UNDERSTANDING THE INTERACTIONS BETWEEN TEXT AND VISUALIZATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Dae Hyun Kim

December 2021

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Maneesh Agrawala, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

James Landay

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Vidya Setlur

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

Abstract

Visualizations and text are commonly used together in various applications, ranging from communicative documents to interactive tools for analyzing and exploring data. However, much about the relationship between visualizations and text remains unexplored. This thesis specifically focuses on three problems related to communicating the connections between the two representations and ways to surface these connections to address the problems.

We begin by asking how readers integrate information between visualizations and text when the two representations emphasize different aspects of the underlying data. Through a user study, we find that readers can miss information presented in the text because they rely more on the visualizations for their takeaways. Based on the study results, we provide guidelines for authoring effective visualization-text pairs that doubly emphasize intended aspects of the underlying data.

Identifying references between visualizations and text, which are often also spatially separated, is a mentally taxing process. The cognitive burden disrupts the flow of reading as readers traverse back and forth between visualizations and text in an attempt to mentally link their contents. We present an interactive document reader that extends existing PDF documents based on automatically extracted references between visualizations and text. Specifically, it facilitates document reading by highlighting the references and dynamically positioning visualizations close to relevant text. Our user study shows that the interface helps readers integrate visualizations into their flow of reading by helping them identify references more quickly and more accurately.

When using a natural language interface for visualizations, users are often not informed about how the system operated on a visualization based on its interpretation of a text query. This lack of transparency leads users to question the system because they cannot easily verify the correctness of its outputs. We present a chart question answering system that generates visual explanations that clarifies how it used an input question and a visualization to obtain an answer. A user study reveals that our visual explanations significantly improve transparency and achieve levels of trust close to human-generated explanations.

Acknowledgments

I would like to thank my advisor, Maneesh Agrawala. He has taught me the ABC's of research, from defining interesting problems to communicating the results. He has also trusted and supported me throughout the program. In addition, I thank the various mentors I had over the course of my PhD, Juho Kim, Enamul Hoque, and Vidya Setlur. They have been guides towards interesting research directions, teachers of knowledge, and coworkers when pushing for deadlines. Next, I would like to express my gratitude to the members of the committee, Maneesh Agrawala, Vidya Setlur, James Landay, Pat Hanrahan and Sean Follmer for agreeing to be on the committee and providing useful insights in putting this thesis together.

I would like to mention Sherry Ruan, Naoki Eto, Sean Liu, and Geza Kovacs, Jackie Yang, as well as all the members of the Stanford HCI Group and Tableau Research for their help at various steps of the projects comprising the thesis, including providing feedback on research ideas, piloting user studies, labeling data, and many more. I am grateful for the admins, including Andrea Kuduk, Julia Gillespie, Wen Chen, Kelly Yilmaz, Jay Subramanaian, Sumalee (Jam) Kiattinant, Yaritza Lara, and Danielle Hoversten, for working with me to overcome various administrative obstacles. I also thank all my friends, both inside and outside of Stanford, for making the past few years far more than just the PhD program with many memorable moments.

I appreciate the support from Samsung Scholarship, NSF (III-1714647), Allen Institute that have funded one or more of the projects comprising this thesis.

Lastly, I would like to thank my parents, Hea Ryeon Choi and Won Seog Kim, and my sister, Soohyun Kim, for always being there for me, providing emotional support and thoughts on many important decisions I had to make.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

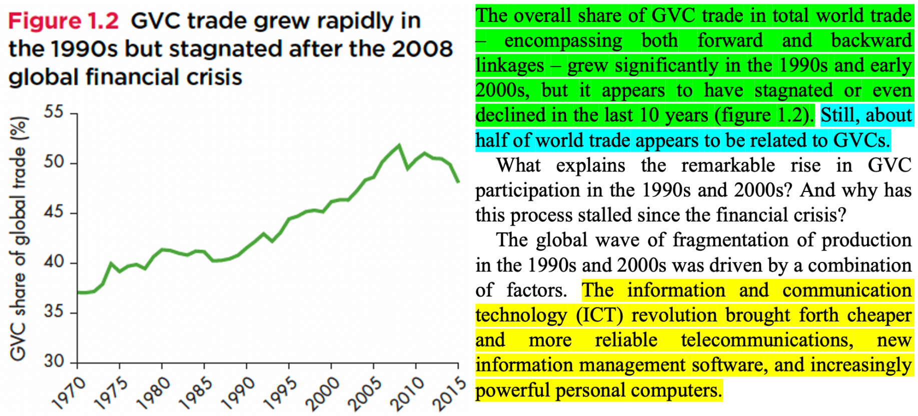


Figure 1.2 **GVC trade grew rapidly in the 1990s but stagnated after the 2008 global financial crisis**

The overall share of GVC trade in total world trade – encompassing both forward and backward linkages – grew significantly in the 1990s and early 2000s, but it appears to have stagnated or even declined in the last 10 years (figure 1.2). Still, about half of world trade appears to be related to GVCs.

What explains the remarkable rise in GVC participation in the 1990s and 2000s? And why has this process stalled since the financial crisis?

The global wave of fragmentation of production in the 1990s and 2000s was driven by a combination of factors. The information and communication technology (ICT) revolution brought forth cheaper and more reliable telecommunications, new information management software, and increasingly powerful personal computers.

Figure 1.1: Chart, caption and body text from a World Development Report from the World Bank [12]. The author uses the univariate line chart about global value chain share of global trade (lower left) with a caption (upper left) and sentences in the body text (right). The caption not only points to the rise up to 2008 and the flattening afterwards, but also brings in the external information that the global financial crisis happened in 2008. The first sentence of the body text (green) again mentions the rise and the flattening for further emphasis of the features. The next sentence (blue) further points out that the value still lingers around 50% to bring this fact into attention. The last sentence (yellow) gives an analysis of potential causes of the rise up to 2008.

As the adage, *"A picture is worth a thousand words"* states, visualizations are efficient carriers of information. They not only are compact representations of data, but also reveal and draw attention to the structures and patterns that lie within the data [164] and improve the memorability of the presented information [20, 126]. These properties make visualizations ideal for both communicating information and exploring data.

Although visualizations are powerful tools on their own, authors often pair them with text such as annotations, captions, or sentences in the body text. The text serves a variety of important roles. For example, it guides readers about how to parse the information, emphasizes certain features of the data, incorporates the presented information into a narrative, presents external information for context and provides citation of the data sources. Figure 1.1 shows a typical example of a visualization used with text and illustrates how they work together. Prior research shows that text accompanying visualizations is effective in guiding people's attention towards certain portions of the visualizations [57, 122] and that it can improve both recall and comprehension of certain aspects of the underlying data [21, 58, 66, 93, 119].

However, connecting information in text and visualizations to understand the key messages is often difficult and results in not getting the full message across. In fact, studies indicate that readers have difficulty integrating information between text and visualizations [110, 122, 124]. This often leads readers to focus on either the visualization or the caption without connecting the information provided in the two representations, resulting in a partial comprehension of the presented information [122].

One major barrier to connecting information between the two representations is the cognitive effort required to see how their contents are related. Text and visualizations are often positioned distant from each other, occasionally ending up on different pages [16, 163, 164]. To fully understand the presented information, the reader has to look back and forth between them and split attention, which leads to an increased cognitive load [10, 49, 157]. The split attention problem is further aggravated by the difficulty of figuring out the references between the text and visualizations, especially when the visualization is complex or if the text refers to multiple features in the visualization. In Figure 1.1, as readers read the first sentence (green), they would first have to establish its relevance to the chart. They then identify the features the sentence refers to: *"grew significantly in the 1990s and early 2000s"* and *"stagnated or even declined in the last 10 years"*. Next, they need to mentally parse the line and match each of the individual features to portions of the line to identify the references between the sentence and the chart. Finally, the readers can look back and forth between the chart and the sentence to find out what they each say about the features.

Another barrier to connecting information between text and visualizations is the occasional mismatch in the emphasis of information. For instance, the second sentence (blue) of Figure 1.1 refers to the fact that the value at the right end of the graph is at about 50%, which is not one of the most visually prominent features in the visualization. Given the conflicting emphasis of the text and the chart, the readers favor either the chart's visually prominent features or the text as presenting the key messages. This can lead the readers to potentially miss certain aspects of the information.

More recently, text has become an important modality for data exploration and has been incorporated into multiple commercial systems [7, 56, 72, 111, 159, 161, 177]. These systems take an input text query and create or modify visualizations or answer questions about the underlying data. The text input allows users to easily express their intents even without much experience [59, 68, 144].

Nonetheless, existing systems do not show a connection between the input text query and the operations performed on the visualization. Yet, for users to trust and rely on the results generated by these systems, the systems must transparently explain how they combined the input queries and the visualizations to arrive at the results [144].

In sum, seeing how visualizations and text are related is difficult and leads to miscommunication and lack of trust. This thesis explores how tools and design principles clarifying connections between text and visualizations can help guide people towards the intended messages.

Chapter 2 positions this thesis among the prior and concurrent progress on (1) Reading & authoring text and visualizations and (2) Natural language interfaces for visualizations.

Chapter 3 presents a user study that aims to understand what readers view as the key messages when text and visualizations do not emphasize the same features. Based on the finding that people pay more attention to the visually prominent features of visualizations and can thus miss information presented in the text, we present some design guidelines that can help keep the emphasis of text and visualizations consistent.

Chapter 4 presents an interactive document reader that helps incorporate both text and visualizations into the reading experience by dynamically positioning visualizations next to relevant text and highlighting references. Through a user study, we show how the interface can help readers identify references more quickly and more accurately and parallelize the reading of text and visualizations.

Chapter 5 presents a chart question answering system that explains how it obtained the answer using the input text query and the chart. The explanations are *visual* and refer directly to the visual features of the chart. Through a user study, we find that the visual explanations from our chart question answering system significantly improves transparency, and is able to earn trust comparable to that of human-generated explanations.

Finally, **Chapter 6** discusses limitations of this thesis and suggests potential directions for future work.

Chapter 2

Related Work

This thesis is relevant to two areas of work: (1) Reading & authoring text and visualizations and (2) Natural language interfaces for visualizations.

2.1 Reading & Authoring Visualizations and Text

The prevalent use of text and visualizations together has attracted much attention from the research community. Researchers have continued to expand our knowledge about how readers parse information in the two representations, and have designed systems that help authors and readers communicate effectively through text and visualizations.

Studies have found that text and visualizations both contribute important pieces of information. Govindaraju et al. [58] found that machine learning algorithms that use both tables and surrounding text instead of just the table or just the text can achieve a much higher F1 score. Elzer et al. [43] and Carberry et al. [23] showed that communicative signals in the graphics are often not repeated by text captions. A group of researchers asked whether using visualizations helps comprehend information presented in the text, but found inconclusive results; while some studies [22, 52, 53, 96] found significant improvements in comprehension of information presented in the text, others [81, 110, 123, 124] did not. In fact, a study by Xiong et al. [182] suggests that readers tend to not integrate information between the two representations.

Many researchers have delved deeper into how readers view text and visualizations together. One branch of research found that text guides readers' attention when looking at visualizations. Gould [57] showed that text directs the readers' attention through the visualization through an eye-tracking study. Furthermore, Xiong et al. [182] found that contents of the text that readers view prior to viewing a chart can influence what they view as visually salient. Another branch of research has looked more carefully into how readers combine and recall information in visualizations and text. Borkin et al. [19] showed that the text, such as chart titles, helps people remember messages in visualizations. Ottley et al. [122] ran an eye-tracking study to understand how readers would combine the two representations in the context of Bayesian reasoning problems. They find that readers more easily *identify* key information using visualizations but more easily *extract* key information from text. Kong et al. [85] looked closely into frames and slants in chart titles. They found that slanted frames in chart titles can bias how people view the visualizations, but people still consider the visualizations as impartial. In their later work, Kong et al. [86] steer further towards misalignment between visualizations and their titles. They observed that titles that contradict the information in the visualizations triggered readers to identify bias. However, the majority of the readers still viewed visualizations as unbiased. More recently, Lungard and Satyanarayan [104] introduced a four-level classification of text descriptions of visualizations by their content. Through a user study, they found that sighted and blind users prefer different types of text descriptions. Whereas sighted users found level 4 descriptions including context and domain knowledge as the most useful, blind users did not find them significantly more useful than basic level 1 descriptions that only explain how the charts encode data.

The work we present in Chapter 3 is along this line of work and studies how readers integrate information between visualizations and text captions. We specifically study how the emphasis of charts and text captions affects what people take away as key information.

