B) line charts, and (C) pie charts, sourced from Pew Research Center articles.

chart and its corresponding table. This involved visually cross-checking the data values encoded in the chart image with those presented in the table and performing corrections where mistakes were identified. This process was necessary, as errors in the data tables would have directly impacted the correctness of sentence-to-data pairing evaluations. In many cases, values were incorrectly recorded, such as swapped values or incomplete phrases, and these were corrected to be better aligned with the data shown in the chart images.

3.2.1 Description Generation

After the verification of the data tables generated from the chart images, we generated textual descriptions for each chart to simulate the captions or summary paragraphs present in data-driven articles. Using metadata from ChartQA, we first extracted question-answer pairs linked to each chart. We then used the GPT-4o model [1] to convert these pairs into declarative sentences. In cases where multiple sentences were generated for a single chart, they were concatenated to form a short descriptive paragraph that reflected trends, comparisons, or data summaries. After generating the textual descriptions from the question-answer pairs, a manual review process was conducted on them. Textual descriptions that primarily contained references to visual features that were not present in the data table, such as color encodings (e.g. “The blue bar”) and spatial orientation (e.g., “the bar on the left”), were either changed or removed. For cases where the descriptions did not reference elements that can be derived from the data table, new descriptions were written that better referenced the corresponding chart data. In addition, all generated sentences were checked for factual correctness, with sentences being changed to better align with the data if they were deemed to be contradictory to the information shown in the chart.

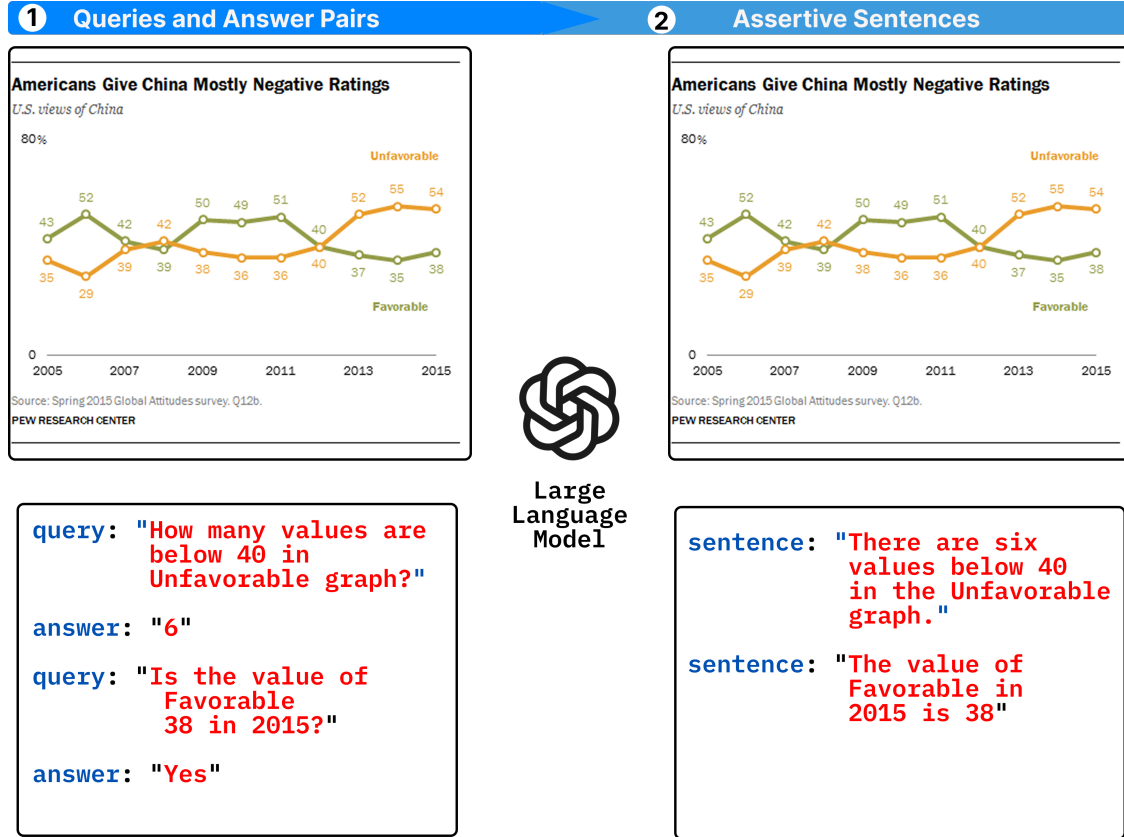


Figure 3.2: Process for generating textual chart descriptions from question-answer pairs. Question-answer pairs from ChartQA are gathered for each chart image (A). Question-answer pairs associated with each chart are then converted into assertive sentences by the large language model (B).

3.2.2 Text-Data Pairing Identification

To evaluate the performance of our sentence-to-data pairing frameworks, we next identified the specific data points referenced by each sentence in our benchmark dataset. Given the large sample size of the curated charts and descriptions, it would be labour-intensive to manually pair the textual descriptions with their correspond-

ing data points. To address this, we used the GPT-4o model to automatically generate initial sentence-to-data pairings by prompting it to identify and match each descriptive sentence to its corresponding data points, based on recognizing trends, comparisons, and reasoning present within the sentences.

However, the accuracy of the generated pairings was influenced by the accuracy of the corresponding data tables. After generating the initial sentence-to-data pairings, we discovered some of the data tables present in the ChartQA dataset contained errors, such as missing or incorrect data, or mismatched column header labels not corresponding to the axes labels present in the chart image. To ensure that the contents of the data table was consistent with the chart, we then compared each data table visually with its corresponding chart image, and any errors in data values or column headers were corrected to ensure consistency.

Once the data tables were verified, we then reviewed the sentence-to-data pairings generated by the model. If sentences were matched to incorrect or unrelated data rows in the table, denoted by tuples, these rows were either removed or corrected. For each revised pairing, cells in the matched tuple that were not explicitly referenced in the sentence were masked with an underscore _ to indicate partial matches with the data row. After this revision process, the final dataset includes 541 validated sentence-to-data pairings across 350 Pew Research charts.

3.2.3 Evaluation Dataset for Chart Object Detection

In addition to the dataset for sentence–data pairing, we constructed a separate evaluation set to assess the accuracy of chart–object detection. Since the ChartQA dataset does not include bounding box annotations, we manually derived ground truth bounding boxes following the style and coordinate format used in ChartOCR [46]. To build this evaluation set, we randomly sampled 25 charts of bar, pie and line charts, resulting in 75 charts in total. Each chart was annotated with bounding boxes corresponding to its constituent marks: bars, pie slices, and line dots. These annotations serve as gold references for evaluating detection systems.

3.3 Chart-Text Agent Pairing Approach

Within the agentic architecture 3.3, the large vision language model pipeline processes the input chart and its corresponding caption in two phases. Given a chart image C , the model first produces a structured data table $D = d_1, d_2, \dots, d_m$ through chart decomposition. Each tuple $d_j = (a_{j1}, a_{j2}, \dots, a_{jk})$ corresponds to a row of categorical attributes and numerical values extracted from C . For each sentence $s_i \in S$, the model takes (D, s_i) as input and outputs a subset of tuples $A_i \subseteq D$ that best correspond to the data referenced in s_i .

- **Chart Deconstruction:** The first stage involves a large vision language

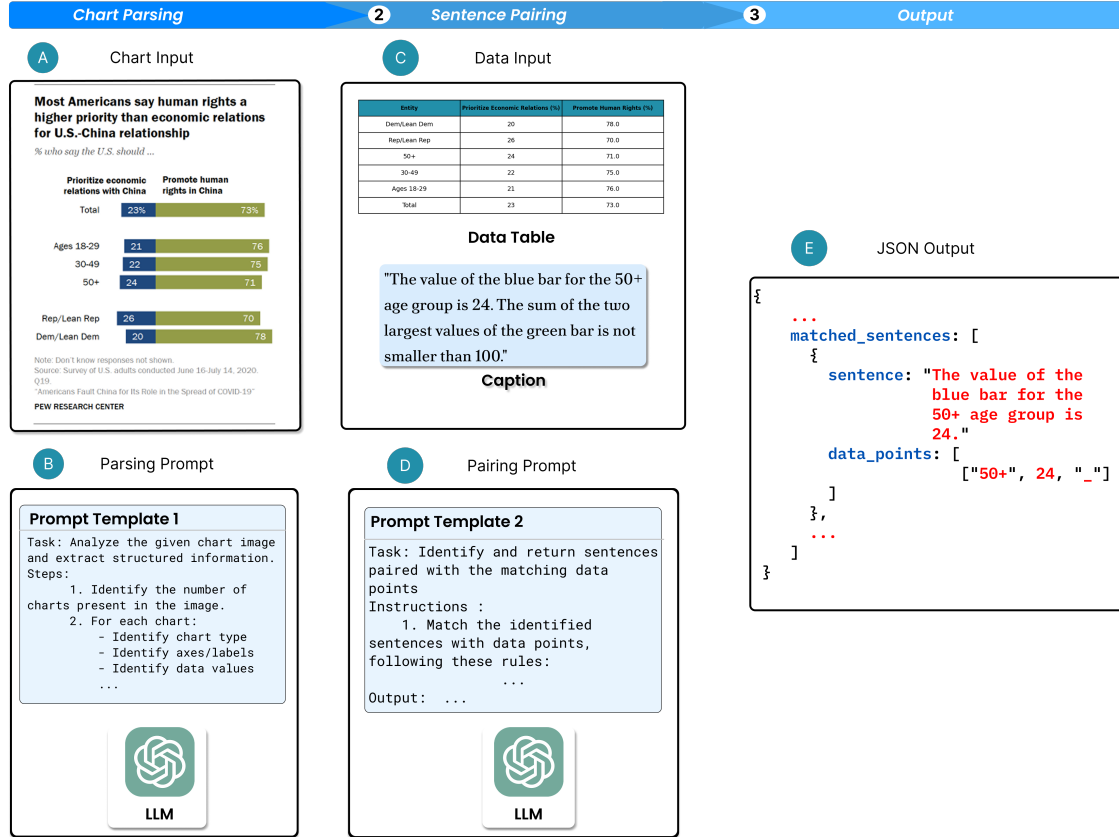


Figure 3.3: An overview of the proposed agentic framework . The chart image in 1 (A) is first processed using an LV-LM (B) that extracts structured chart data in the form of a Markdown-formatted table. The textual description associated with the chart, in addition to the data table (C), is given as an input to a second prompt (D), which identifies matching sentences and aligns them with corresponding rows of the data. The output (E) returns the sentence–data pairings in JSON format, where each sentence is linked to specific data tuples from the table.

model processing a chart image as input. The image is passed as a prompt into the LV-LM, where the model is prompted to identify the type and number of charts present within the image and extract the visual marks present within the image, which consist of axes labels, color encodings and underlying data

points. The extracted information is stored within a JSON object, with the information of the data points and the axes formatted into a markdown table. The outline of this prompting strategy is given in Fig 3.4.