The decoupling of text and visualizations has been long-known [10, 16, 58, 88, 122, 157, 164] and has led researchers to build tools to help readers incorporate visualizations into their flow of reading. A part of the major effort was towards reducing the context switching between the two representations by moving them closer. Tufte [165] introduced *sparklines*, which are word-sized visualizations embedded in the text to reduce the effort in traversing back and forth between text and visualizations. Chang et al. [25] presented fluid documents that allow document elements to reorganize and change style to bring supporting information into the main text. Goffin et al. [55] and Beck and Weiskopf [15] explored ways to add interactivity to sparklines, including ideas from fluid documents to allow readers to more easily access details on demand. Research has also focused on connecting information between various representations. Kong et al. [88] present a method of using crowdsourcing to identify the references between text and bar charts. The work presents an interface that highlights references between text and bar charts. Wakita and Arimoto [170] support a similar reading interface for showing links between text and visualizations in industrial reports that highlights both references and context required to understand the references. TIARA [102, 103, 173] ties text analysis with interactive visual tools to make summaries of collections of text easier to understand. Victor [169] proposed the concept of Explorable Explanations, a reading interface that encourages active reading by interacting with the data presented in text and visualizations. Dragicevic et al. [40] voice support for applying similar ideas for scientific research papers. The idea was quickly adapted by media such as Distill [39]. Recently, Crichton [34] proposed connecting portions of programming language proofs to parts of the text or prior statements to reduce the readers' burden of having to look back and forth to make sense of the proof statements.

Many of the reader-focused prototypes have not put much attention to the amount of authors' efforts that go into making contents for these document interfaces. The recent publication hiatus of the interactive academic journal platform for artificial intelligence, Distill [160], shows the importance of making it easy for authors to write and maintain content for such a model to be sustained. Fortunately, some researchers have focused on using automated or semi-automated means of helping authors generate content that involve both text and visualizations. One approach has been to introduce frameworks for authors. Latif et al. [94] describe a framework for linking sparklines and text using markup. Idyll [32] is a language for specifying web-based interactive documents. In addition, major efforts have gone into automation of various portions of the document authoring process. A major automation effort has gone into generating text captions and annotations. Qian et al. [130] surveyed existing captions in the real-world and suggested that efforts on automatic captioning should include an accurate caption generation module as well as a stitching module that combines the sentences in various ways. Many visual analysis tools include a feature for generating basic captions for charts based on how data attributes are encoded into visual attributes [35, 37, 70, 159, 167, 176]. As basic captions do not emphasize features in the visualizations, research focusing on text generation has striven to go beyond the basic encoding information and towards specific features or external information providing context. Multiple researchers have designed systems or algorithms that generate text descriptions or summaries of features in graphics and charts [23, 27, 28, 44, 47, 73, 69, 116, 120, 138, 186]. Contextifier [71] automatically adds annotations on noteworthy features in visualizations of stocks with headlines of news articles. PostGraphe [46] takes data in tabular form and a list of user-defined intents to generate text and graphics together. Researchers have also attempted to automate reference extraction between text and visualizations. Badam et al. [11] parse text and a data table and link information between them to dynamically generate visuals. Lai et al. [92] used a deep-learning approach to automate finding references between text and image charts. Kori [95] is a mixed-initiative interface for helping authors build in references between text and visualizations. Their interface uses natural language processing techniques to offer suggestions, while also allowing authors to manually construct references.

The interactive document reader we present in Chapter 4 is an automatic tool focused on helping the readers link text and visualizations by directly highlighting the references between visualizations and text, while also displaying them closer together. The work is among the first to automatically determine references between text and visualizations and generate a complete document reading interface for displaying references.

2.2 Natural Language Interfaces for Visualizations

The advances in natural language processing techniques have made text an important modality for interacting with visualizations and data. Researchers have built prototypes that respond to text queries and generate visual or text outputs. Typically, they create new visualizations [33, 38, 51, 79, 100, 105, 118, 154, 155, 156, 185], edit visualizations to surface relevant information [31, 33, 38, 68, 90, 91, 118, 143, 144, 145, 151, 152, 154, 185], or perform operations on visualizations and output a result [26, 76, 77, 100, 109, 150]. Because these systems allow even the novice users to easily express their intent, natural language interfaces have been incorporated into many commercial data analysis and visualization tools such as Tableau Software [159], Microsoft Power BI [111], IBM Watson Analytics [72], Amazon QuickSight [7], WolframAlpha [177], Google Sheets [56], and ThoughtSpot [161].

Researchers have studied how people would use natural language when interacting with visualizations and incorporated the findings into natural language interface systems. Amar et al. [5] used a collection of questions about visualizations to break analytic activity while using a visualization tool into ten low-level tasks. Works have focused on dealing with ambiguities and vagueness that are inherent properties of natural language. A large number of systems handle ambiguities in the input query by prompting the user to give further clarifications [33, 38, 51, 68, 118, 144, 152, 154, 155, 156, 185]. Hearst et al. [65] and Setlur et al. [147] studied how an intelligent system should respond when given a query including vague modifiers, such as ‘high’ or ‘expensive’. Many systems also allow users to ask follow-up queries that require information in previous queries [31, 38, 68, 144, 154, 155, 156]. There have also been effort to help users express queries that lead to meaningful exploration. Many systems include an autocompletion component based on syntax [51, 56, 68, 90, 91, 111, 144, 154, 159, 161, 177, 185]. Sneak Pique [145] and GeoSneakPique [143] are widget-based autocompletion systems that guide users towards more meaningful queries by building expectations through previews of the results of queries. Snowy [153] recommends queries based on the significance of data features and language pragmatics.

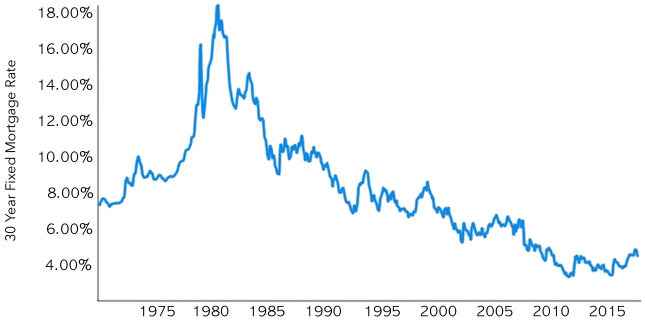
A branch of natural language interfaces that has recently gained considerable attention is question answering. Earlier works on question answering for visualizations have focused on certain types of visualizations, such as photographic images and videos (e.g., [8, 17, 60, 184]), data tables (e.g., [3, 97, 127, 183, 187, 188]) and text (e.g., [67, 131, 134, 137, 148]). Kafle et al. [76] noted that traditional question answering methods for photographic images do not generalize to charts and introduced DVQA, one of the first question answering systems for charts. This pioneering work paved the way for later researchers to focus on chart question answering as a separate problem. A number of datasets [26, 78, 109, 162] and chart question answering algorithms based on neural networks arose [26, 76, 77, 109, 133, 150].

With the advancement of artificial intelligence an increased concern has emerged regarding transparency. Why and how a system makes a decision is often difficult to comprehend. This issue has not been overlooked by the research community and many researchers have studied ways to design systems that are explainable by shedding light on the relationship between the input and output and its decision-making mechanisms. Some focus on explaining how the system arrived at each output [1, 135, 136, 149, 174]. Others focus on explaining the system's general behavior [4, 13, 50, 114, 180]. For instance, the confusion wheel introduced by Alsallakh et al. [4] visualizes the results of a multi-class classifier on a large number of data points and helps the users see potential reasons for classes of errors. Setlur et al. [144] noted that transparency is also an important problem for the natural language interface community to address. Yet, not much work has been done to introduce explainability into natural language interfaces.

The work we present in Chapter 5 is the first to address issues around transparency and trust of chart question answering systems through visual explanations. Since the work, researchers focusing on chart question answering have begun to take transparency into account. Singh and Shekhar [150] demonstrated that their structural transformer based model can achieve a level of interpretability.

Chapter 3

How Readers Integrate Charts and Captions[1]



Charts often contain visually prominent features that draw attention to aspects of the data and include text captions that emphasize aspects of the data. Through a crowdsourced study, we explore how readers gather takeaways when considering charts and captions together. We first ask participants to mark visually prominent regions in a set of line charts. We then generate text captions based on the prominent features and ask participants to report their takeaways after observing chart-caption pairs. We find that when both the chart and caption describe a high-prominence feature, readers treat the doubly emphasized high-prominence feature as the takeaway; when the caption describes a low-prominence chart feature, readers rely on the chart and report a higher-prominence feature as the takeaway. We also find that external information that provides context, helps further convey the caption's message to the reader. We use these findings to provide guidelines for authoring effective chart-caption pairs.

[1] The contents of this chapter has been adapted from Kim et al. [84] (`https://doi.org/10.1145/3411764.3445443`). The thesis author was the first author and a major contributor to the work.

3.1 Introduction

Charts provide graphical representations of data that can draw a reader's attention to various visual features such as outliers and trends. Readers are initially drawn towards the most *visually salient* components in the chart such as the chart title and the labels [107]. However, they eventually apply their cognitive processes to extract meaning from the most *prominent* chart features [24, 163]. Consider the line chart at the beginning of this article. What do you think are the main visual features of the chart and what are its key takeaways?

Such charts are often accompanied by text captions that emphasize specific aspects of the data as chosen by the chart author. In some cases, the data emphasized in the caption corresponds to the most visually prominent features of the chart and in other cases it does not. Prior studies have shown that charts with captions can improve both recall and comprehension of some aspects of the underlying information, compared to seeing the chart or the caption text alone [21, 66, 93, 119]. But far less is known about how readers integrate information between charts and captions, especially when the data emphasized by the visually prominent features of the chart differs from the data that is emphasized in the caption.

Consider the visually prominent features in our initial line chart and then consider each of the following caption possibilities one at a time. How do your takeaways change with each one?

(1) The chart shows the 30-year fixed mortgage rate between 1970 and 2018.

(2) The 30-year fixed mortgage rate increased slightly from 1997 to 1999.

(3) The 30-year fixed mortgage rate reached its peak of 18.45% in 1981.

(4) The 30-year fixed mortgage rate reached its peak of 18.45% in 1981 due to runaway inflation.

The first caption simply describes the dimensions graphed in the chart and only provides redundant information that could be read from the axis labels. Automated caption generation tools often create such *basic* descriptive captions [111, 159]. The next three captions each emphasizes aspects of the data corresponding to a visual feature of the chart (i.e., upward trend, peak) by explicitly mentioning the corresponding data point or trend. However, the second caption emphasizes a feature of low visual prominence – a relatively local and small rise in the chart between 1997 and 1999. The third caption describes the most visually prominent feature of the chart – the tallest peak that occurs in 1981. The final caption also describes this most visually prominent feature, but adds external information that is not present in the chart and provides context for the data.

In this chapter, we examine two main hypotheses - (1) When a caption emphasizes more visually prominent features of the chart, people are more likely to treat those features as the takeaway; even when a caption emphasizes a less visually prominent feature, people are still more likely to treat a more visually prominent feature in the chart as the takeaway. (2) When a caption contains external information for context, the information serves to further emphasize the feature described in the caption and readers are therefore more likely to treat that feature as the takeaway.

We considered univariate line charts for our work because they are among the most common basic charts and are easily parameterizable, making them useful for the initial exploration of our hypotheses. We synthesized 27 single charts with carefully chosen parameters and collected 16 real-world single line charts to confirm the generalizability of our findings. We ran a data collection activity on the 43 single-line charts, where we asked 219 participants to mark visually prominent regions on the line charts. We generated text captions for the ranked set of prominent features using templates to control variations in natural language. Finally, we conducted a crowdsourced study with a new set of 2168 participants to report their takeaways after seeing the chart-caption pairs.

Our findings from the study support both of our hypotheses. Referring back to our initial line chart, when the caption mentions the most prominent feature as in the third caption (i.e., the peak in 1981), readers will probably take away information from that feature. When the caption mentions a less prominent feature as in the second caption (i.e., the increase from 1997 to 1999), there is a mismatch in the message between the chart and the caption. Readers will have a strong tendency to go with the message conveyed in the chart and take away information about the peak value. Finally, the external information about the peak value present in the fourth caption will reinforce the message in the caption and the readers will more likely take away information about the peak.

These findings help better understand the relationship between charts and their captions when conveying information about certain aspects of the data to the reader. Based on these studies, we provide guidelines for authoring charts and captions together in order to emphasize the author's intended takeaways. Visualization authors can more effectively convey their message to readers by ensuring that both charts and captions emphasize the same set of features. Specifically, authors could make visual features that are related to their key message, more prominent through visual cues (e.g., highlighting or zooming into a focus area, adding annotations [42, 98] or include external information in the caption to further emphasize the feature described in the caption. Often, an alternative chart representation may be more conducive to making certain visual features more prominent.

3.2 Study

We conducted a crowdsourced study to understand how captions describing features of varying prominence levels and the effect of including or not including external information for context, interacts with the chart in forming the readers' takeaways. Through an initial data collection activity, we asked participants to identify features in the line charts that they thought were visually prominent. We generated captions corresponding to those marked features of various levels of prominence. We then ran a study asking a new set of participants to type their takeaways after viewing a chart and caption pair. Figure 3.1 shows the study pipeline.

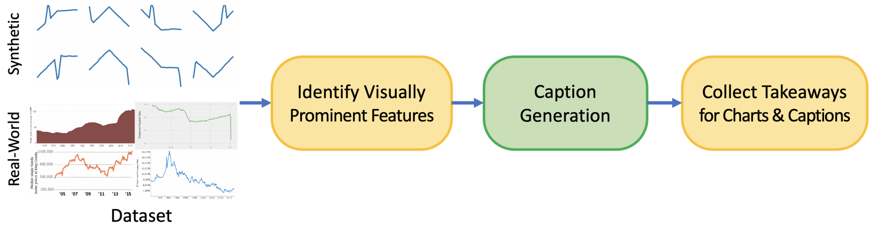

Synthetic
Real-World
Dataset
Identify Visually Prominent Features
Caption Generation
Collect Takeaways for Charts & Captions

Figure 3.1: Our study pipeline. The inputs to the study are 27 synthetic and 16 real-world charts. Yellow boxes represent steps where we employed crowdsourcing. The green box indicates that the step did not involve crowdsourcing.

3.2.1 Datasets

We ran the study on two different datasets - (1) synthetically generated line charts that we designed to ensure good coverage of a variety of visual features that occur in line charts and (2) line charts gathered from real-world sources to serve as a more ecologically valid setting for our study.

Synthetic Charts. We generated a set of synthetic line charts with common visual features (i.e., trends, extrema, and inflection points) while maintaining realistic global shapes. To keep the overall design space tractable, we limited global shapes to include at most two trends (i.e., up, down, and flat) and added at most one perturbation to induce features (e.g. inflection points) in either the positive or negative direction, resulting in a total of 27 data shapes (Figure 3.2). To provide context to the charts, we labeled the x-axis with time unit values implying that the chart represents a time series. Specifically, we selected the start and end of the x-axis from the set of years {1900, 1910, 1920,..., 2020}. To label the y-axis, we chose a domain for the y-axis and its value range from the MassVis dataset [20].

Real-world Charts. To build a more ecologically representative dataset of line charts with various shapes, styles, and domains, we collected 16 charts (Figure 3.3) from sources such as The Washington Post [172], Pew Research [128], Wikipedia [175], and Tableau Public [158]. Because our study focuses on prominence arising from intrinsic features in line charts, we removed all graphical elements that could potentially affect the prominence of the features in the charts (e.g., text annotations, highlighting, and background shading). In addition, we removed all text except for the axis labels (e.g. chart titles) so that the captions serve as the primary source of text provided with the chart. We added axis labels to those charts without labels to ensure readability.

3.2.2 Identify Visually Prominent Features

To identify the most visually prominent features in our dataset, we recruited at least five workers from Amazon Mechanical Turk [6] for each line chart and asked them to draw rectangular bounding boxes around the top three most prominent features in the chart. We also asked them to briefly

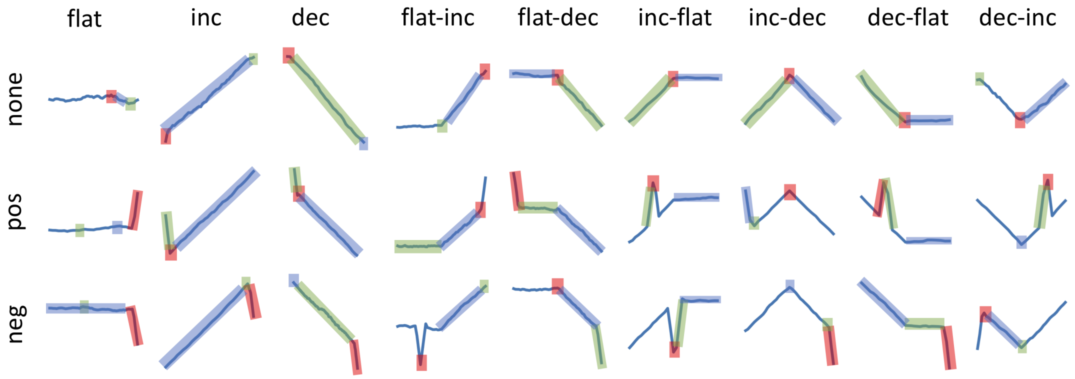


Figure 3.2: The 27 data shapes generated for the study and their top three prominent features. Columns represent the nine possible global shapes and rows represent the three possible local outlier types. Here, ‘flat’, ‘inc’, and ‘dec’ denote flat, increasing, and decreasing trends respectively. ‘none’, ‘neg’, and ‘pos’ denote none, negative, and positive outlier types respectively. Red, green, and blue regions indicate the top three prominent features in order.

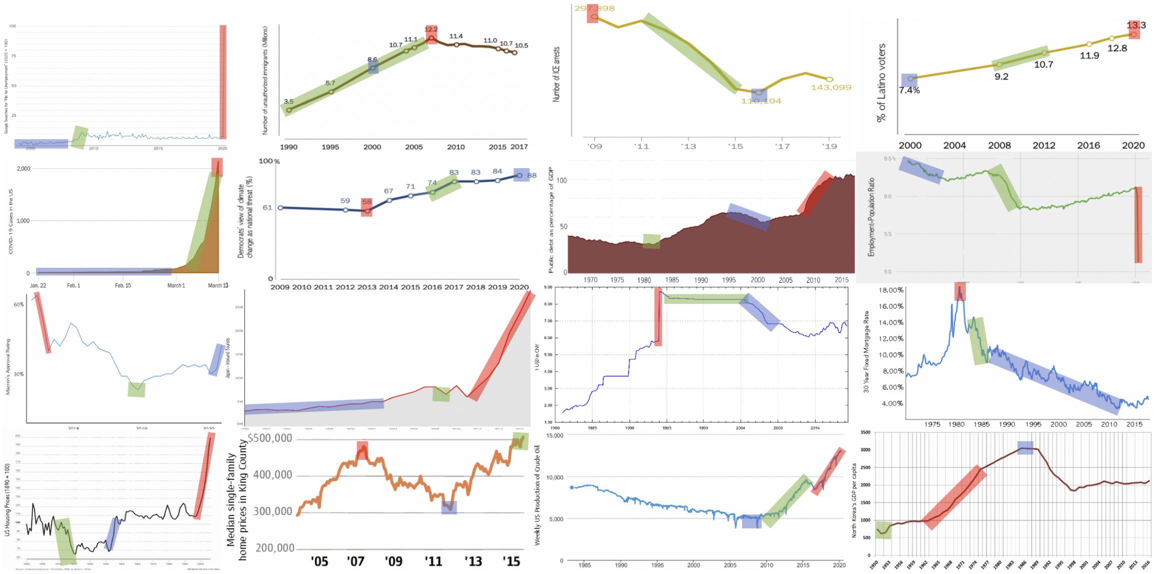
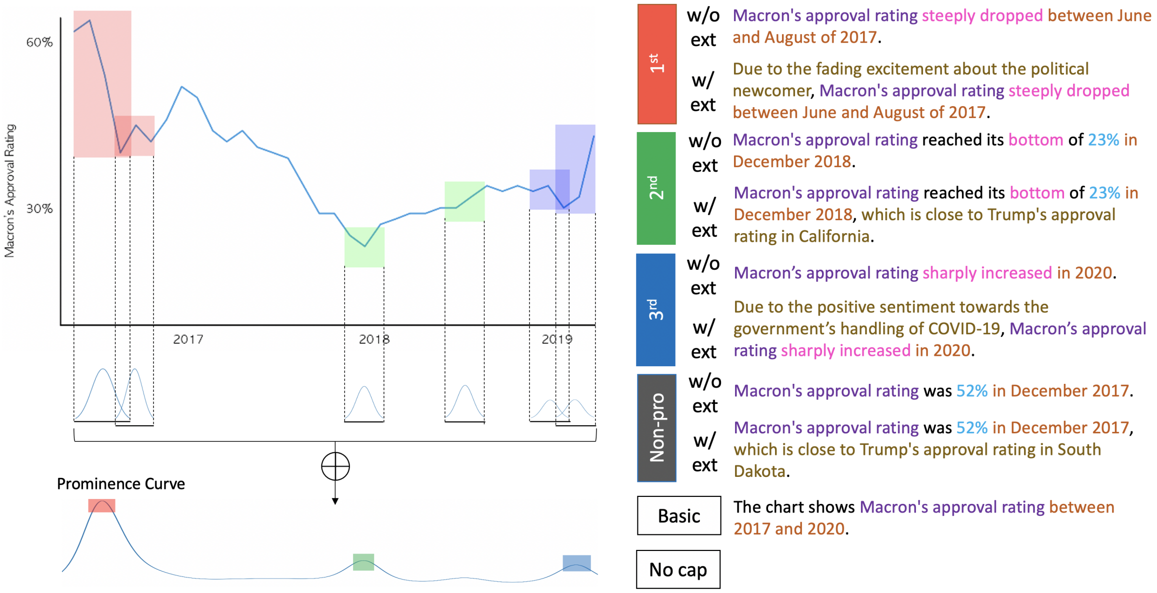


Figure 3.3: The 16 real-world charts. Red, green, and blue regions indicate the top three prominent features in order.



1st	w/o ext	Macron's approval rating steeply dropped between June and August of 2017.
	w/ ext	Due to the fading excitement about the political newcomer, Macron's approval rating steeply dropped between June and August of 2017.
2nd	w/o ext	Macron's approval rating reached its bottom of 23% in December 2018.
	w/ ext	Macron's approval rating reached its bottom of 23% in December 2018, which is close to Trump's approval rating in California.
3rd	w/o ext	Macron's approval rating sharply increased in 2020.
	w/ ext	Due to the positive sentiment towards the government's handling of COVID-19, Macron's approval rating sharply increased in 2020.
Non-pro	w/o ext	Macron's approval rating was 52% in December 2017.
	w/ ext	Macron's approval rating was 52% in December 2017, which is close to Trump's approval rating in South Dakota.
Basic		The chart shows Macron's approval rating between 2017 and 2020.
No cap		

Figure 3.4: The line on the bottom left shows the prominence curve for the line chart above. From this curve, we obtain the most prominent (red), the second most prominent (green), and the third most prominent (blue) features in the chart. The 10 caption variants (one of them being a no-caption variant) generated based on these prominent features, are shown on the right. The text colors indicate the types of fill-in values based on the caption templates; **purple** for dimensions, **fuchsia** for the feature description, **blue** for data values, and **brown** for the time period.

describe each marked feature in their own words so that we could differentiate between trend and slope features versus peak, inflection, and other point features.

In each trial of the data collection, we presented one of the 43 line charts. Because we were seeking subjective responses, each participant completed only one trial to avoid biases that might arise from repeated exposure to the task. Participation was limited to English speakers in the U.S. with at least a 98% acceptance rate and 5000 approved tasks. We payed a rate equivalent to $2 / 10 mins.

We asked a total of 219 participants (average of 5.09 per chart) to label the top three features for a total of 657 prominence boxes. We then aggregated all of the feature bounding boxes provided by first projecting each box onto the x-axis, to form a 1D interval (Figure 3.4 upper left). We weighted each interval inversely proportional to the ranking provided by the participant. Specifically, the top ranked feature bounding box for each participant was assigned a weight of 3, while the 3rd ranked feature was assigned a weight of 1. We noticed that bounding boxes corresponding to the same features were pretty consistent in the central regions although the exact boundary drawn by the participants varied. In order to boost the signal in the central regions while suppressing the noise in the boundary regions, we multiplied the weight assigned to each interval by a Gaussian factor centered at the interval and with standard deviation set to half the width of the interval. Summing all of the Gaussian weighted intervals, we obtained a *prominence curve* (Figure 3.4 bottom left). However, a region defined by a local maximum of the curve may not have an obvious one-to-one mapping with a feature in the chart because it roughly indicates a high prominence region instead of pinpointing a specific visual feature. We considered all the bounding boxes containing the region along with the participants' text descriptions of the features to associate the local maximum to a certain feature. We iterated this process for the region around the top three local maximum to identify three prominent features. Results of the algorithm for the charts in our dataset are shown in Figures 3.2 and 3.3.