Prompt Type: Chart Breakdown

Input: Chart image .

Process:

- (a) Detect the number of charts N_{CHARTS} in each image.
- (b) For each chart, classify its type $\text{type} \in \{\text{bar}, \text{line}, \text{pie}, \text{other}\}$.
- (c) Extract chart components:
 - Axes and variable labels $\rightarrow$ <axes>
 - Series/marks colors $\rightarrow$ <series_colors>
 - Data values $\rightarrow$ <structured table>
- (d) Construct a Markdown data table where the leftmost column is the independent variable.

Output: JSON object containing the extracted components <output>

Figure 3.4: An outline of the Chart Deconstruction prompt strategy, showcasing chart image as input, processes followed for extracting data, and generated output.

- **Sentence Data Pairing** : The data table generated in the first step will then be entered as an input into another prompt, alongside the textual description of the chart. The model is prompted to then match the sentences of the caption text to one or more relevant data points of the Markdown-formatted ta-

ble. To match the sentences with their corresponding data points, we prompt the model to first isolate terms that are lexically distinct, such as group names, dates, or category labels, that appear in both the caption and the data table, as these unique terminologies refer to specific data entries. The identification of such distinct terms is necessary to reduce ambiguity in identifying which rows of the table are being referenced. We also prompt the model to analyze the syntax and semantics of the sentences to detect comparative expressions, such as “the highest,” “larger than,” or “increased from.” The presences of such phrases gives an indication of a relationship between multiple data points rather than one-to-one references, as shown in the sentence “... *the number of babies born to Muslims is expected to modestly exceed birth to Christians.*” 1.1, indicating the data points when the birth rate of Muslims is greater than that of Christians. This analysis also removes ambiguity that certain words, such as “it”, “them” etc., can have if the sentences of the captions are evaluated in a standalone manner. Outside of lexical, syntactical and semantic analysis, some sentences require inferential reasoning or mental computation to connect them with the data. Sentence such as “*The difference in share of births between Muslims and Christians is 10%.*” 1.1 require the model to perform basic subtraction to identify which two rows satisfy this condition. These cases cannot be resolved through lexical overlap alone and necessitate

logical analysis over the numerical values. The outline of this strategy is given as follows :

Prompt Type: Sentence-Data Pairing

Input:

- (a) Description text <description>
- (b) Markdown data table <table>.

Process:

- (a) **Sentence extraction** — Split <description> by '.','?','!',';'; select sentences with numbers, distinct category terms, or relational semantics relevant to <table>.
- (b) **Filtering** — Discard sentences about non-tabular attributes unless explicitly linked to <table>.
- (c) **Lexical matching** — Align distinct terms (group names, dates, categories) between <caption> and rows of <table>.
- (d) **Syntactic & semantic cues** — Detect relations like "highest", "larger than", "increased from" indicating multi-row comparisons.
- (e) **Logical inference** — Apply simple arithmetic/relations when implied (e.g., differences, ratios, exceeds).
- (f) **Pairing** — For each sentence, map to one or more tuples from <table>. Replace irrelevant tuple elements with _.

Output: JSON array <output>

Figure 3.5: An outline of the prompt strategy followed for pairing textual descriptions with data, showing structured table data and textual description, steps followed for the pairing process, and paired data as output.

3.4 Browser Extension Development

The second stage of this research involves the development of a browser extension that integrates the text–visualization pairing framework into Pew Research articles. The purpose of the extension is to generate sentence–to-chart pairings for embedded figures in real time and to render interactive visual overlays on chart images that link textual descriptions to their corresponding visual marks. These overlays are designed to support augmented document reading by reducing the effort required to locate and interpret chart elements referenced in the text. Given an article A consisting of a sequence of paragraphs $P = \{S_1, S_2, \dots, S_m\}$ and a set of embedded charts $C = \{C_1, C_2, \dots, C_k\}$, the extension produces a mapping $\Phi = \{(S_u, C_v, \Pi_{uv})\}$, where S_u denotes a paragraph, C_v is the chart determined to be most relevant to that paragraph, and $\Pi_{uv} = \{(s_i, d_j)\}$ represents the set of sentence–data pairings linking each sentence $s_i \in S_u$ with one or more tuples d_j from the structured data table D_v extracted from chart C_v . The mapping Φ is then used to highlight sentences in the text and render overlays to emphasize the corresponding marks in the chart image. The overall workflow of the browser extension consists of four main steps: extracting article content, associating paragraphs with charts, (3) pairing sentences with chart data, and generating interactive visual overlays with a persistent chart display.

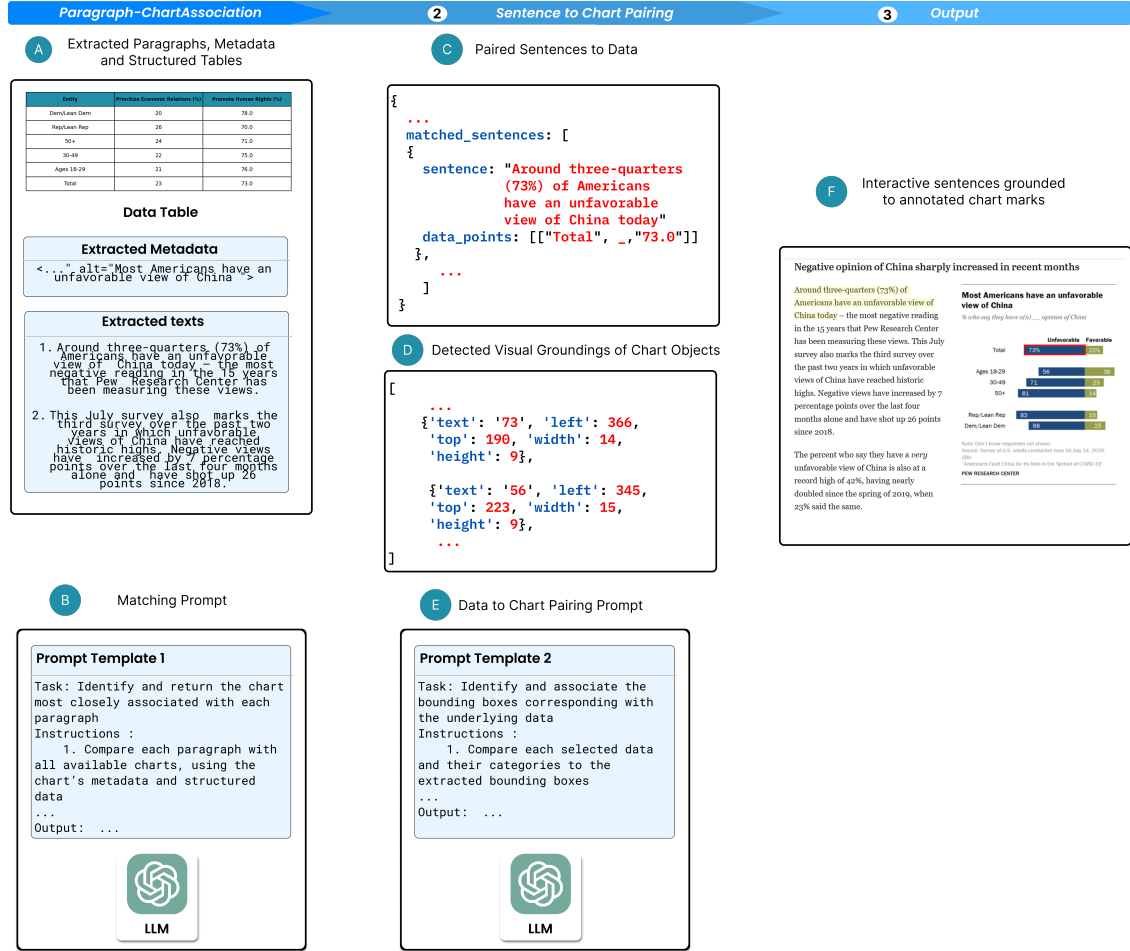


Figure 3.6: Overview of the sentence-to-chart pairing and overlay generation pipeline. Paragraph text, chart metadata, and data tables (A) are used to associate sentences with candidate charts using prompts (B). Sentence–data pairs (C) are grounded to visual chart marks (D) to generate overlay regions that highlight referenced data points (E). The final output (F) presents an augmented reading interface with synchronized text highlights and chart overlays.

3.4.1 Content Extraction

When the extension is activated, it retrieves the full content of a Pew Research article, including both the textual body and the embedded chart images. The article’s HTML is parsed so that only the main content is retained, while navigation elements, banners, and footers are removed. The cleaned text and chart images are then transmitted to the backend server for processing. To support long-form and multi-page reports, the extension identifies and retrieves the HTML contents of linked pages, allowing the entire article to be processed as a single document.

3.4.2 Paragraph–Chart Association

After content extraction, the system associates each paragraph S_u with the chart C_v that it most likely describes. A Pew Research article typically contains multiple paragraphs, which can be regarded as textual descriptions, as well as multiple embedded chart images. Since the framework has no prior knowledge of which description corresponds to which chart, an association must first be established between the textual descriptions and chart images.

Prompt Type: Paragraph–Chart Association

Input:

- `<paragraphs>` = $[S_u]$ — paragraphs from the article ($u = 0, \dots, U - 1$).
- `<charts>` = $[C_v]$ — charts embedded in the article ($v = 0, \dots, V - 1$).
- For each chart C_v :
 - `metadata_v` = $\{\text{title}, \text{caption}, \text{alt_text}\}$.
 - `table_v` — Markdown data table extracted from chart image.

Process:

- (a) **Context Comparison** — Each paragraph S_u is considered against every chart C_v , drawing on the chart’s metadata $\{\text{title}, \text{alt_text}\}$ and structured table `<table>`.
- (b) **Association Estimation** — The relation between S_u and C_v is assessed through a combination of textual correspondence, alignment with table headers and categories, and consistency with referenced values or quantities.
- (c) **Score Derivation** — The degree of association is expressed as a score `score[u,v]`, derived from weighted contributions of the different alignment aspects.
- (d) **Chart Selection** — For each S_u , the chart with the highest score is chosen. If the score exceeds a threshold τ , the association is accepted; otherwise the paragraph is marked with no match.