3.2.3 Caption Generation

Feature	Template
Extremum	`[dimension]` *reached its* `[extrema-word]` *of* `[value]` *in* `[time-period]`.
Trend	`[dimension]` `[slope-word]` *in / between* `[time-period]`.
Inflection	`[dimension]` *started* `[slope-word]` *in* `[time-period]`.
Point	`[dimension]` *was* `[value]` *in* `[time-period]`.

Table 3.1: Examples of templates we employed for generating captions about specific features. The text colors indicate the types of fill-in values based on the caption templates; **purple** for dimensions, **fuchsia** for feature descriptions, **blue** for data values, and **brown** for time periods. Examples of filled in captions are in Figure 3.4 (right).

To carefully control the language used in the captions and keep the number of conditions manageable, we generated captions using templates that only vary the feature mentioned and whether external information is introduced. Using the templates, we produced the following caption variants: (1) two captions (one with and one without external information) for each of the top three visually prominent features identified earlier, (2) two caption (one with and one without external information) describing a minimally prominent feature that is neither an extremum nor an inflection point, and (3) a basic caption that simply describes the domain represented in the chart without describing a particular feature.

We generated 10 caption variants (including the no caption variant in which we presented a chart without caption) for each of the 43 charts, providing a total of 430 chart-caption pairs. We manually generated all the captions rather than using the original captions for the real-world charts to control for word use and grammatical structure. For real-world charts, we searched for information from the document that they originally appeared in, to extract information not present in the charts. In particular, we looked for information about potential reasons for trends or change (e.g., the external information included in the caption about the most prominent feature in Figure 3.4) or comparisons with a similar entity (e.g., comparison between Macron's approval rating with Trump's approval rating in the second most prominent feature in Figure 3.4). For synthetically generated charts and real-world charts that were not accompanied with additional information about their features, we referenced Wikipedia [175] articles to create a plausible context.

We employed simple language templates for caption generation to minimize the effects of linguistic variation (Table 3.1). The captions generated with the templates were allowed to vary in the features they describe in the charts. To make the descriptions of the features appear natural, we used words the participants used to describe the features during the prominent feature collection phase. Because the participants usually described each of the features using a noun occasionally with an adjective modifier (e.g., *"sharp increase"*), we manually lemmatized the words and modified the forms to correctly fit into our template (e.g., *"sharply increased"* in the caption about the third most prominent figure in Figure 3.4).

3.2.4 Collect Takeaways for Charts & Captions

Design

We ran a between-subjects design study for collecting takeaways for charts and their captions. For each of the 43 charts, we presented one of the ten variants (including the no caption variant) (examples in Figure 3.4):

(1) [1st w/o ext] Caption for most prominent feature, no external info.

(2) [1st w/ ext] Caption for most prominent feature, has external info.

(3) [2nd w/o ext] Caption for 2nd most prominent feature, no external info.

(4) [2nd w/ ext] Caption for 2nd most prominent feature, has external info.

(5) [3rd w/o ext] Caption for 3rd most prominent feature, no external info.

(6) [3rd w/ ext] Caption for 3rd most prominent feature, has external info.

(7) [non-pro w/o ext] Caption for non-prominent feature, no external info.

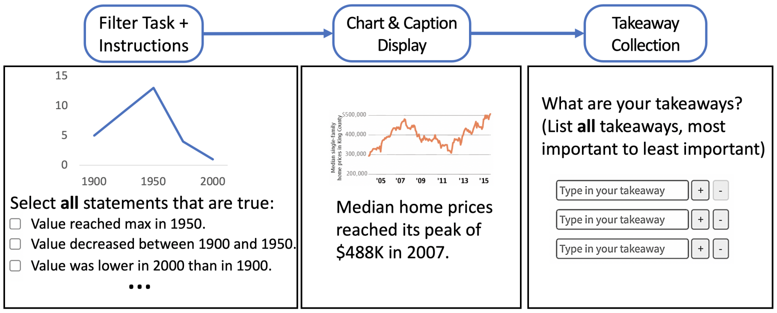
(8) [non-pro w/ ext] Caption for non-prominent feature, has external info.

(9) [basic] Caption about domain represented in the chart and x-range

(10) [no cap] No caption

Procedure

The study began with a screening test to ensure that the participant had a basic understanding of line charts and could read values and encodings, extract extrema and trends, and compare values (Figure 3.5 first step). Only participants who passed this test were allowed to continue with the study. After they read the instructions, the participants were presented with a chart and a caption underneath the chart, similar to most charts in the real world (unless it is the no-caption variant) (Figure 3.5 second step). We did not impose a time constraint on the amount of time spent looking at the chart and the caption to allow participants sufficient time to read and digest the information at their own pace, like document reading in the real world. On the next screen for collecting



Select **all** statements that are true:

- □ Value reached max in 1950.
- □ Value decreased between 1900 and 1950.
- □ Value was lower in 2000 than in 1900.

•••

Median home prices reached its peak of $488K in 2007.

What are your takeaways? (List **all** takeaways, most important to least important)

Type in your takeaway

Type in your takeaway

Type in your takeaway

Figure 3.5: The procedure for collecting takeaways for chart-caption pairs. The images show simplified versions of the screen that the participants saw during each step.

takeaways, the chart and the caption were removed to constrain readers to provide the takeaways based on memory instead of simply re-reading from the chart and the caption. The participants were asked to list as many text takeaways as they could in the order of importance (Figure 3.5 third step). Finally, using a 5-point Likert scale, we asked how much they relied on the chart and caption individually when determining their takeaways.

We asked each participant to provide takeaways for exactly one chart-caption pair to prevent potential biases from already having read a different caption about a chart. From 2168 participants (average of 5.04 per chart-caption pair), we collected a total of 4953 takeaways (average of 2.28 per participant).

Labeling Takeaways

In order to analyze the takeaways, we manually labeled each takeaway with the corresponding chart feature described. Since participants often described multiple chart features in a single takeaway, we first split each takeaway into separate takeaways for each visual feature mentioned. At the end of this process, we identified on average 1.31 features per takeaway. If the referenced feature was one of three most prominent features or the non-prominent feature we identified during caption generation, we labeled the takeaway with the corresponding feature, otherwise we labeled the takeaway as referring to an *other* feature. If the takeaway did not refer to any specific feature in the chart, we labeled the takeaway as a *non-feature*. Examples of *non-feature* takeaways include an extrapolation such as *"The value will continue to rise after 2020"* or a judgment such as *"I should buy gold"* when looking at a chart showing the price of gold over time. One of the authors labeled the features and discussed any confusing cases with the other authors to converge on the final label.

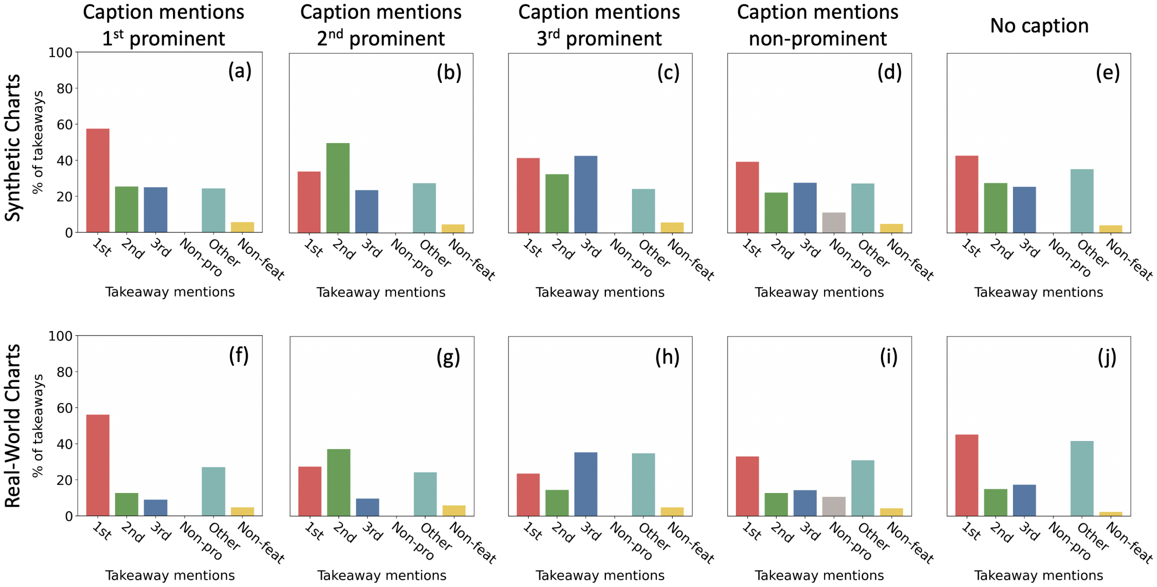


Figure 3.6: Study results. Each column shows bar charts for each prominence level mentioned in the caption (i.e., the leftmost bar chart is for captions mentioning the 1st ranked visual feature, the next bar chart is for captions mentioning the 2nd ranked visual feature, while the rightmost bar chart is for the no-caption condition). Within a bar chart, each bar represents the percentage of takeaways mentioning the visual feature at that prominence level. For example, the leftmost bar in each bar chart represents the percentage of total takeaways that mention the top ranked takeaway. Each bar chart also reports the percentage of *Other* features and *Non-features* that were mentioned in the takeaways. These charts aggregate data for captions with and without external information. The percentages do not sum to 100% as some takeaways mention multiple takeaways.

3.3 Results

The primary goal of our study is to understand what readers take away when charts and captions are presented together and how the emphasis on different prominent features and presence of external information affects the takeaways. We analyze our results with respect to two hypotheses:

[**H1**] When captions emphasize more visually prominent features of the chart, people are more likely to treat the features as the takeaway; when a caption emphasizes a less visually prominent feature, people are less likely to treat that feature as the takeaway and more likely to treat a more visually prominent feature in the chart as the takeaway.

[**H2**] When captions contain external information for context, the external information serves to further emphasize the feature presented in the caption and people are therefore more likely to treat that feature as the takeaway, compared to when the caption does not contain external information.

Assessing H1. To evaluate **H1**, we examine how varying the prominence of a visual feature mentioned in a caption (independent variable), affects the visual feature mentioned in the takeaways

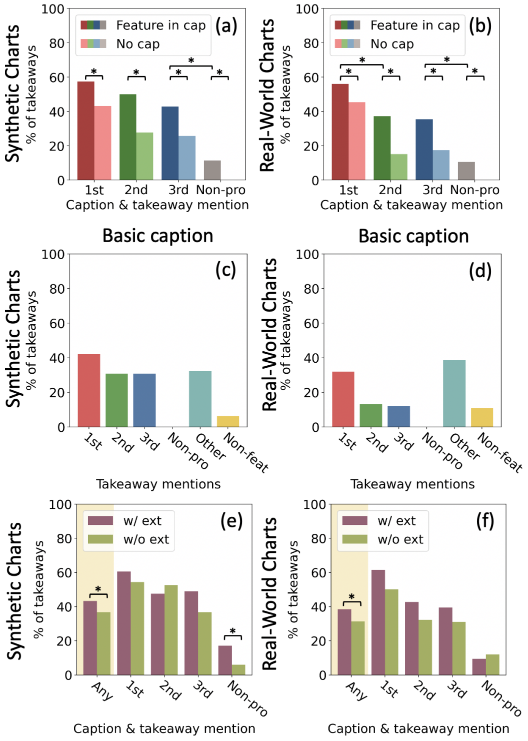


Figure 3.7: (Top row) Comparison of percentages of takeaways that mention the same feature as the caption for the synthetic (a) and real-world (b) datasets (i.e., darker bars on the left correspond to the red bar from Figure 3.6a, the green bar from 3.6b, the blue bar from 3.6c, and the grey bar from 3.6d), and percentages of takeaways that mention the feature in the no caption condition (i.e., the right lighter-hued bars in the chart correspond to the bars from Figure 3.6e). (Middle row) Percentage of takeaways mentioning the visual features at each prominence level when presented with the basic caption. (Bottom row) Dividing the left bars in charts (top row)a and (top row)b based on whether the caption contains external information (purple bars) or does not (olive bars). The leftmost *Any* bars show aggregates over all prominence levels. Asterisks indicate significant difference.

(dependent variable) by our study participants. Figure 3.6 summarizes the study results for the synthetic charts (top row) and the real-world charts (bottom row).

In general, these results suggest that when a caption mentions visual features of differing prominence levels, the takeaways also differ. Omnibus Pearson's chi-squared tests confirm a significant difference between the bar charts for the 5 different caption conditions in both the synthetic ($\chi^2(20) = 202.211$, $p < 0.001$) and real world ($\chi^2(20) = 207.573$, $p < 0.001$) datasets. These results also suggest that when the caption mentions a specific feature, the takeaways also tend to mention that feature, when compared to the baseline 'no-caption' condition.

Figures 3.7a and 3.7b collect the percentage of takeaways that mention the same feature as in the caption for the synthetic and the real-world datasets respectively (left darker bars) and compare

	Caption-Takeaway 1		Caption-Takeaway 2			
Source	Caption	Takeaway	Caption	Takeaway	Z	p
Block 1. Takeaways mentioning feature in caption vs. without caption						
Synthetic	1st	1st	no cap	1st	2.846	0.002*
	2nd	2nd	no cap	2nd	4.641	< 0.001*
	3rd	3rd	no cap	3rd	3.643	0.001*
	non-pro	non-pro	no cap	non-pro	6.195	< 0.001*
Real-world	1st	1st	no cap	1st	1.660	0.049
	2nd	2nd	no cap	2nd	4.225	< 0.001*
	3rd	3rd	no cap	3rd	3.347	< 0.001*
	non-pro	non-pro	no cap	non-pro	4.732	< 0.001*
Block 2. Between takeaways mentioning feature in caption						
Synthetic	1st	1st	2nd	2nd	1.782	0.037
	2nd	2nd	3rd	3rd	0.705	0.044
	3rd	3rd	non-pro	non-pro	8.989	< 0.001*
Real-world	1st	1st	2nd	2nd	3.708	< 0.001*
	2nd	2nd	3rd	3rd	0.363	0.358
	3rd	3rd	non-pro	non-pro	5.940	< 0.001*
Block 3. When caption = 1st: takeaway = 1st vs. takeaway ≠ 1st						
Synthetic	1st	1st	1st	2nd	8.168	< 0.001*
	1st	1st	1st	3rd	8.275	< 0.001*
	1st	1st	1st	non-pro	19.463	< 0.001*
Real-world	1st	1st	1st	2nd	9.981	< 0.001*
	1st	1st	1st	3rd	11.301	< 0.001*
	1st	1st	1st	non-pro	11.536	< 0.001*
Block 4. When caption ≠ 1st: takeaway = 1st vs. takeaway = caption						
Synthetic	2nd	2nd	2nd	1st	3.829	< 0.001*
	3rd	3rd	3rd	1st	0.258	0.398
	non-pro	1st	non-pro	non-pro	8.342	< 0.001*
Real-world	2nd	2nd	2nd	1st	2.010	0.022
	3rd	3rd	3rd	1st	2.521	0.006*
	non-pro	1st	non-pro	non-pro	5.454	< 0.001*

Table 3.2: Pairwise Z-test results of comparisons between various ratios of takeaways that mention a certain feature (third, fifth columns) when provided a caption describing a certain feature (second, fourth columns). The tests were one-sided with the alternative hypothesis that the ratio of takeaways for 'Caption-Takeaway 1' is greater than the ratio of takeaways for 'Caption-Takeaway 2'. Asterisks indicate significance with Bonferroni correction.

them with the percentages corresponding to the no-caption case (lighter-hued bars on the right). We see that captions do play a role in forming takeaways and the takeaway is thus more likely to mention that feature (i.e., each darker bar in Figures 3.7a and 3.7b is usually longer than the corresponding lighter-hued bar to its right). Planned pairwise Z-tests with Bonferroni correction are shown in Table 3.2. Block 1 shows that the differences between the corresponding color bars are significant for the second most prominent, third most prominent, and non-prominent features. For the most prominent feature, we find that while a higher proportion of people mentioned the most prominent feature in their takeaways when the caption mentions it, the difference is only significant for the synthetic charts. We believe that this is possibly because people already include the most prominent features in their takeaways in the no-caption condition and the difference hence is not significant.

While we confirmed that both the chart and caption play a role as to what the reader takes away from them, the key question is how the chart and the caption interact with each other – Do they have a synergistic effect when they emphasize the same feature? Which one wins over when they emphasize different features? Referring to Figure 3.6, we see the synergistic effect of the double-emphasis from the chart and caption when they emphasize the same feature (Figures 3.6a and 3.6f). In particular, the participants took away from the most prominent feature significantly more often than from any other feature in the chart (Table 3.2 Block 3). When the caption diverged from the chart and described a feature that was not prominent, the participants relied more on the chart and took away from the most prominent feature significantly more than the feature described in the caption (Table 3.2 Block 4, rows 3 and 6; Figures 3.6d and 3.6i). When the caption did not diverge as much and described the second or the third most prominent feature, the takeaways mentioned the feature described in the caption more than the most prominent feature (Table 3.2 Block 4, rows 1, 2, 4, and 5; Figures 3.6b, 3.6c, 3.6g, and 3.6h). However, the difference was smaller than the difference between the ratio of people who took away from the most prominent feature and the ratio of people who took away from any of the other features. We believe this result may be due to the fact that the charts still had more influence on the readers than the captions as the second and the third most prominent feature are still among the top prominent features and are among the features emphasized by the chart.

We observe from Figure 3.7 that the chart also plays an important role in what people take away – when a caption mentions a higher-prominent feature, the takeaways more consistently mentions that feature. Specifically, we see that the bars for the higher-prominence features are taller than the bars for the lower-prominence features, indicating an increase in the effectiveness of chart in reinforcing the message in the caption. Planned pairwise Z-tests with Bonferroni correction between each subsequent pair of bars (red bar vs. green bar, green bar vs. blue bar, blue bar vs. gray bar) (Table 3.2 Block 2) find that the red bar vs. green bar is significant for real-world charts and the blue bar vs. gray bar is significant both synthetic and real-world charts, whereas the green bar vs. blue bar difference is not significant. We believe that the visual prominence levels for some of the top-ranked features are similar in several charts (i.e., the difference in prominence between the 1st and 2nd ranked features is small) in our dataset and this results in a smaller difference between them, although the trend is in the right direction.

Table 3.3 shows average and standard deviation of how much the participants reported to have relied on the chart and the caption respectively on a 5-point Likert scale. The results in Table 3.3 Block 1 suggest that the participants drew information from both the chart and the caption when determining their takeaways, although they consistently relied on the chart more than the caption. These results potentially shed light on why participants took away more often from the chart than the caption when they start to diverge – they relied more on the chart than the caption. The results further suggest that the participants' tendency to rely on the charts grew while their tendency to

		Reported Reliance	
Source	Caption Type	Chart	Caption
Block 1. Overall			
Synthetic	all	4.675 ± 0.670	2.624 ± 1.609
Real-world	all	4.536 ± 0.784	2.779 ± 1.679
Block 2. Prominence			
Synthetic	1st	4.590 ± 0.711	3.249 ± 1.327
	2nd	4.567 ± 0.814	3.082 ± 1.433
	3rd	4.567 ± 0.726	3.059 ± 1.408
	non-pro	4.775 ± 0.549	2.447 ± 1.429
	basic	4.850 ± 0.377	2.593 ± 1.320
Real-world	1st	4.494 ± 0.838	3.405 ± 1.481
	2nd	4.462 ± 0.890	3.165 ± 1.359
	3rd	4.503 ± 0.805	3.236 ± 1.354
	non-pro	4.595 ± 0.718	2.680 ± 1.545
	basic	4.628 ± 0.601	2.718 ± 1.568
Block 3. External Information			
Synthetic	w/o ext	4.679 ± 0.688	2.798 ± 1.402
	w/ ext	4.573 ± 0.728	3.110 ± 1.448
Real-world	w/o ext	4.606 ± 0.741	3.061 ± 1.481
	w/ ext	4.424 ± 0.875	3.194 ± 1.439

Table 3.3: The reported reliance on the chart and the caption respectively on 5-point Likert scales. Block 1 shows the reported reliance across all the captions. Block 2 shows the reported reliance depending on the prominence of the feature described in the chart and Block 3 shows the reported reliance depending on the inclusion of external information. The values are reported in the form of $\mu \pm \sigma$.

rely on the captions declined as the prominence of the feature described in the caption decreased (Table 3.3 Block 2). We found a significant drop in the self-reported reliance on the caption when the caption described a non-prominent feature compared to when it described the third-most prominent feature (synthetic: Mann-Whitney $U = 28941$, $p < 0.001$; real-world: Mann-Whitney $U = 9666$, $p < 0.001$) whereas the increase in the reported reliance on the chart when the caption described a non-prominent feature compared to when it described the third-most prominent feature was only significant with the synthetic charts (Mann-Whitney $U = 32844.5$, $p < 0.001$). Although the general trend is in the right direction, we did not find significant differences in the reliance scores when the caption mentioned one of the top three prominent features. This may be because the difference in prominence is not as great among these features as it is with the non-prominent feature. These results are in line with our findings from the takeaways; we find that when the chart contains a high-prominence visual feature, but the caption emphasizes a low-prominence feature, participants relied more on the chart and less on the caption.

Considering all these results together suggests that we can accept our hypothesis **H1** – readers take away from the highly prominent features when the chart and caption both emphasize the same feature and that their inclination to rely more on the most prominent feature instead of the feature described in the caption becomes greater when the caption describes a less prominent feature.

H1 Additional Results. We also collected takeaways for charts with *basic* captions that describe the axes of the chart. (Figure 3.7 - middle row). We find that the percentage of takeaways for each of the features is similar to that of the no-caption condition. In fact, Pearson's chi-square test finds no significant difference between the takeaway histograms of the basic caption and the no-caption conditions (synthetic: $\chi^2(4) = 1.564$, $p = 0.815$; real-world: $\chi^2(4) = 7.168$, $p = 0.127$). While many automated captioning tools [159, 111] generate captions corresponding to our basic captions that do not describe specific features in the chart, we were unable to find evidence that these captions affect what people take away. Such captions may help readers with accessibility needs; however, we believe further exploration will help future systems determine appropriate uses for such captions.

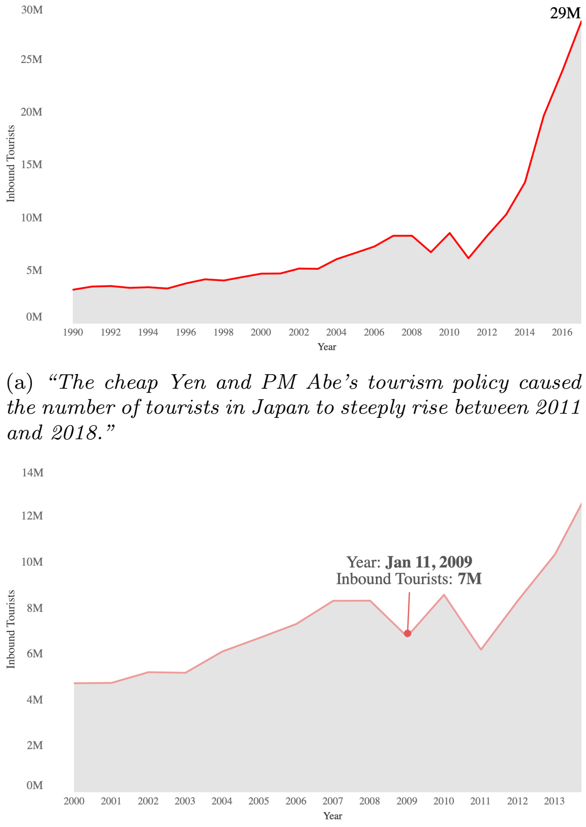
Assessing H2. To evaluate **H2**, we examine whether including external content information in the caption makes it more likely for readers to take away the feature mentioned in the caption. We find that people are significantly more likely to mention the feature described in the caption when it includes external information than when it does not (Figures 3.7e and Figures 3.7f *Any* bars). A pairwise Z-test finds significant difference between these ratios (synthetic: $Z = 2.273, p = 0.011$; real-world: $Z = 2.032, p = 0.021$). In addition, the reported reliance on the chart and the captions shifted towards the captions with external information, which is in-line with our findings (Figure 3.3 Block 3). Specifically, the reported reliance on the chart was significantly lower with external information (synthetic: Mann-Whitney $U = 137318$, $p < 0.001$; real-world: Mann-Whitney $U = 45292$, $p = 0.001$); the reported reliance on the caption was higher with external information, but the difference was only significant for the synthetic charts (synthetic: Mann-Whitney $U = 131594$, $p < 0.001$; real-world: Mann-Whitney $U = 48599.5$, $p = 0.132$).

The results together suggest that we can accept **H2** that states that including external information in the caption helps reinforce the message in the caption and users are more likely to take away from the feature described in the caption.