Output: JSON per paragraph S_u , with the matched chart

Figure 3.7: An outline of the prompt strategy followed for pairing paragraphs in Pew Research article with their corresponding embedded charts.

To achieve this, the textual content of each paragraph is compared against chart-related information, including chart titles, captions, alt-text, and the structured data table extracted from the chart image. The chart with the strongest semantic similarity to the paragraph is selected as the match, ensuring that subsequent sentence–data pairing and overlay generation steps are applied to the appropriate chart.

3.4.3 Sentence–Chart Pairing

Once a chart has been associated with a paragraph, the sentence-to-data pairing framework is applied to align individual sentences with corresponding rows in the chart’s structured data table. The system identifies explicit references such as category labels or dates, interprets comparative expressions such as “larger than” or “increased from,” and performs arithmetic reasoning when sentences describe values that are not explicitly stated in the table but can be derived from it.

Prompt Type: Sentence–Chart Region Pairing

Input:

- **<sentence>** — textual statement referencing part of a chart.
- **<segment regions>** — list of chart segments (bars, pie slices, line points) with their bounding boxes and any linked textual labels.

Process:

- (a) **Reference Identification** — Analyze **<sentence>** to detect which value(s), category label(s), or comparative expressions indicate chart data being referenced.
- (b) **Text Alignment** — Match these references against **<segment regions>** that correspond to the sentence content.
- (c) **Bounding Box Selection** — Select the bounding box coordinates for each matched chart region

Output: An array of bounding boxes aligned with the references in the **<sentence>**

Figure 3.8: An outline of the prompt strategy followed for aligning textual descriptions paired with structured data to their corresponding visual marks in embedded chart images. In this strategy, the visual groundings of chart marks is given to the prompt, which then follows the outlined steps for associating with their corresponding marks

Sentences that are successfully matched with chart data are highlighted within the article text. These highlights act as interaction triggers for rendering visual overlays that emphasize the corresponding chart elements.

3.4.4 Interactive Chart Annotations

To visually link highlighted sentences with their corresponding chart elements, the system generates interactive visual overlays that are rendered directly on top of the chart image. Each chart image C_v is decomposed into its constituent visual marks, such as bars, pie slices, and line points. In addition, textual elements including axis labels and numeric values are detected, and the positions of both visual marks and text elements within the chart image are recorded.

The recorded positions are then used to construct overlay regions that associate each matched data tuple with its corresponding visual element in the chart. During interaction, the overlay regions corresponding to the referenced data are emphasized using bounding boxes, shaded regions, or point-level highlights.

Bar Charts: Bars were initially detected using edge-based methods; however, this approach proved unreliable for pale colors and small bars. To address this, chart images were converted into a color representation that improved separation between adjacent bars. Regions matching the expected rectangular shape and size were retained, while irregular regions were discarded. Each detected bar was paired with the nearest numeric label, enabling the generation of bounding box overlays that accurately highlight referenced bars during interaction.

Pie Charts: Pie chart segments were separated based on color differences after

transforming the image into a representation where slices were visually distinct. Background artifacts were filtered out by removing excessively large or irregular regions. Labels placed inside slices were assigned directly, while external labels were associated based on angular alignment. These associations enabled the creation of slice-level overlay regions that could be highlighted when corresponding sentences were selected.

Line Charts: Line charts required detecting circular point markers representing data values. After masking text regions to avoid false detections, ChartOCR was used to identify line points and their positions. Although initial detections included false positives, these were reduced by excluding points that could not be paired with valid textual anchors. The resulting positions were used to generate point-level overlay highlights, making ChartOCR the preferred method for overlay generation in line charts.

3.4.5 Persistent Chart Display

The final stage of the extension ensures that the chart associated with the currently focused paragraph remains visible in a pinned sidebar as the user scrolls through the article. When a highlighted sentence is selected, the corresponding visual overlays are activated on the chart, explicitly marking the relevant data points. This persistent display, combined with overlay-based highlighting, reduces the need for

readers to switch attention between different regions of the page and helps maintain visual context throughout the reading process.

3.5 Summary

We outlined the methodology for pairing textual descriptions with chart data and visual elements, starting with the formulation of the sentence-to-data pairing task. We then described the construction and validation of a benchmark dataset consisting of 350 charts and 541 sentence–data pairings, which serves as the basis for evaluating the proposed framework. The methodology detailed a large vision–language model–based pipeline designed to extract structured data from chart images and pair it with descriptive sentences using lexical, syntactic, semantic, and arithmetic reasoning. We then extended this framework into a practical setting through the development of a browser extension that integrates sentence-to-chart linking directly within Pew Research articles, enabling persistent chart display and interactive visual overlays.

4 Evaluation and Results

This chapter presents the evaluation methodology, experimental setup, and results of the proposed framework for linking textual descriptions with chart data and visual elements. We first define the criteria for measuring the effectiveness of the sentence-to-data pairing framework and compare our approach across multiple large vision-language models. The performance is assessed against a benchmark dataset of sentence-to-data pairs generated from charts sourced from Pew Research articles, where each chart is associated with a Markdown-formatted data table and a paragraph of descriptive sentences. In addition, we further evaluate against the dataset by Kong et al. [40], which consists of document-chart pairs that were manually generated via crowdsourcing. We assess the results using a structured classification scheme and log hallucination errors to analyze model reliability.

Building on these metrics, we then describe the experimental results obtained by applying the frameworks in practice. The first stage of experiments focuses on pairing sentences with chart data extracted from chart images, where each predicted

pairing is compared against manually annotated gold-standard tuples. The second stage extends this evaluation into an interactive reading environment, where the frameworks are integrated into a browser extension that aligns textual descriptions with annotated chart elements in Pew Research articles. We then present the results for both stages, showing the accuracy of sentence-to-data pairing across different chart types and models, and accuracy of visual object annotation in charts corresponding to paired sentences.

4.1 Reference Extraction Evaluation Criteria

To evaluate the effectiveness of our proposed sentence-to-data pairing frameworks, we conduct experiments on a benchmark dataset constructed from Pew Research charts. The dataset consists of 350 charts in total, including 125 bar charts, 125 line charts, and 100 pie charts. Each chart is paired with a Markdown-formatted data table and an accompanying paragraph of descriptive text. Every sentence in the paragraph is manually annotated with one or more corresponding data tuples, which serve as gold-standard references for evaluation.

In addition to this primary dataset, we further evaluate the proposed pipeline on a document-chart dataset by Kong et al. [40], which provides manually crowd-sourced annotated paragraph-chart pairs for evaluating text-chart pairings.

For the primary dataset, each chart–description pair is processed using the pro-

posed sentence-to-data pairing framework to generate predicted mappings between sentences and chart data rows. Specifically, sentences in the textual descriptions are mapped to data tuples, where each tuple corresponds to a row in the associated Markdown-formatted data table. To account for partial matches, cells within a data row that are not referenced by a sentence are masked using an underscore (`_`). Each sentence is then evaluated based on the degree of alignment between the predicted data tuples and the gold-standard annotations.

Formally, let S denote a sentence, G the set of gold-standard annotated tuples for S , and A the set of tuples predicted by the model. Each prediction is classified into one of the following categories:

- **Correct Reference:** The predicted data points match the gold-standard annotations exactly. That is, for a given sentence S , the predicted tuple set A is equal to the gold tuple set G (i.e., $A = G$).
- **False Negative:** The prediction omits some data points that should have been included. That is, $A \subset G$, meaning the model only captures a subset of the correct tuples.
- **False Positive:** The prediction includes additional tuples that are not part of the gold-standard set. That is, $G \subset A$, indicating over-prediction.
- **False Positive + False Negative:** The prediction contains both missing

and extraneous tuples. That is, $A \not\subseteq G$ and $G \not\subseteq A$, meaning there is a partial but incorrect overlap between predicted and gold tuples.

4.2 Reference Extraction Evaluation Process

Direct comparison between predicted tuples and gold-standard tuples presents several challenges due to the way large vision-language models (LVLMs) generate tabular representations of chart data. During preliminary testing, it became clear that variations in column order and header naming, particularly in multi-variable charts such as stacked bar or multi-line charts, complicate alignment. LVLM outputs sometimes introduced headers that were not present in the chart image, or replaced missing categories with generic placeholders like Entity, Value, or Percentage. Without accounting for these inconsistencies, tuple-level comparisons would yield unreliable evaluations of model accuracy.

To mitigate these issues, a tuple matching strategy was designed around dynamic column alignment. First, both predicted and gold-standard headers are normalized by converting them to lowercase and stripping units. Semantic similarity is then computed using cosine similarity between token embeddings of the headers, allowing each predicted column to be mapped to the gold-standard column with the highest similarity score, provided it exceeds a defined threshold. This ensures that minor lexical or structural differences do not prevent accurate alignment of

tuples across the two tables.

A refinement is introduced for charts that naturally produce two-column tables, such as pie charts or single-series line charts. In such cases, generic placeholder headers are directly mapped, with the label "Entity" corresponding to the categorical column and Value to the numerical column, bypassing the semantic similarity step. For charts with more than two columns, however, placeholder mapping is avoided, as it risks ambiguous alignments. For tables with more than two columns, semantic similarity alignment remains the primary method.

Once column mappings are established, predicted tuples are reordered to match the schema of the gold-standard tuples. Final evaluation then compares the values of aligned tuples and classifies each prediction into one of four categories: True Positive (TP), False Negative (FN), False Positive (FP), or False Positive + False Negative (FP+FN)

4.2.1 Prompt Strategy Evaluation

To examine the effectiveness of different prompt strategies on the process of sentence-data pairing, we compare two prompting strategies: a zero-shot approach and a few-shot approach, using the GPT-4o model. The primary focus of this analysis is to identify the effect of different prompt strategies is identifying and associating data tuples with their corresponding sentences correctly.

The evaluation is conducted on the dataset introduced by Kong et al. [40], consisting of paragraph-chart pairs, with crowdsourced references linking textual sentences to specific visual marks and their associated data.

For the zero-shot prompting strategy, the sentence-data pairing prompt consisted of only the task instructions on how to identify the relevant data tuples associated with the sentences present in the textual description, with no explicit examples being provided. For the few-shot strategy, several working examples were provided alongside the task instructions to illustrate correct sentence-tuple mappings. We assess the outputs from both prompt strategies using the same four accuracy categories: True Positive (TP), False Negative (FN), False Positive (FP), and False Positive + False Negative (FP+FN).