H2 Additional Results. Figure 3.7 (bottom row) breaks down the ratio of the takeaways that mention the feature described in the caption by level of prominence of the feature. The figure shows that there is usually an increase in the ratio of the takeaways that mentioned the feature described in the caption when the caption included external information for each level of prominence. Among the differences, we only found significant difference when the caption mentioned a non-prominent feature for synthetic charts ($Z = 3.027$, $p = 0.001$). Further study could shed light on the correlation between the prominence of the feature described in the caption and how external information affects the readers' takeaways.

3.4 Design Guidelines

Our findings indicate that the readers will take away from the feature doubly emphasized by both the chart and caption if they provide a coherent message. However, when the chart and caption



(a) *"The cheap Yen and PM Abe's tourism policy caused the number of tourists in Japan to steeply rise between 2011 and 2018."*

(b) *"Due to the 2008 Financial Crisis, the number of tourists in Japan decreased in 2009."*

Figure 3.8: Examples of chart-caption pairs authored to emphasize the same feature in the data. (a) Both the caption and chart emphasize the sharp positive trend. (b) The original chart is modified to zoom into a portion of the time range and the feature is made more visually prominent with an annotation showing the dip in the number of tourists. The caption describes that dip with additional context.

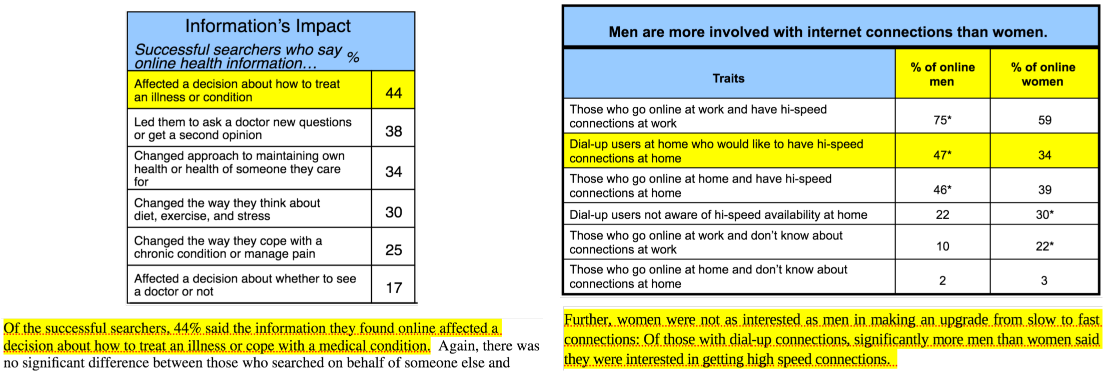
diverge in terms of the feature that they are emphasizing, readers are less likely to use information from the caption in their takeaways. To improve the efficacy of the chart-caption pair, authors could (1) design the chart to make the feature described in the caption more prominent and (2) include external information in the caption to give more context to the information in the caption.

There are several ways for authors to emphasize aspects of the data in a chart so that readers' attention is drawn to these visual features. One technique is to ensure that aspects of the data such as trends and outliers are presented at the right level of detail or interval range; too-broad of a measurement interval may hide a signal. For example, assume that we were given the chart in Figure 3.8a with the caption in Figure 3.8b. The decrease in 2009 is not very prominent because the large increase starting in 2011 overshadows the decrease. Zooming closer to the intended feature and cropping out irrelevant features (Figure 3.8b), helps make the feature more visually prominent. However, when zooming into the data in this manner, authors must take precaution to avoid removing important information or rendering the chart misleading [121, 125].

A simple way to further facilitate effective chart reading is to enhance the visualization with highlighting and graphical overlays such as annotations to guide the audience's attention to the image area they are describing [87] (Figure 3.8b). Sometimes, a different chart altogether may be more effective to emphasize a particular aspect of the data. For example, converting continuous data in line charts into discrete values could help emphasize individual values that the author would like to focus on. The consistency between the redesigned chart-caption pairs helps readers take away from the doubly emphasized feature (Figure 3.8).

Chapter 4

Linking Visualizations and Text[1]



Information's Impact *Successful searchers who say online health information…*	%
Affected a decision about how to treat an illness or condition	44
Led them to ask a doctor new questions or get a second opinion	38
Changed approach to maintaining own health or health of someone they care for	34
Changed the way they think about diet, exercise, and stress	30
Changed the way they cope with a chronic condition or manage pain	25
Affected a decision about whether to see a doctor or not	17

Of the successful searchers, 44% said the information they found online affected a decision about how to treat an illness or cope with a medical condition. Again, there was no significant difference between those who searched on behalf of someone else and

Men are more involved with internet connections than women.

Traits	% of online men	% of online women
Those who go online at work and have hi-speed connections at work	75*	59
Dial-up users at home who would like to have hi-speed connections at home	47*	34
Those who go online at home and have hi-speed connections at home	46*	39
Dial-up users not aware of hi-speed availability at home	22	30*
Those who go online at work and don't know about connections at work	10	22*
Those who go online at home and don't know about connections at home	2	3

Further, women were not as interested as men in making an upgrade from slow to fast connections: Of those with dial-up connections, significantly more men than women said they were interested in getting high speed connections.

Figure 4.1: Documents often include tables that provide evidence for arguments made in the main body text. In explicit references (left), the sentence text *"Of the successful searchers, 44% said the information they found online affected a decision about how to treat an illness or cope with a medical condition"* directly matches the text and numbers in the table cells (yellow highlights). In implicit references (right), the sentence text *"… Of those with dial-up connections, significantly more men than women said they were interested in getting high speed connections"* corresponds to row and column headers and readers must identify data cells at the intersection of two – i.e. the cells containing 47 and 34. Our interactive document reader automatically extracts such references for an input PDF document. Readers can click on a sentence to highlight the corresponding table cells and vice versa.

Document authors commonly use tables to support arguments presented in the text. But, because tables are usually separate from the main body text, readers must split their attention between different parts of the document. We present an interactive document reader that automatically links document text with corresponding table cells. Readers can select a sentence (or tables cells) and our reader highlights the relevant table cells (or sentences). We provide an automatic pipeline

[1] The contents of this chapter has been adapted from Kim et al. [83] (`https://doi.org/10.1145/3242587.3242617`). The thesis author was the first author and a major contributor to the work.

for extracting such references between sentence text and table cells for existing PDF documents that combines structural analysis of tables with natural language processing and rule-based matching. On a test corpus of 330 (sentence, table) pairs, our pipeline correctly extracts 48.8% of the references. An additional 30.5% contain only false negative (FN) errors – the reference is missing table cells. The remaining 20.7% contain false positive (FP) errors – the reference includes extraneous table cells and could therefore mislead readers. A user study finds that despite such errors, our interactive document reader helps readers match sentences with corresponding table cells more accurately and quickly than a baseline document reader.

4.1 Introduction

Data tables frequently appear in news articles, financial reports and scientific articles. For example, a news article may describe a trend, a relationship or a comparison in the text, and include a table that provides additional corroborating data. Fully understanding the document often requires mentally connecting and making sense of the text together with the corresponding table. In fact, previous work has shown that people can achieve much higher recall by jointly reading the text and tables in a journal article than by looking at the tables or the surrounding text alone [58].

Unfortunately, reading text together with a data table is challenging. As shown in Figure 4.1, the body text can contain explicit references (left), where the sentence text directly matches text in table cells or implicit references (right), where the sentence text matches the text in row and column header cells, but leaves it up to readers to identify the data cells at the intersection of the two. Moreover, readers must split their attention between the text and table and mentally integrate the two mutually dependent information sources. Such split-attention increases cognitive load [10, 49]. As a consequence, people often struggle to associate the text with the corresponding cells in the table, especially if the table is large and the text references multiple cells. Moreover, readers must break their flow and move their locus of attention from the main body text to the table and back again. The more time it takes to find the corresponding table cells, the more difficult it is to smoothly resume reading the main body text. Although text references and corresponding table cells are intended to be read together, many readers end up trying to make sense of them separately.

We present an interactive document reader designed to facilitate reading such documents and reduce split attention. Readers can select a sentence (or table cells) and our reader highlights the corresponding table cells (or sentences). We provide an automatic pipeline for extracting such references between body text and table cells for an input PDF document. After breaking the document into sentences and tables, our pipeline considers each (sentence,table) pair and operates in three main stages: (1) In the table structure extraction stage, it identifies the data type (e.g. text, number, percent, money, etc.) and cell type (title, header, or data) of each table cell. (2) It next matches sentence text to cells based on natural language processing (NLP) techniques. (3) Finally,

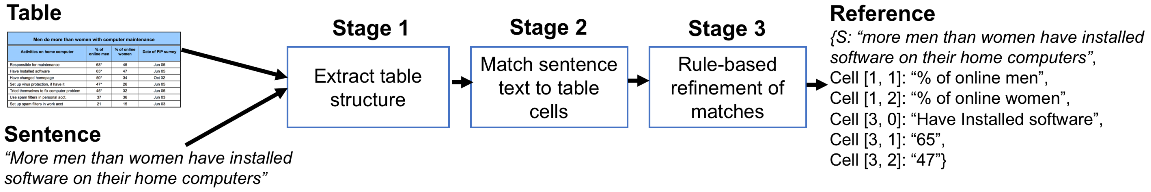


Table
Sentence
"More men than women have installed software on their home computers"
Stage 1
Extract table structure
Stage 2
Match sentence text to table cells
Stage 3
Rule-based refinement of matches
Reference
{S: "more men than women have installed software on their home computers",
Cell [1, 1]: "% of online men",
Cell [1, 2]: "% of online women",
Cell [3, 0]: "Have Installed software",
Cell [3, 1]: "65",
Cell [3, 2]: "47"}

Figure 4.2: Our automatic reference extraction pipeline. The input to the pipeline is a (sentence, table) pair and the output is a reference matching the sentence with a corresponding set of table cells it refers to. The reference includes both the cell indices (Figure 4.3a) and cell contents.

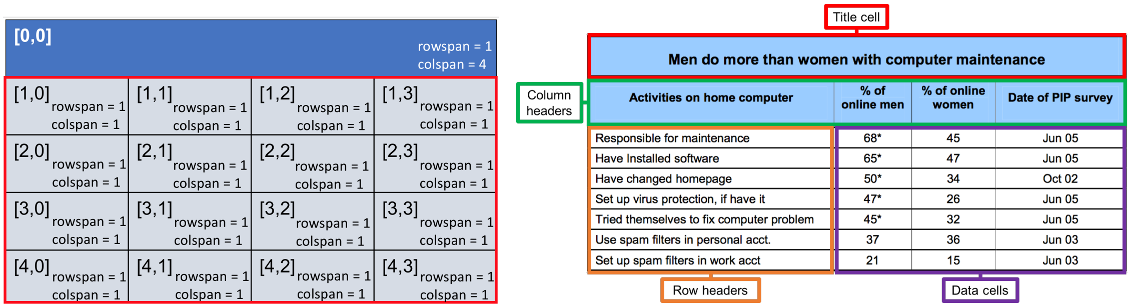
it applies rule-based refinement of the matches based on the table structure. For each sentence, the pipeline either outputs a reference consisting of one or more matching cells in the table, or it outputs a null reference if it cannot find such a match.

We compare our automatically generated references to human-generated gold standard references for a set of 330 (sentence, table) pairs gathered from a variety of source documents written for general audiences (e.g. Pew research reports [128], articles from the Economist magazine [41]) and for computer science researchers (e.g. ACL papers). Our pipeline correctly extracts 48.8% of the references. An additional 30.5% contain only false negative errors – the reference is incomplete and missing one or more table cells, while the remaining 20.7% contain false positive errors – the reference includes extraneous table cells and could therefore mislead readers.

We also conduct a controlled user study comparing our interactive document reader with a baseline reader that does not link text with tables. We find that despite the errors in reference extraction, when using our interface, participants match sentence text to table cells 26.4% more accurately and spend 22.9% less time than when using the baseline interface. These results suggest that automatically extracting and highlighting the links between document text and table cells reduces the split attention problem and facilitates reading the whole document. Our study also finds an asymmetry in the effects of FP errors and FN errors on this matching task. Participants are 23.2% less accurate and 27.8% slower at matching sentence text to table cells when the highlighted reference contains an FP error compared to highlighting a reference that contains only FN errors. This asymmetry suggests that FP errors are far more harmful to readers than FN errors because FP errors are misleading, while FN errors only omit information.

4.2 Automatic Reference Extraction

Our automatic reference extraction algorithm takes a PDF document as input and outputs references between each sentence in the text and cells in a table (Figure 4.2). The algorithm first breaks the main body text in the PDF into sentences using the Stanford CoreNLP toolkit [106]. It also identifies and extracts all the tables in HTML format using Adobe Acrobat Reader [2]. Our algorithm then



[0,0]
rowspan = 1
colspan = 4			

[1,0]
rowspan = 1
colspan = 1
	[1,1]
	rowspan = 1
	colspan = 1
		[1,2]
		rowspan = 1
		colspan = 1
			[1,3]
			rowspan = 1
			colspan = 1

[2,0]
rowspan = 1
colspan = 1
	[2,1]
	rowspan = 1
	colspan = 1
		[2,2]
		rowspan = 1
		colspan = 1
			[2,3]
			rowspan = 1
			colspan = 1

[3,0]
rowspan = 1
colspan = 1
	[3,1]
	rowspan = 1
	colspan = 1
		[3,2]
		rowspan = 1
		colspan = 1
			[3,3]
			rowspan = 1
			colspan = 1

[4,0]
rowspan = 1
colspan = 1
	[4,1]
	rowspan = 1
	colspan = 1
		[4,2]
		rowspan = 1
		colspan = 1
			[4,3]
			rowspan = 1
			colspan = 1

(a) Row and column indices and `span` values

Men do more than women with computer maintenance			
Activities on home computer	% of online men	% of online women	Date of PIP survey
Responsible for maintenance	68*	45	Jun 05
Have Installed software	65*	47	Jun 05
Have changed homepage	50*	34	Oct 02
Set up virus protection, if have it	47*	26	Jun 05
Tried themselves to fix computer problem	45*	32	Jun 05
Use spam filters in personal acct.	37	36	Jun 03
Set up spam filters in work acct	21	15	Jun 03

(b) Cell types (title, headers, data)

Figure 4.3: In Stage 1 of our pipeline, we first compute the spatial row and column indices [rowindex, colindex] as well as the rowspan and colspan values for each cell (a). Specifically, we set [rowindex,colindex] = [0,0] for the top left cell and process the HTML table tags from top to bottom, left to right, incrementing colindex or rowindex each time we encounter a `</td>` or `</tr>` tag respectively. We similarly annotate each cell with the HTML `<rowspan>` and `<colspan>` information. Later in Stage 1, we use this spatial information to identify the regular subgrid (red outline). We then classify each cell of the table as a title cell, header cell or data cell using the span and subgrid structure (b).

takes each (sentence, table) pair as input and applies a three stage pipeline to output either the set of cells the sentence refers to, or a null reference if there is no correspondence between the sentence and table.

4.2.1 Stage 1: Extract Table Structure

The first stage of our pipeline analyzes the input table to extract low-level information about the (1) spatial indices and spans, (2) data type (e.g. text, number, percentage, etc.) and (3) cell type (title, header, data) of each table cell. It also (4) normalizes the values held in each cell to a standardized format. Stages 2 and 3 of our pipeline use this low-level information to build the reference between the sentence and the table.

Compute spatial indices and spans of each cell

Given an input HTML table, we analyze the `<tr>` tags corresponding to each table row and `<td>` tags corresponding to each cell to generate row and column indices [rowindex, colindex] as well as the rowspan and colspan for each table cell (Figure 4.3a). The resulting indices encode the spatial position of the cells relative to one another and the spans indicate cells that span more than one row or column.

Identify data type of each cell

We classify the data type of each cell into one of six categories:

- ***Money:*** numeric value that represents an amount of money in some currency (e.g. "$10").
- ***Percent:*** numeric value that represents a percentage (e.g. "10%", "3 percent").
- ***Date:*** time value at granularity greater than one day (e.g. "1980/01/01", "April 2014").
- ***Time:*** time value at granularity finer than a day (e.g. "11:12", "11 o'clock", "15 sec").
- ***Number:*** numeric value that does not represent money, percent, date or time (e.g. a count, a rank).
- ***Text:*** text that does not fall into any other category.

To obtain this data type information, we apply the 7-class model of the Stanford Named Entity Recognizer [48] which labels each cell as either a Location, Organization, Person, Money, Percent, Date or Time. We ignore the Location, Organization and Person labels as the later stages of our pipeline do not need this information. We label any cells that do not fall into the Money, Percent, Date or Time categories as either Text or a Number depending on whether the cell contains a numeric value or it includes additional text.

Identify cell type of each cell

Tables commonly contain three types of cells (Figure 4.3b):

- ***Title cells*** are sometimes included as a part of a table and describe its overall contents.
- ***Header cells*** often appear at the top of columns (or left side of rows) and provide metadata describing the cells in the column (or row).
- ***Data cells*** appear in all tables as they hold the specific data values reported in the table.

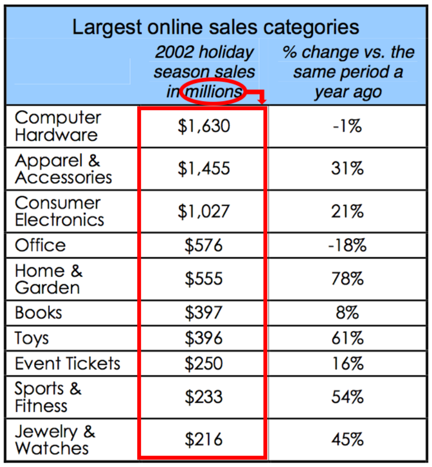
Classifying table headers and titles versus data cells is challenging as their formats can vary from document to document or even table to table within a document [45]. In fact, we analyzed a collection of example tables from a variety of PDF documents (newspaper articles, research papers, reports, etc.) and found that they use a variety of formats to distinguish titles and headers from data cells. In general, however, we found that for most tables the titles and headers appear in the topmost rows and/or the leftmost columns of the table and can sometime span multiple rows or columns. In contrast, data cells usually appear in the lower right part of the table and form a *regular grid* at the finest level of granularity (i.e. the cells do not span multiple rows or columns).

Based on these observations, we classify the cell type of each cell in the table in a two step process. First, we label any irregular rows or columns in the table – i.e. rows or columns that contain cells spanning more than one column or row, respectively (e.g. topmost row of Figure 4.3a). The remaining unmarked cells then form a regular grid (e.g. subgrid outlined in red in Figure 4.3a). We assume that the topmost row and leftmost column of this regular grid are headers at the finest level of granularity, and label all other cells within the regular grid as data cells. If the topmost header row contains a single cell that spans all the columns, we label it a title cell (Figure 4.3b).

Some tables do not contain row or column headers at the finest level of granularity. To properly handle such tables, we further rely on the assumption that the data type of header cells is often different from the data type of data cells. In many tables for example, header cells contain text while the data cells contain numbers. Therefore, we check if the cells in the topmost row and the leftmost column of the regular grid contain the same data type as the cells immediately below or to the right, respectively. If the data type is the same, we re-classify the finest granularity header cells as data cells.

Document authors sometimes leave data type information out of the data cells and only include it in the corresponding header cell. For example, the column header "% of online men" (Figure 4.3b) suggests that the data cells in the column are percentages, but the data cells only contain numeric values. To identify the data type of these cells, we parse header cells using a variety of common regular expressions (e.g. '% of', 'in $', 'in USD', etc.) and then propagate the data type information to the data cells in the corresponding columns or rows.

Normalize cell values



Largest online sales categories

	2002 holiday season sales in millions	*% change vs. the same period a year ago*
Computer Hardware	$1,630	-1%
Apparel & Accessories	$1,455	31%
Consumer Electronics	$1,027	21%
Office	$576	-18%
Home & Garden	$555	78%
Books	$397	8%
Toys	$396	61%
Event Tickets	$250	16%
Sports & Fitness	$233	54%
Jewelry & Watches	$216	45%

Figure 4.4: This table includes an order of magnitude term 'million' in a column header. In the normalization step, we propagate this magnitude to the data cells in the column.

Authors sometimes put the order of magnitude of data values (e.g. billions, millions, etc.) into a header so that the table remains concise (Figure 4.4). To identify such order of magnitude information we again parse the header cells using common order of magnitude expressions (e.g. 'billions', '(B)', 'mill', etc.) and propagate the information to the corresponding data cells.

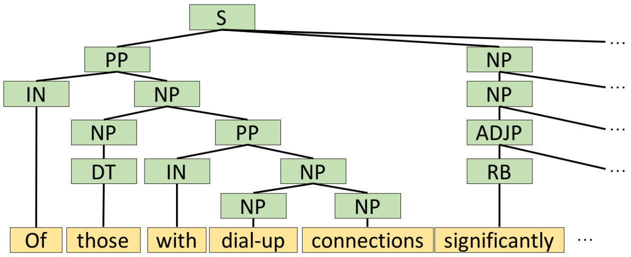


Figure 4.5: The phrase tree generated by the Stanford CoreNLP constituency parser for the sentence *"Of those with dial-up connections, significantly more men than women said they were interested in high-speed connections"*. Each subtree indicates a phrase and our syntactic matching algorithm finds a match between the phrase 'dial-up connections' and a cell in the table of Figure 4.1(right).

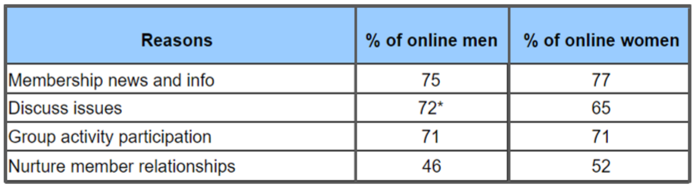
4.2.2 Stage 2: Match Sentence Text to Table Cells

In the second stage, we match the text of the input sentence to a corresponding set of cells in the input table using a combination of four different strategies using natural language processing (NLP) techniques. The first three strategies are designed to find matches against cells that contain text, while the fourth strategy is designed to find matches against cells that contain the other numeric data types.

Matching text cells based on unique words

Document authors often reference a specific table cell by including words in the sentence that uniquely appear in that cell and no other cell in the table. Consider the sentence *"However, mirroring the overall softness of the tech sector, sales of computer hardware decreased 1% versus a year-ago to $1.6 billion."* and the table in Figure 4.4. The terms 'computer' and 'hardware' appear in only one cell and it is likely that the sentence refers to it. We algorithmically implement this matching strategy by removing stop words from both the sentence and the table and lemmatizing the remaining words. Then for each cell, we store only the unique words, which do not appear in any other table cell. Finally, we match the remaining sentence words with the unique words in each table cell to identify a set of cells that the sentence is likely to be referencing.

Unfortunately, this simple matching strategy can produce incorrect references. Consider the sentence *"Of those who said they had virus protection on their home computers, significantly more men than women said they were responsible for setting up the protection"* and the table in Figure 4.3b. The word 'responsible' matches with the cell containing "Responsible for Maintenance", while the words 'virus' and 'protection' match with the cell containing "Set up virus protection, if have it". Only the second match is relevant to the sentence. We use syntactic and semantic analyses to better handle such cases.



Reasons	% of online men	% of online women
Membership news and info	75	77
Discuss issues	72*	65
Group activity participation	71	71
Nurture member relationships	46	52

Figure 4.6: The sentence *“Significantly more men than women think that talking about topics is an important reason to email with these special interest groups.”* matches the cell text “Discuss issues”, because the sentence phrase ‘talking about topics’ has the same meaning as the cell text even though they have no words in common. Our semantic similarity matching strategy detects this match.

.

Matching based on syntactic analysis

Syntactic analysis identifies the hierarchical phrase structure of a sentence as well as the grammatical dependencies between words. We use this syntactic structure to improve our matching algorithm. We first apply the constituency parser in the Stanford CoreNLP toolkit [106] to the input sentence to obtain its phrase tree (Figure 4.5). We then traverse the phrase tree breadth-first, starting at the root, and check if the entire phrase in the current subtree (after removing stop words and lemmatization) is uniquely contained within a single cell of the table – every word in the phrase must appear in exactly one cell. If such a unique cell exists, we add it to the reference.

Using this approach, we can identify references where individual sentence words do not uniquely match to a single table cell, but multi-word phrases do uniquely match. Consider the sentence *“Of those with dial-up connections, significantly more men than women said they were interested in high-speed connections”* with respect to the table in Figure 4.1(right). The words ‘dial-up’ and ‘connections’ appear separately in multiple table cells, but both words in the phrase ‘dial-up connections’ appear together in only one cell. Our syntactic analysis strategy identifies the matching cell correctly.

Matching based on semantic analysis

Sometimes, a sentence phrase and the words in a cell have the same meaning, but do not have any words in common. Consider the sentence *“Significantly more men than women think that talking about topics is an important reason to email with these special interest groups”* and the table in Figure 4.6. The sentence phrase ‘talking about topics’ has the same meaning as the cell containing “Discuss issues”, yet none of the words match and neither of our previous strategies would match them. To better handle such cases, we analyze the semantic similarity (i.e. similarity in meaning) between sentence phrases and the words in each cell.