4.2.2 Model Comparison

The evaluation is conducted across multiple large vision-language models (LVLMs) to compare their effectiveness in pairing textual descriptions with chart data. Each large vision-language model is evaluated using the same agentic framework, using the same benchmark dataset of Pew Research charts, with predicted outputs compared against the manually verified gold-standard annotations following the evaluation procedure described above. For this study, we employ three state-of-the-art LVLMs : GPT-4o, Claude 3.7 Sonnet, and Gemini 2.5 Pro. These models

are included for their strong performance across multimodal reasoning, vision understanding, and contextual reasoning capabilities [1, 16]. In addition to these models, we also evaluate on the smaller scale versions of two leading open-source LVLMs: Qwen3-VL-8B-Instruct and InternVL3-8B-Instruct.

4.2.3 Error Analysis

In addition to evaluating the accuracy of our frameworks to identify the corresponding data points of a textual description, we also track instances of hallucinated content generated by the vision large language model. In the context of the sentence-to-data pairing task, the hallucinated content tend to be output that do not correspond to the data in the chart image, incoherent content generated from the textual description given as an input or inconsistent syntax that do violate the format designated by the prompting strategies. To conduct this analysis, we randomly sample sentence-to-data pairings generated from approximately 100 chart images and their corresponding textual descriptions. Each sampled output is then manually reviewed and classified into different types of hallucination categories, some of which are as follows :

- **Factual Contradiction:** The model links a sentence to data that directly contradicts the values shown in the chart.

- **Entity Error:** The model introduces or misidentifies group names, axis labels, or categories that are either incorrect or absent from the chart.
- **Unverifiable Reference:** The output includes references to values, categories, or ranges that are not present in the chart or data table.
- **Overclaim:** The model assigns a sentence to more data rows than appropriate, going beyond what is semantically supported.
- **Logical Inconsistency:** The model demonstrates flawed reasoning, such as performing inaccurate computations or making incorrect trend inferences.
- **Instruction Violation:** The output does not follow the structure required by the prompt, such as deviating from the expected format.

The hallucinations are logged and analyzed separately from the structural pairing metrics and are useful for evaluating the robustness and reliability of each framework and large vision language model.

4.3 Web Browser Extension Evaluation

The second part of the evaluation focuses on assessing the browser extension that integrates sentence-to-data linking directly into Pew Research articles. In contrast to the evaluation of sentence-to-data alignment by the LVLM frameworks,

this stage evaluates the effectiveness of identifying and aligning the visual marks of chart images with their corresponding data entries in structured data tables generated from chart images. Specifically, we evaluate our proposed methodologies of detecting visual marks, such as bars, pie segments and line dots, with the output from ChartOCR, which similarly works across the corresponding chart types of our interest. After evaluating the different methodologies of detecting and anchoring chart marks, we then compare the accuracy of pairing textual descriptions to their appropriate chart regions.

4.3.0.1 Chart Mark Detection

This stage evaluates the identification of chart elements such as bars, line points, and pie segments by comparing our proposed detection methodology with the baseline ChartOCR pipeline. Let $B_p = \{b_{p1}, b_{p2}, \dots, b_{pm}\}$ represent the set of predicted bounding boxes for a chart and $B_g = \{b_{g1}, b_{g2}, \dots, b_{gn}\}$ the corresponding gold-standard set. The Intersection-over-Union (IoU) between a predicted box b_{pi} and a ground-truth box b_{gj} is defined as

$$IoU(b_{pi}, b_{gj}) = \frac{|b_{pi} \cap b_{gj}|}{|b_{pi} \cup b_{gj}|}.$$

A prediction is regarded as correct if $IoU(b_{pi}, b_{gj}) \geq 0.5$ for a matched pair

(b_{pi}, b_{gj}) . To measure detection performance across the dataset, we employ Average Precision at $\text{IoU} \geq 0.5$ (AP@0.5), which aggregates precision and recall by considering all predicted-gold pairs above this threshold as correct detections. AP@0.5 has been widely used in object detection as it captures both the sensitivity of the detector and the accuracy of localization. For chart mark detection, this metric provides a reasonable balance between penalizing missed or redundant boxes and allowing for minor annotation variability.

4.4 Experiments and Results

We first report the results of text–chart reference extraction, followed by an evaluation of the web browser extension.

4.4.1 Reference Extraction Evaluation

We design our experiments to answer the following research questions:

- **RQ1:** How do different vision–language models compare in sentence–chart pairing performance under the proposed reference extraction pipeline?
- **RQ2:** How do prompting strategies (zero-shot vs. few-shot) affect sentence–chart pairing accuracy?

4.4.1.1 Model Comparisons

To address **RQ1**, we evaluate a range of state-of-the-art proprietary and open-source vision–language models, including GPT-4o, Claude-3.7-Sonnet, and Gemini-2.5-Pro, as well as smaller open-source models Qwen3-VL-8B-Instruct and InternVL3-8B-Instruct.

Evaluation is conducted on two datasets. The primary dataset consists of 350 chart images from the Pew Research website, including 125 bar charts, 125 line charts, and 100 pie charts. In addition, we evaluate the pipeline on the document–chart dataset introduced by Kong et al. [40], which contains crowdsourced annotations originally designed for studying text–chart reference extraction.

Table 4.2 reports sentence–data pairing accuracy across all four evaluation categories. Results are shown under two experimental settings: (i) the *full pipeline*, where chart images are first converted into structured tables before sentence–data pairing, and (ii) the *oracle table* setting, where models directly operate on manually provided gold-standard tables. This comparison isolates errors introduced by chart-to-table conversion from those arising during sentence–data pairing.

4.4.1.2 Prompt Strategies

To address **RQ2**, we examine the impact of prompting strategies on sentence–data pairing performance. We use GPT 4o for this comparison. Table 4.1 summarizes results obtained using zero-shot and few-shot prompting.

Under the zero-shot setting, the model achieves a true positive (TP) rate of 31.40%, with corresponding false positive (FP), false negative (FN), and combined FP+FN rates of 3.49%, 26.74%, and 38.37%, respectively. These results highlight the difficulty of accurately identifying sentence–data alignments without explicit guidance.

In contrast, few-shot prompting leads to a substantial performance improvement. The TP rate increases to 57.8%, indicating a markedly stronger ability to identify correct sentence–data associations when examples are provided. The FP rate remains low at 3.6%, while the FN rate is comparable to the zero-shot condition at 26.5%. Notably, the combined FP+FN category decreases sharply to 12.0%, reflecting fewer partially correct but misaligned predictions.

Overall, these findings demonstrate that few-shot prompting provides a significantly more reliable basis for sentence–data pairing. The increase in TP rate and reduction in severe error cases suggest that example-driven prompting helps guide the model toward more consistent and accurate reference extraction.

Model	TP			FP			FN			FP+FN		
	Bar	Line	Pie	Bar	Line	Pie	Bar	Line	Pie	Bar	Line	Pie
<i>Pew Research Dataset (350 Charts)</i>												
<i>Full Pipeline (Chart Extraction + Sentence-Data Pairing)</i>												
GPT-4o	56.28	42.79	49.34	2.01	4.65	1.32	19.10	30.70	32.24	22.61	21.86	17.11
Claude-3.7	54.30	53.71	68.83	0.90	3.06	1.30	6.79	12.66	7.14	38.01	30.57	22.73
Gemini-2.5 Pro	59.62	54.19	58.28	1.41	3.08	1.99	10.33	20.26	13.25	28.64	22.47	26.49
Qwen3-VL-8B-Instruct	44.10	37.72	39.87	3.60	5.26	4.58	7.21	11.84	13.07	45.05	45.16	42.48
InternVL3-8B-Instruct	1.53	4.42	8.46	0.00	0.63	0.00	62.98	53.63	58.70	35.50	41.32	32.84
<i>Pairing Only (Using Fixed Gold Data Tables)</i>												
GPT-4o	61.10	49.80	55.30	2.02	2.30	1.32	17.70	27.20	32.90	19.20	20.74	10.53
Claude-3.7	64.55	62.30	74.50	2.27	0.44	3.27	4.09	9.21	7.84	29.09	28.07	14.38
Gemini-2.5 Pro	68.40	60.00	74.34	1.42	1.78	3.95	8.96	17.30	11.20	21.20	20.90	10.53
Qwen3-VL-8B-Instruct	44.84	37.39	65.58	2.24	3.48	7.79	5.83	11.30	10.39	47.09	47.83	16.23
InternVL3-8B-Instruct	4.55	3.37	13.43	0.00	0.31	1.49	48.05	49.38	50.75	47.40	46.93	34.33
<i>Kong et al. Dataset (Pairing Only, Aggregated)</i>												
GPT-4o	57.8	–	–	3.6	–	–	26.5	–	–	12.0	–	–
Claude-3.7	38.3	–	–	5.32	–	–	27.66	–	–	28.72	–	–
Gemini-2.5 Pro	65.85	–	–	3.66	–	–	13.41	–	–	17.07	–	–
Qwen3-VL-8B-Instruct	29.11	–	–	3.8	–	–	43.04	–	–	24.05	–	–
InternVL3-8B-Instruct	0.78	–	–	0.00	–	–	44.96	–	–	54.26	–	–
<i>Kong et al. Dataset (Full Pipeline, Aggregated)</i>												
GPT-4o	39.51	–	–	1.23	–	–	22.22	–	–	37.04	–	–
Claude-3.7	28.57	–	–	3.30	–	–	28.57	–	–	39.56	–	–
Gemini-2.5 Pro	40.48	–	–	1.19	–	–	21.43	–	–	36.90	–	–
Qwen3-VL-8B-Instruct	22.22	–	–	2.47	–	–	40.74	–	–	34.57	–	–
InternVL3-8B-Instruct	0.00	–	–	0.00	–	–	100.00	–	–	0.00	–	–

Table 4.2: Sentence-data pairing accuracy across models on Pew Research charts (bar, line, pie) and the Kong et al. dataset under two evaluation settings.

Table 4.1: Comparison of pairing accuracy results across prompting strategies

Strategy	TP (%)	FP (%)	FN (%)	FP + FN (%)
Zero-shot	31.40	3.49	26.74	38.37
Few-shot	57.8	3.6	26.5	12.0

Across all models, performance is consistently higher in the sentence-pairing-only setting than in the full pipeline, indicating that errors introduced during the extraction of structured data from chart images propagate into the pairing stage. This difference in performance is observed across all chart types, although the mag-

Model	F1 Score
<i>Kong et al. Dataset (Pairing Only, Aggregated)</i>	
GPT-4o	0.793
Claude-3.7	0.699
Gemini-2.5 Pro	0.885
Qwen3-VL-8B-Instruct	0.554
InternVL3-8B-Instruct	0.034
<i>Kong et al. Dataset (Full Pipeline, Aggregated)</i>	
GPT-4o	0.771
Claude-3.7	0.642
Gemini-2.5 Pro	0.782
Qwen3-VL-8B-Instruct	0.507
InternVL3-8B-Instruct	0.000

Table 4.3: F1-scores for sentence-data pairing on the Kong et al. dataset under pairing-only and full-pipeline evaluation settings.

nitude of the difference varies across the different charts and large vision language models. The three closed sourced models, GPT-4o, Claude-3.7-Sonnet, and Gemini-2.5-Pro, show comparatively higher positive rates in comparison to the open-source models, particularly when evaluated in the end-to-end condition.

Across the different chart types, line charts exhibit lower pairing performance relative to bar and pie charts. While bar and pie charts maintain comparatively higher accuracy, performance on line charts degrades more sharply when using extracted tables. This trend suggests that the characteristics of line charts, such as overlapping series and closely spaced data points, make them more susceptible to errors when structured data is generated from the chart images, which in turn negatively affect sentence–data alignment.

False positive rates remain low across all models and chart types, indicating

that incorrect pairings are relatively rare. Instead, performance degradation in the full pipeline setting is primarily driven by increases in false negatives, reflecting cases where correct sentence–data associations are missed due to incomplete or inaccurate table extraction.

Similar to the results generated for the curated dataset, results for the dataset of document-chart pairs exhibited a similar overall pattern. When pairing is performed using verified tabular data, models achieve substantially higher true positive rates. However, in case of full pipeline setting, where structured tables are extracted from the chart images, true positive rates decrease, accompanied by a corresponding increase in false negatives and combined false-positive–false-negative errors. This trend is pronounced for open-source models, which exhibit substantially lower true positive rates under the full pipeline setting.

In the original study by Kong et al [40], sentence-level matching performance was reported using a distance measure defined as $1 - F1$. with their clustering-and-merging pipeline measuring this distance to 0.22, corresponding to an approximate F1-score of 0.78 after aggregation. For our evaluation on their dataset, F1-scores were computed directly from the reported true positive (TP), false positive (FP), and false negative (FN) percentages using the standard harmonic mean formulation:

$$F1 = \frac{2TP}{2TP + FP + FN}$$

This formulation is derived from precision ($\frac{TP}{TP+FP}$) and recall ($\frac{TP}{TP+FN}$), and enables a direct comparison with the sentence-level F1 implied by the distance metric.

Under the pairing-only setting, closed-source models consistently outperform the open-source alternatives. In particular, Gemini-2.5 Pro and GPT-4o exceed the aggregated sentence-level performance reported by Kong et al., whereas, open-source models remain substantially below the crowdsourced benchmark.

When evaluated under the full pipeline setting, performance declines across all models, reflecting the impact of errors introduced during structured data extraction from chart images. Gemini-2.5 Pro marginally surpasses the clustered crowdsourcing benchmark reported by Kong et al., while GPT-4o remains closely comparable but slightly below it. In contrast, open-source models fall substantially below the original crowdsourced pipeline performance.

Overall, these results indicate that strong closed-source LVLMs are capable of matching or surpassing the aggregated crowdsourcing performance reported by Kong et al. in controlled pairing settings. However, when structured data must first be extracted from chart images in the full pipeline setting, performance declines across all models, indicating that upstream extraction errors negatively impact sentence-data alignment. While the top-performing models remain competitive with the crowdsourced benchmark, the degradation observed in the end-to-end

condition highlights the sensitivity of automated systems to earlier pipeline stages.

4.4.1.3 Error Analysis

Other than the accuracy of the pairings, we also observe the rate of errors generated by each of the models. Table 4.4 shows the count of malformed JSON structures produced across each chart type by the different LVLMS, with the table showing that the Claude model generally demonstrated greater structural consistency in comparison to the OpenAI model, generating zero unparsable files for the pie charts.

Table 4.4: Count and percentage of unparsable files generated by each model, per chart type.

Model	Pie	Line	Bar
GPT-4o	14 (14.0%)	24 (19.2%)	12 (9.6%)
Claude 3.7 Sonnet	0 (0.0%)	27 (21.6%)	9 (7.2%)

4.4.2 Web-browser extension

Following the methodology of evaluating the sentence-to-chart pairing pipeline of the web-browser extension, we first assess the accuracy of the chart mark detection, after which we evaluate the accuracy of mapping sentences to detected chart regions.

For chart-object detection, we construct two evaluation sets. The primary evaluation set consists of 25 bar charts, 25 line charts, and 25 pie charts sampled from the curated Pew Research dataset, each annotated with gold-standard bounded corresponding to chart marks. In addition, chart-object detection is evaluated on

the Kong et al. dataset, consisting of document–chart pairs with their chart marks manually annotated through a crowdsourcing pipeline.

4.4.2.1 Chart-Object Detection

We compare our detection procedures against the ChartOCR pipeline as a baseline.

Let B_p and B_g denote the predicted and gold sets of boxes, respectively. Detections are considered correct if $\text{IoU}(b_{pi}, b_{gj}) \geq 0.5$.

Table 4.5: Mark detection results across charts in Kong et al. comparing Proposed Method vs. ChartOCR Baseline

Method	Chart Type	Total	Detected	Accuracy (%)
Proposed	Horizontal	575	364	63.30
	Vertical	197	84	42.69
	Overall	772	448	58.06
ChartOCR	Horizontal	575	15	2.61
	Vertical	197	178	90.36
	Overall	772	193	25.00

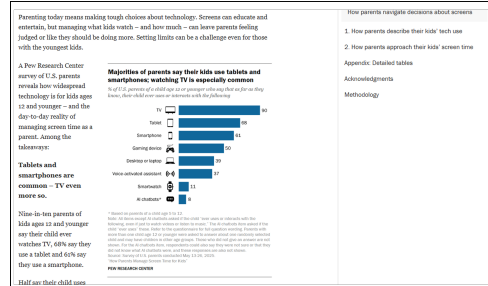
In Table 4.5, we report the chart-object detection accuracy of the proposed method in comparison with the ChartOCR baseline on the Kong et al.[40] dataset. Results are reported separately for horizontal and vertical bar charts, as well as in an aggregated form across all bar charts.

Overall, our method achieves higher detection accuracy than ChartOCR [46] when results are aggregated across the different types of bar charts. While both methods exhibit variation in performance across chart orientations, our method

maintains more consistent detection behavior across horizontal and vertical bar charts.

4.5 Summary

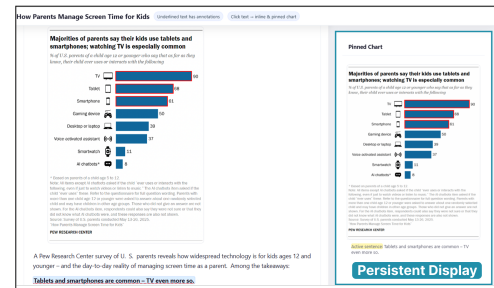
We presented the evaluation methodology, experimental setup, and results for assessing the proposed sentence-to-data pairing frameworks and detecting chart objects. Evaluation was conducted on a benchmark dataset of Pew Research charts alongside a dataset of document-chart pairs by Kong et al. [40]. The results show that pairing performance is consistently higher when models operate directly on manually verified gold tables than when using the full pipeline, indicating that errors introduced during chart-to-table extraction propagate into the sentence-pairing stage. This performance gap is observed across all chart types, though its magnitude varies by chart structure and model. Among the evaluated large vision-language models, closed-source models demonstrate higher pairing accuracy compared to smaller open-source models. For chart-object detection, our method achieves higher overall detection accuracy than the ChartOCR baseline across the different types of bar charts.



A Pew Research Report



B Report Extracted by Browser Extension



C Interactive Reading of Pew Report

Across chart orientations, detection accuracy is higher for horizontal bar charts than for vertical bar charts under our method/In contrast, ChartOCR exhibits highly uneven performance across orientations, achieving strong performance on vertical bar charts but substantially lower accuracy on horizontal bar charts. These results indicate that the proposed detection approach is less sensitive to chart orientation compared to the baseline, resulting in a robust performance.

Figure 4.1: Illustration of the browser extension's deconstruction of a Pew Research webpage, where the original article view (A) is parsed to separate text from embedded charts and reorganized into a processed layout (B) with sentences linked to corresponding visual elements. Selecting a linked sentence highlights the relevant chart marks and renders the full chart in a persistent display panel (C), keeping it visible during scrolling.

5 Conclusions and Future Work

5.1 Conclusions

In this thesis, we examined the problem of connecting textual descriptions with their corresponding visualizations in data-rich articles, focusing on Pew Research reports. To this end, we evaluated large vision–language models (LVLMs) on their ability to align sentences with structured chart data by proposing a single-agent pipeline and a multi-agent architecture that automatically link descriptive sentences to rows in structured data tables extracted from chart images and introduced a browser extension that applies these alignments directly to Pew articles through interactive annotations.

This thesis makes two primary contributions. First, we constructed a gold-standard dataset of sentence–data pairs across bar, line, and pie charts, enabling systematic evaluation of LVLM performance; second, we translated these findings into practice by developing a browser extension that enhances article readability through sentence-linked chart annotations and persistent chart displays. Together,

these contributions demonstrate the feasibility of automated text–visualization linking and its potential to improve the user experience of data-rich documents.

5.2 Future Work

Several directions remain for future exploration:

- (i) Expanding the dataset to include additional chart types such as scatterplots, area charts, and bubble charts would allow for a more comprehensive evaluation of alignment methods across a wider variety of visualization formats.
- (ii) Evaluating the performance of the proposed framework using additional evaluation metrics would provide a more comprehensive assessment of alignment quality by enabling systematic examination of error patterns, misalignments, and the overall reliability and generalizability of the sentence–data pairing process.
- (iii) Applying the framework to other sources of data-rich documents, such as scientific publications, educational materials etc. would test its adaptability and extend its usefulness to a broader range of communication contexts.
- (iv) Conducting user studies to evaluate the browser extension in practice is necessary to measure its effectiveness. Controlled experiments assessing comprehension, cognitive load, and user satisfaction would provide evidence of the

real-world benefits of interactive text–visualization linking and guide refinements of the system.

- (*v*) Developing fine-tuned models specifically for sentence-to-chart alignment could improve the accuracy and reliability of the pairing process beyond what is achievable with general-purpose LVLMs.

These directions provide a foundation for advancing automated text–visualization integration toward more robust, generalizable, and user-centered systems.

Bibliography

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Rabah A Al-Zaidy and C Lee Giles. Automatic extraction of data from bar charts. In *Proceedings of the 8th international conference on knowledge capture*, pages 1–4, 2015.
- [3] Ankit Rohatgi. PlotDigitizer. <https://plotdigitizer.com/>, 2015. Accessed: 2025-04-28.
- [4] Tal August, Lucy Lu Wang, Jonathan Bragg, Marti A Hearst, Andrew Head, and Kyle Lo. Paper plain: Making medical research papers approachable to healthcare consumers with natural language processing. *ACM Transactions on Computer-Human Interaction*, 30(5):1–38, 2023.
- [5] Sriram Karthik Badam, Zhicheng Liu, and Niklas Elmqvist. Elastic documents: Coupling text and tables through contextual visualizations for enhanced document reading. *IEEE transactions on visualization and computer graphics*, 25(1):661–671, 2018.
- [6] Katy Börner, Andreas Bueckle, and Michael Ginda. Data visualization literacy: Definitions, conceptual frameworks, exercises, and assessments. *Proceedings of the National Academy of Sciences*, 116(6):1857–1864, 2019.
- [7] Jeremy Boy, Ronald A Rensink, Enrico Bertini, and Jean-Daniel Fekete. A principled way of assessing visualization literacy. *IEEE transactions on visualization and computer graphics*, 20(12):1963–1972, 2014.
- [8] Chris Bryan, Kwan-Liu Ma, and Jonathan Woodring. Temporal summary images: An approach to narrative visualization via interactive annotation generation and placement. *IEEE transactions on visualization and computer graphics*, 23(1):511–520, 2016.

- [9] Paul Chandler and John Sweller. The split-attention effect as a factor in the design of instruction. *British Journal of Educational Psychology*, 62(2):233–246, 1992.
- [10] Joseph Chee Chang, Amy X Zhang, Jonathan Bragg, Andrew Head, Kyle Lo, Doug Downey, and Daniel S Weld. Citesee: Augmenting citations in scientific papers with persistent and personalized historical context. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2023.
- [11] Xiang “Anthony” Chen, Chien-Sheng Wu, Lidiya Murakhovska, Philippe Laban, Tong Niu, Wenhao Liu, and Caiming Xiong. Marvista: exploring the design of a human-ai collaborative news reading tool. *ACM Transactions on Computer-Human Interaction*, 30(6):1–27, 2023.
- [12] Yang Chen, Scott Barlowe, and Jing Yang. Click2annotate: Automated insight externalization with rich semantics. In *2010 IEEE symposium on visual analytics science and technology*, pages 155–162. IEEE, 2010.
- [13] Yang Chen, Jing Yang, Scott Barlowe, and Dong H Jeong. Touch2annotate: Generating better annotations with less human effort on multi-touch interfaces. In *CHI’10 Extended Abstracts on Human Factors in Computing Systems*, pages 3703–3708. 2010.
- [14] Sagnik Ray Choudhury, Shuting Wang, and C Lee Giles. Scalable algorithms for scholarly figure mining and semantics. In *Proceedings of the International Workshop on Semantic Big Data*, pages 1–6, 2016.
- [15] Mathieu Cliche, David Rosenberg, Dhruv Madeka, and Connie Yee. Scatter-act: Automated extraction of data from scatter plots. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10*, pages 135–150. Springer, 2017.
- [16] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.
- [17] Wenjing Dai, Meng Wang, Zhibin Niu, and Jiawan Zhang. Chart decoder: Generating textual and numeric information from chart images automatically. *Journal of Visual Languages & Computing*, 48:101–109, 2018.

- [18] Raymond Fok, Joseph Chee Chang, Tal August, Amy X Zhang, and Daniel S Weld. Qlarify: Bridging scholarly abstracts and papers with recursively expandable summaries. *arXiv preprint arXiv:2310.07581*, 6(3), 2023.
- [19] Raymond Fok, Hita Kambhamettu, Luca Soldaini, Jonathan Bragg, Kyle Lo, Marti Hearst, Andrew Head, and Daniel S Weld. Scim: Intelligent skimming support for scientific papers. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 476–490, 2023.
- [20] 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*, pages 489–500, 2015.
- [21] Kanika Goswami, Puneet Mathur, Ryan Rossi, and Franck Deroncourt. Chartcitor: Multi-agent framework for fine-grained chart visual attribution. *arXiv preprint arXiv:2502.00989*, 2025.
- [22] Kanika Goswami, Puneet Mathur, Ryan Rossi, and Franck Deroncourt. Plotedit: Natural language-driven accessible chart editing in pdfs via multimodal llm agents. *arXiv preprint arXiv:2501.11233*, 2025.
- [23] Kanika Goswami, Puneet Mathur, Ryan Rossi, and Franck Deroncourt. Plotgen: Multi-agent llm-based scientific data visualization via multimodal feedback. *arXiv preprint arXiv:2502.00988*, 2025.
- [24] Yi Guo, Danqing Shi, Mingjuan Guo, Yanqiu Wu, Nan Cao, and Qing Chen. Talk2data: A natural language interface for exploratory visual analysis via question decomposition. *ACM Transactions on Interactive Intelligent Systems*, 14(2):1–24, 2024.
- [25] Chang Han and Katherine E Isaacs. A deixis-centered approach for documenting remote synchronous communication around data visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 2024.
- [26] Han L Han, Junhang Yu, Raphael Bournet, Alexandre Ciorascu, Wendy E Mackay, and Michel Beaudouin-Lafon. Passages: Interacting with text across documents. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2022.
- [27] Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S Weld, and Marti A Hearst. Augmenting scientific papers with just-in-time, position-sensitive definitions of terms and symbols. In *Proceedings*

of the 2021 CHI Conference on Human Factors in Computing Systems, pages 1–18, 2021.

- [28] Jeffrey Heer, Matthew Conlen, Vishal Devireddy, Tu Nguyen, and Joshua Horowitz. Living papers: A language toolkit for augmented scholarly communication. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–13, 2023.
- [29] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. International Conference on Learning Representations, ICLR, 2024.
- [30] Weihua Huang, Chew Lim Tan, and Wee Kheng Leow. Model-based chart image recognition. In *International Workshop on Graphics Recognition*, pages 87–99. Springer, 2003.
- [31] Weihua Huang, Ruizhe Liu, and C-L Tan. Extraction of vectorized graphical information from scientific chart images. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 1, pages 521–525. IEEE, 2007.
- [32] Jessica Hullman, Nicholas Diakopoulos, and Eytan Adar. Contextifier: automatic generation of annotated stock visualizations. In *Proceedings of the SIGCHI Conference on human factors in computing systems*, pages 2707–2716, 2013.
- [33] Mohammed Saidul Islam, Md Tahmid Rahman Laskar, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. Datanarrative: Automated data-driven storytelling with visualizations and texts. *arXiv preprint arXiv:2408.05346*, 2024.
- [34] Daekyoung Jung, Wonjae Kim, Hyunjoo Song, Jeong-in Hwang, Bongshin Lee, Bohyoung Kim, and Jinwook Seo. Chartsense: Interactive data extraction from chart images. In *Proceedings of the 2017 chi conference on human factors in computing systems*, pages 6706–6717, 2017.
- [35] Hyeonsu Kang, Joseph Chee Chang, Yongsung Kim, and Aniket Kittur. Threddy: An interactive system for personalized thread-based exploration and organization of scientific literature. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pages 1–15, 2022.
- [36] Hyeonsu B Kang, Tongshuang Wu, Joseph Chee Chang, and Aniket Kittur. Synergi: A mixed-initiative system for scholarly synthesis and sensemaking. In

Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, pages 1–19, 2023.

- [37] Taslim Arefin Khan, Dongwook Yoon, and Joanna McGrenere. Designing an eyes-reduced document skimming app for situational impairments. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.
- [38] Dae Hyun Kim, Enamul Hoque, Juho Kim, and Maneesh Agrawala. Facilitating document reading by linking text and tables. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 423–434, 2018.
- [39] Dae Hyun Kim, Seulgi Choi, Juho Kim, Vidya Setlur, and Maneesh Agrawala. Emphasis checker: A tool for guiding chart and caption emphasis. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):120–130, 2023.
- [40] Nicholas Kong, Marti A Hearst, and Maneesh Agrawala. Extracting references between text and charts via crowdsourcing. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 31–40, 2014.
- [41] Bum Chul Kwon, Florian Stoffel, Dominik Jackle, Bongshin Lee, and Daniel Keim. Visjockey: Enriching data stories through orchestrated interactive visualization. In *Poster compendium of the computation+ journalism symposium*, volume 3, page 3, 2014.
- [42] Chufan Lai, Zhixian Lin, Ruike Jiang, Yun Han, Can Liu, and Xiaoru Yuan. Automatic annotation synchronizing with textual description for visualization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.
- [43] Shahid Latif and Fabian Beck. Vis author profiles: Interactive descriptions of publication records combining text and visualization. *IEEE transactions on visualization and computer graphics*, 25(1):152–161, 2018.
- [44] Shahid Latif, Diao Liu, and Fabian Beck. Exploring interactive linking between text and visualization. In *EuroVis (Short Papers)*, pages 91–94, 2018.
- [45] Shahid Latif, Zheng Zhou, Yoon Kim, Fabian Beck, and Nam Wook Kim. Kori: Interactive synthesis of text and charts in data documents. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):184–194, 2021.

- [46] Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. Chartocr: Data extraction from charts images via a deep hybrid framework. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1917–1925, 2021.
- [47] Weihong Ma, Hesuo Zhang, Shuang Yan, Guangshun Yao, Yichao Huang, Hui Li, Yaqiang Wu, and Lianwen Jin. Towards an efficient framework for data extraction from chart images. In *International Conference on Document Analysis and Recognition*, pages 583–597. Springer, 2021.
- [48] Martin Weber and others. AutoTrace. <http://autotrace.sourceforge.net/>, 2002. Accessed: 2025-04-28.
- [49] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.
- [50] Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. Charagraph: Interactive generation of charts for realtime annotation of data-rich paragraphs. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2023.
- [51] Ronald Metoyer, Qiyu Zhi, Bart Janczuk, and Walter Scheirer. Coupling story to visualization: Using textual analysis as a bridge between data and interpretation. In *Proceedings of the 23rd international conference on intelligent user interfaces*, pages 503–507, 2018.
- [52] Ales Mishchenko and Natalia Vassilieva. Chart image understanding and numerical data extraction. In *2011 sixth international conference on digital information management*, pages 115–120. IEEE, 2011.
- [53] Ali Modarressi, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. Ret-llm: Towards a general read-write memory for large language models. *arXiv preprint arXiv:2305.14322*, 2023.
- [54] Haris Mumtaz, Shahid Latif, Fabian Beck, and Daniel Weiskopf. Explorative code quality documents. *IEEE transactions on visualization and computer graphics*, 26(1):1129–1139, 2019.
- [55] Fred Paas, Tamara Van Gog, and John Sweller. Cognitive load theory: New conceptualizations, specifications, and integrated research perspectives. *Educational psychology review*, 22:115–121, 2010.

- [56] Srishti Palani, Aakanksha Naik, Doug Downey, Amy X Zhang, Jonathan Bragg, and Joseph Chee Chang. Relatedly: Scaffolding literature reviews with existing related work sections. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2023.
- [57] Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulators of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.
- [58] Zhenhui Peng, Yuzhi Liu, Hanqi Zhou, Zuyu Xu, and Xiaojuan Ma. Crebot: Exploring interactive question prompts for critical paper reading. *International Journal of Human-Computer Studies*, 167:102898, 2022.
- [59] Pew Research Center. Pew research center: Nonpartisan, data-driven research on social trends, politics, and more, 2024. URL <https://www.pewresearch.org/>. Accessed: March 18, 2025.
- [60] Joao Pinheiro and Jorge Poco. Charttext: Linking text with charts in documents. *arXiv preprint arXiv:2201.05043*, 2022.
- [61] Jorge Poco and Jeffrey Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. In *Computer graphics forum*, volume 36, pages 353–363. Wiley Online Library, 2017.
- [62] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.
- [63] Napol Rachatasumrit, Jonathan Bragg, Amy X Zhang, and Daniel S Weld. Citeread: Integrating localized citation contexts into scientific paper reading. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, pages 707–719, 2022.
- [64] Donghao Ren, Matthew Brehmer, Bongshin Lee, Tobias Höllerer, and Eun Kyoung Choe. Chartaccent: Annotation for data-driven storytelling. In *2017 IEEE Pacific Visualization Symposium (Pacific Vis)*, pages 230–239. Ieee, 2017.
- [65] Hugo Romat, Nathalie Henry Riche, Ken Hinckley, Bongshin Lee, Caroline Appert, Emmanuel Pietriga, and Christopher Collins. Activeink: (th) inking with data. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

- [66] Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 393–402, 2011.
- [67] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.
- [68] Vidya Setlur and Arathi Kumar. Sentifiers: Interpreting vague intent modifiers in visual analysis using word co-occurrence and sentiment analysis. In *2020 IEEE Visualization Conference (VIS)*, pages 216–220. IEEE, 2020.
- [69] 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*, pages 365–377, 2016.
- [70] 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*, pages 966–978, 2020.
- [71] Leixian Shen, Yizhi Zhang, Haidong Zhang, and Yun Wang. Data player: Automatic generation of data videos with narration-animation interplay. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):109–119, 2023.
- [72] Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Divvala, and Ali Farhadi. Figureseer: Parsing result-figures in research papers. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 664–680. Springer, 2016.
- [73] Jaya Sreevalsan-Nair, Komal Dadhich, and Siri Chandana Daggubati. Tensor fields for data extraction from chart images: bar charts and scatter plots. In *Topological Methods in Data Analysis and Visualization VI: Theory, Applications, and Software*, pages 219–241. Springer, 2021.
- [74] Arjun Srinivasan, Steven M Drucker, Alex Endert, and John Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE transactions on visualization and computer graphics*, 25(1):672–681, 2018.

- [75] Nicole Sultanum, Fanny Chevalier, Zoya Bylinskii, and Zhicheng Liu. Leveraging text-chart links to support authoring of data-driven articles with vizflow. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2021.
- [76] Binbin Tang, Xiao Liu, Jie Lei, Mingli Song, Dapeng Tao, Shuifa Sun, and Fangmin Dong. Deepchart: Combining deep convolutional networks and deep belief networks in chart classification. *Signal Processing*, 124:156–161, 2016.
- [77] Significant Gravitas Torant. Autogpt. <https://github.com/Significant-Gravitas/AutoGPT>, 2023. Accessed: 2025-09-27.
- [78] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.
- [79] Lucy Lu Wang, Isabel Cachola, Jonathan Bragg, Evie Yu-Yen Cheng, Chelsea Haupt, Matt Latzke, Bailey Kuehl, Madeleine N van Zuylen, Linda Wagner, and Daniel Weld. Scia11y: Converting scientific papers to accessible html. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–4, 2021.
- [80] Luoxuan Weng, Xingbo Wang, Junyu Lu, Yingchaojie Feng, Yihan Liu, and Wei Chen. Insightlens: Discovering and exploring insights from conversational contexts in large-language-model-powered data analysis. *arXiv e-prints*, pages arXiv–2404, 2024.
- [81] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.
- [82] Qiang Xu and Thomas Hurtut. From graphs to words: A computer-assisted framework for the production of accessible text descriptions. In *2024 IEEE Visualization and Visual Analytics (VIS)*, pages 296–300. IEEE, 2024.
- [83] Zhiyu Yang, Zihan Zhou, Shuo Wang, Xin Cong, Xu Han, Yukun Yan, Zhenghao Liu, Zhixing Tan, Pengyuan Liu, Dong Yu, et al. Matplotagent: Method and evaluation for llm-based agentic scientific data visualization. *arXiv preprint arXiv:2402.11453*, 2024.

- [84] Jian Zhao, Shenyu Xu, Senthil Chandrasegaran, Chris Bryan, Fan Du, Aditi Mishra, Xin Qian, Yiran Li, and Kwan-Liu Ma. Chartstory: Automated partitioning, layout, and captioning of charts into comic-style narratives. *IEEE transactions on visualization and computer graphics*, 29(2):1384–1399, 2021.
- [85] Yuheng Zhao, Junjie Wang, Linbin Xiang, Xiaowen Zhang, Zifei Guo, Cagatay Turkay, Yu Zhang, and Siming Chen. Lightva: Lightweight visual analytics with llm agent-based task planning and execution. *IEEE Transactions on Visualization and Computer Graphics*, 2024.
- [86] Qiyu Zhi, Alvitta Ottley, and Ronald Metoyer. Linking and layout: Exploring the integration of text and visualization in storytelling. In *Computer Graphics Forum*, volume 38, pages 675–685. Wiley Online Library, 2019.
- [87] Yan Ping Zhou and Chew Lim Tan. Hough technique for bar charts detection and recognition in document images. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 2, pages 605–608. IEEE, 2000.
- [88] Chen Zhu-Tian and Haijun Xia. Crossdata: Leveraging text-data connections for authoring data documents. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2022.

A Appendix

Automatic Chart Data Extraction Prompt

Task Description:

You are given an image that may contain one or more charts. Your goal is to extract structured data for each chart by following the steps below.

Step 1: Identify Charts

Determine the number of **distinct charts** visible in the image.

Chart distinction rule:

- a) Treat multiple visuals as **one chart** when they share the **same set of entities**, even if they show different dependent variables, units, or legends.
 - i) Merge all dependent variables horizontally into a single markdown table.
 - ii) Keep units or qualifiers (% , \$, index, etc.) in the header.
- b) Only treat visuals as **distinct charts** when:
 - i) Entity sets differ.
 - ii) Populations or topics differ.
 - iii) Visual structures cannot be meaningfully aligned.

Separate distinct chart tables using **|**|**. Charts may appear side by side, stacked, or embedded.

Step 2: Classify Chart Types

- a) Bar chart
- b) Line chart
- c) Pie chart
- d) Other

Step 3: Identify Axes / Variables

- a) Identify independent and dependent variables.
- b) Use only labels inside or directly adjacent to the chart.
- c) Preserve full descriptive headers and unit qualifiers.
- d) Treat multiple statements as separate dependent-variable columns.
- e) Keep entity labels exactly as shown.
- f) Store results in "axes".

Step 4: Identify Colors

Identify mark colors for each dependent variable. Color count must match axis count.

Step 5: Extract or Estimate Values

- a) Record visible numeric values exactly.
- b) If missing, estimate using geometry:
 - i) Bar length or height.
 - ii) Line point position.
 - iii) Pie slice proportions.
- c) Merge multiple variables horizontally.
- d) Round to nearest integer and remove symbols.

e) If estimation is used, set "estimated": true.

Step 6: Construct Markdown Table

- a) Leftmost column is *Entity*.
- b) Remaining columns are dependent variables.
- c) Preserve visual order and descriptive headers.
- d) Separate distinct charts using **|**|**.

Step 7: Output Format

Return a single JSON object:

```
{
  "number_of_charts": <integer>,
  "chart_informations": [{
    "chart_type": <string>,
    "axes": [<string>],
    "colour": [<string>],
    "estimated": <boolean>,
    "data_table": <markdown_table_as_string>
  }]
}
```

Figure A.1: Chart data extraction prompt used for automatic chart-to-table generation.

Zero Shot Sentence-data pairing prompt

You are a chart assistant with expertise in pairing sentences from text to data points in charts. You are given the following information as input.

Input

- a) Text: {description}
- b) Data table: {data_table}

Note that the structure of the data table is in Markdown format. You will follow the structured steps below to complete the task.

Instructions

- a) Review the data table and the provided text.
- b) From the text, identify and extract sentences that contain numerical values, unique words, or relational semantics relevant to the contents of the data table.
 - i) Sentence boundaries are indicated by ., ?, !, or ;.
 - ii) Periods used in abbreviations within a sentence do not indicate sentence boundaries.
- c) Review the extracted sentences to determine whether they contain information such as colour or direction that cannot be directly derived from the data table.
 - i) Check whether a relationship between such information and the data table is present in the description.
 - ii) If a relationship exists, link the sentence to the appropriate data points.
 - iii) If no relationship exists, ignore the sentence.
- d) Match the identified sentences with data points using the following rules:
 - i) Use tuple format [a, b, c, d, ...] corresponding to rows of the data table.
 - ii) The leftmost column of the table is the independent variable and must always be included.
 - iii) A sentence may be matched with one or more tuples.
 - iv) Replace tuple elements not relevant to the sentence with "_".

Output format

Return the output strictly in JSON format with the following structure:

```
{
```

```

    "description": "description given as input",
    "matched_sentences": [
      {
        "sentence": "sentence identified",
        "data_points": [
          [a, b, c, ...],
          [a2, b2, c2, ...]
        ]
      }
    ]
  }
}

```

Points of note

- Although the example above shows double curly braces, use a single JSON object.
- Return only the JSON output and nothing else.
- The data table may contain multiple distinct tables separated by the symbol **|**|**.

Task

Based on the instructions above, return sentences paired with their matching data points in the specified JSON format.

Figure A.2: Zero Shot Sentence-data pairing prompt used for linking textual descriptions to chart data.

Few Shot Sentence-Data Pairing Prompt Template

You are a chart assistant with expertise in pairing sentences from text to data points in charts. You are given the following information.

Input

- Text (may contain one or more sentences): {**description**}
- Data table: {**data_table**}

Notes

- Data tables are in Markdown. Multiple tables may be provided; they are separated by a line exactly: **|**|**.
- The given text may contain one or more sentences. You must consider **every** sentence.
- Extract sentences and link only those that are supported by the table data.

Instructions

- Preserve sentence text **exactly** (punctuation, quotes, casing). Do not normalize the sentence string.
- A tuple represents one complete connection between a sentence and a row in the table. It links the textual statement to the numerical data associated with that entity.
 - Every tuple must include the row label and at least one valid column value.
 - Correct: ["Sub-Saharan Africa", 37.5, "-", "-"] → includes one row label and one valid column value.
 - Incorrect: ["Sub-Saharan Africa", "-", "-", "-"] → invalid because it contains no actual column values.
- A single sentence may link to multiple tuples (e.g., if it mentions several entities).
- Build one tuple per entity per sentence; do not split a single logical claim into multiple partial tuples.
- Include only columns explicitly or implicitly referenced by the sentence.
- Use "-" only when a column does not apply or is not mentioned.
- Prefer omission to speculation. If checks fail, skip that entity, except where the implicit linking rule applies.

Examples (No-split rule)

- Sentence references **both** women and men:

- i) Sentence: "In Sub-Saharan Africa, women write nearly 40% of posts while men write about 60%."
- ii) Table row: Female = 37.5, Male = 64, Unknown = 4
- iii) Bad (split into partials): ["Sub-Saharan Africa", 37.5, "-", "-"], ["Sub-Saharan Africa", "-", 64, "-"]
- iv) Good (one complete tuple): ["Sub-Saharan Africa", 37.5, 64, "-"]
- b) Sentence references **only** women:
 - i) Sentence: "In Sub-Saharan Africa, women write nearly 40% of posts."
 - ii) Table row: Female = 37.5, Male = 64, Unknown = 4
 - iii) Good: ["Sub-Saharan Africa", 37.5, "-", "-"]

Sentence separation

- a) Split the input into sentences using ., ?, ! as boundaries.
- b) Do not split inside paired quotes or parentheses.
- c) Keep quotation marks and punctuation in the sentence string.
- d) When a sentence ends with a parenthesis, include the entire parenthetical block (and any punctuation inside it).
 - i) "Support is high among older adults (62%)." → keep as one sentence.
 - ii) "Approval dropped in Japan (see Fig. 3)." → keep as one sentence.
 - iii) If a closing parenthesis comes immediately after a period (e.g., "... 62%)."), treat that as a single sentence boundary at the final period only.

Reasoning steps for linking sentences to table data

A) Entity identification (row-level grounding)

- a) Determine whether the entity in the sentence corresponds to a row label in the table.
- b) Use light normalization (lowercase, trimmed spaces) and recognize clear paraphrases when unambiguous.
 - i) "government spending" = "Public expenditure"
 - ii) "internet use" = "Online activity"
 - iii) "tariffs" = "Import duties"
- c) Do not alter the entity name in output; normalization is for matching only.
- d) If the entity cannot be clearly matched in any table, try other provided tables; if still unclear, skip rather than guessing.

B) Column selection

- a) Include only the columns explicitly or implicitly referenced by the sentence.
- b) For broad statements such as "across all regions", "overall", or "median":
 - i) Use an "All", "Full", or "Total" row if it exists.
 - ii) If the overall value is clearly derivable from the table, you may use that derived row.
 - iii) If no clear derivation is possible, skip rather than invent a number.
- c) Interpret column headers carefully; semantics determine which column(s) should be selected.
 - i) If headers express polarity (e.g., "Better in U.S." vs "Better in country of origin"), include only the column matching the sentence.
 - ii) If headers are entities (e.g., "United States", "Japan", "Men", "Women"), select the column(s) consistent with the comparison.

C) Numeric and verbal-numeric cues

- a) Identify explicit numbers and verbal numeric expressions (e.g., "majority", "plurality", "a quarter").
- b) Treat verbal numerics as approximate descriptors.
 - i) "nearly half" → roughly 50%
 - ii) "majority" → greater than 50%
 - iii) "minority" → less than 50%
 - iv) "fewer than one-in-ten" → less than 10%
 - v) "plurality" → largest value even if < 50%
- c) If phrasing and table values are inconsistent beyond reasonable interpretation, do not link that entity.
- d) If a sentence mentions a count of groups (e.g., "nine countries"), treat it as an approximate narrative cue; return all entities that satisfy the condition even if the count differs.

D) Qualitative or relational language

- a) Interpret comparative language relative to numeric evidence, and confirm the table supports the relationship.

E) Comparison direction (axis alignment)

- a) Verify that the comparison described matches the structure of the table.
- b) If multiple tables are provided (separated by |**|), evaluate against all tables and select the clearest match; if none fit, skip.
- c) Never mix cross-column and within-column reasoning in the same sentence.

F) Arithmetic reasoning

- a) Perform arithmetic only when clearly implied and operands are unambiguous.
 - i) All operands exist in the table and can be unambiguously located.
 - ii) The operation is clearly implied by the sentence.
 - iii) The result matches the sentence (exactly or approximately).
 - iv) Always include the original tuples used in the reasoning; do not fabricate new rows.
- b) Tuples related to arithmetic reasoning must be identified; no new rows may be fabricated.
- c) Convert verbal ratios (e.g., "twice as high", "half as much") to numerical equivalents ($2\times$, $0.5\times$).
- d) If operands are ambiguous or missing, skip rather than guessing.

Output format

Return only a JSON object of the form:

```
{
  "description": "<copy the input text exactly>",
  "matched_sentences": [
    {
      "sentence": "<copy the exact sentence text verbatim>",
      "data_points": [
        ["<Row Entity>", "<col1_or_\\"_\\>", "<col2_or_\\"_\\>", ...]
      ]
    }
  ]
}
```

Concrete example process

Text:

"The percentage value in favor for Ages 30-49 is 23. The ratio between Favorable and Unfavorable in the Total category is 3.31818."

Data table:

Entity	Unfavorable	Favorable
Dem/Lean Dem	68.0	25.0
Rep/Lean Rep	83.0	15.0
50+	81.0	14.0
30-49	71.0	23.0
Ages 18-29	56.0	36.0
Total	73.0	22.0

Output:

```
{
  "description": "The percentage value in favor for Ages 30-49 is 23. The ratio between Favorable and Unfavorable in the Total category is 3.31818.",
  "matched_sentences": [
    {
      "sentence": "The percentage value in favor for Ages 30-49 is 23.",
      "data_points": [["30-49", "_", 23.0]]
    },
    {
      "sentence": "The ratio between Favorable and Unfavorable in the Total category is 3.31818.",
      "data_points": [["Total", 73.0, 22.0]]
    }
  ]
}
```

Task

Based on the above, extract all sentences and return the subset that are supported by the data, pairing them with the correct tuples in the specified JSON format.

Figure A.3: Few Shot Sentence-data pairing prompt template used in the structured chart understanding pipeline.

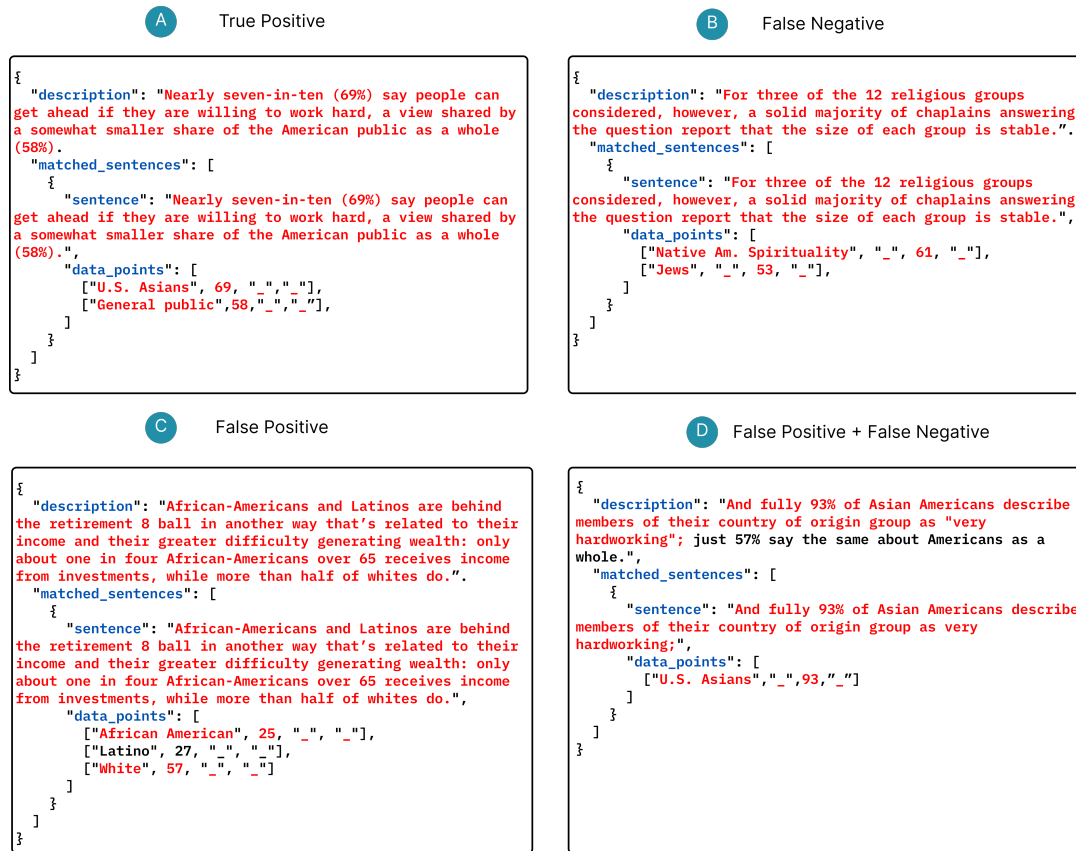


Figure A.4: An overview of sentence-data pairing outcomes produced by the sentence-data pairing framework. Example outputs illustrate different alignment cases: a correct sentence-data match (true positive), a missed data tuple despite a correct sentence (false negative), an incorrectly inserted data tuple (false positive), and a combined error where the paired sentence does not match the description, resulting in both an incorrect pairing and a missed correct alignment (false positive + false negative)