We modify our syntactic matching strategy to compare each sub-phrase in the breadth-first traversal of the phrase tree using a distance based on *word2vec* – a vector model of words that encodes semantics. We use the pre-trained model of Mikolov et al. [113, 112] which was trained on parts of the Google News dataset [178] and produces vectors with 300 dimensions.

Specifically, we look up the word2vec vector for each word in the sentence phrase (after stop word removal and lemmatization) and sum them to generate a vector v_s representing the phrase. We similarly look up and sum the vectors for each word in the cell text to generate v_c, and then compute the cosine similarity between these vectors as

$$cos(\boldsymbol{v}_s, \boldsymbol{v}_c) = \frac{\boldsymbol{v}_s \cdot \boldsymbol{v}_c}{||\boldsymbol{v}_s|| \cdot ||\boldsymbol{v}_c||} \tag{4.1}$$

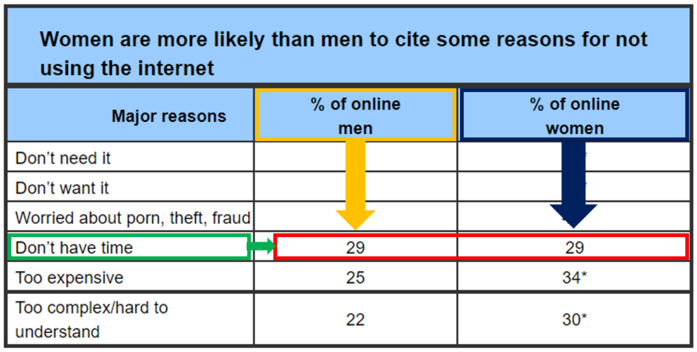
The word2vec model is designed so that the closer this cosine similarity is to 1, the greater the semantic similarity between the sentence phrase and the cell text. Therefore, whenever the cosine similarity between them is greater than a threshold τ (set empirically to 0.75), we treat them as a semantic match. This procedure correctly handles the example in Figure 4.6 because the semantic similarity between the sentence phrase 'talking about topics' and the cell text "Discuss issues" is above our semantic matching threshold τ.

Handling cells containing numeric and time values

To match sentence text to cells containing numeric values (i.e. numbers, percents or money), we first detect all strings in the sentence that represent numbers using the Stanford Named Entity Recognizer [48] and convert them into numerals (e.g. 'five million' is converted to 5,000,000).

Document authors often refer to table cells containing numeric values by rounding their rightmost significant digit. For instance, the sentence phrase *"about 1.5 meters"* may be used to refer to a table cell containing the value 1.53 meters. In some cases, the sentence phrase may also suggest the direction of rounding – either up or down. For example, the phrase *"more than 5 million"* may be used to refer a table cell containing the value 5,700,000. Thus, whenever we encounter a numeric value in the sentence text, we examine the surrounding words to check whether they indicate a rounding direction (up, down or nearest) – e.g. 'more than' indicates the value in the sentence has been rounded down. Then, if we do not find an exact match to the numeric value in the sentence, we compare the rounded value. As shown in Figure 4.4, this approach allows us to match the dollar amount in the sentence *"... sales of computer hardware decreased 1% versus a year-ago to \$1.6 billion"* to the topmost data cell in the second column containing "\$1,630".

We have found that document authors use a variety of formats to express dates (e.g. '01/01/1980' and 'Jan 01, 1980'), time (e.g. '14:02' and '2:02 PM'), and proportions (e.g. '20%' and '1 in 5'). To handle such variability, we detect dates, time and proportions within the sentence text using regular expression templates (e.g. 'dd/mm/yy', 'dd-mm-yyyy', 'hh:mm:ss', etc.) and normalize them to a standardized format so that our algorithm can correctly match equivalent expressions.



Women are more likely than men to cite some reasons for not using the internet

Major reasons	% of online men	% of online women
Don't need it		
Don't want it		
Worried about porn, theft, fraud		
Don't have time	29	29
Too expensive	25	34*
Too complex/hard to understand	22	30*

Figure 4.7: The sentence *"Equal numbers of men and women said they didn't have time"* implicitly refers to the data cells containing the value 29. However, the matching stage of our pipeline (Stage 2) only matches the row and column headers (green, yellow, blue outlines) to the sentence text. In Stage 3, we apply the *add implicit data cells rule* to correctly add in the implicit data cells (red outlines) to the reference.

4.2.3 Stage 3: Rule-based Refinement of Matches

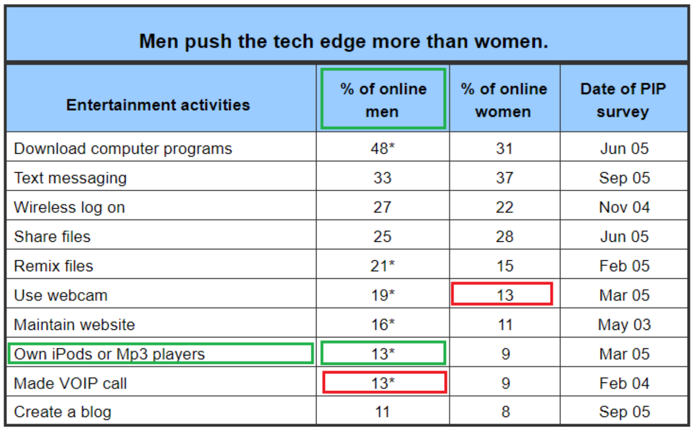
While many of the matches produced in Stage 2 are correct, because Stage 2 does not consider table structure (i.e. cell type – title, header, data of each cell), it can miss matches between the sentence and cells and it can incorrectly match the sentence to irrelevant cells. For instance, implicit references as in Figures 4.1(right) and 4.7, occur when a sentence only describes the row and column header cells in the text, leaving it up to the reader to identify the data cells that fall in the intersection of the these rows and columns. Our matching algorithm in Stage 2 would miss the matches to these data cells. The third stage of our pipeline is designed to handle such implicit references and also remove irrelevant matches based on table structure.

Rule 1: Add implicit data cells in the intersection of headers

To properly handle implicit references, our first rule considers all of the row and column headers returned by the matching algorithm in Stage 2 and automatically adds the data cells that fall in the intersection of the corresponding rows and columns to the set of matched cells. Applying this rule on the example in Figure 4.7 correctly adds the implicitly referenced data cells.

Rule 2: Remove data cells not in the intersection of headers

In some tables, the same value may appear in multiple data cells and if a sentence contains the value, our matching algorithm (Stage 2) identifies all such cells as a match to the sentence even though some of them may be irrelevant (Figure 4.8). But if the sentence also refers to the row and column headers, we can use the table structure to remove the irrelevant data cell matches. Our second rule only retains data cells that lie at the intersection of matched row and columns headers



Men push the tech edge more than women.

Entertainment activities	% of online men	% of online women	Date of PIP survey
Download computer programs	48*	31	Jun 05
Text messaging	33	37	Sep 05
Wireless log on	27	22	Nov 04
Share files	25	28	Jun 05
Remix files	21*	15	Feb 05
Use webcam	19*	13	Mar 05
Maintain website	16*	11	May 03
Own iPods or Mp3 players	13*	9	Mar 05
Made VOIP call	13*	9	Feb 04
Create a blog	11	8	Sep 05

Figure 4.8: The sentence *"In March 2005, 13% of men owned iPods or Mp3 players"*, matches with all the cells outlined in red and green after the matching stage (Stage 2) of our pipeline. However, only the cells with green outline are correct matches. In Stage 3, we remove the cells with red outline based on the rule that cells which do not appear in the intersection of row and column headers should be removed.

and eliminates all other data cells.

Rule 3: Add implicit header cells if data cells match uniquely

Header cells are sometimes referenced implicitly as well. Consider the sentence *"34% of women cited cost as the reason for not using the internet"* and the table in Figure 4.7. Our matching stage (Stage 2) matches the sentence text 'women' with the header cell "% of online women" and the sentence text '34%' with a unique data cell "34." But the sentence does not explicitly reference the row header cell "Too expensive" and our semantic matching strategy does not find a matching sentence phrase that is above its match threshold. We handle such implicit references to header cells by automatically adding the header cells whenever a single data cell matched uniquely within the table in Stage 2. In this case, since the single data cell "34" is uniquely matched, this rule allows us to correctly include the row header cell "Too expensive."

Rule 4: Remove potentially irrelevant header cells

In some cases, our matching algorithm in Stage 2 finds matches between the sentence text and row and column header cells, but the sentence also contains numeric data values that do not appear in the table. Consider the example sentence *"58% of men and 47% of women said they know how to upload images or other files to a website so others could see them"* with respect to the table in Figure 4.8. In Stage 2, we obtain matches to the columns "% of online men" and "% of online women". However, the numeric data values given in the sentence 58% and 47% do not appear in any of the table cells. In such cases, our fourth rule removes the header cells based on the assumption that the sentence is unlikely to be related to the table. In this case, the rule removes "% of online men" and "% of online women" from the reference.

4.3 Pipeline Evaluation

Figures 4.1, 4.9 and 4.12 show references generated using our automatic reference extraction pipeline. To quantitatively evaluate the accuracy of our automatic reference pipeline, we gathered a representative sample of (sentence, table) pairs from documents written for general audiences as well as scientific papers written for researchers. We obtained a gold reference set for each pair and then compared the results from our reference extraction pipeline to the gold reference set.

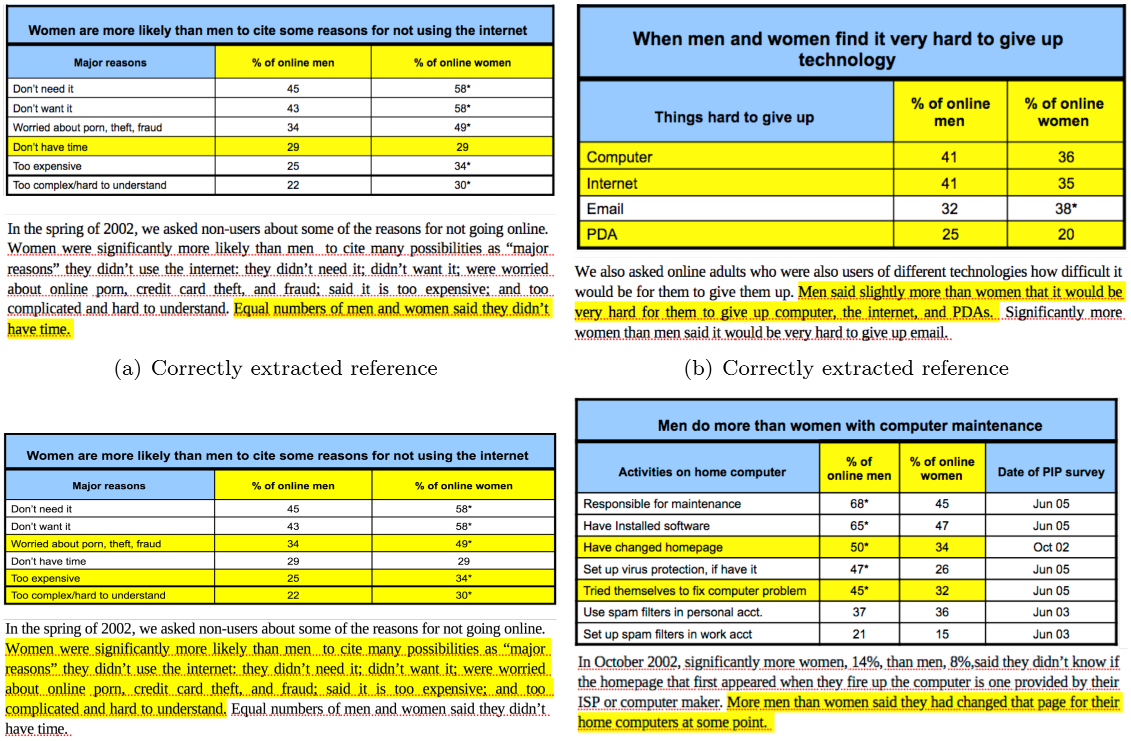
4.3.1 Corpus

To build the representative sample of (sentence, table) pairs, we gathered two sets of PDF documents written for different audiences. Our *Pew* dataset contains 10 research reports written for general audiences and published by Pew Research [128] in the area of public policy. Our *Academic* dataset contains 6 research papers written for computer science researchers from the ACL conference [29, 54, 62, 89, 101, 171]. Since most sentences in a document are unrelated to any table within it, we manually identified tables as well as paragraphs related to these tables from the corpus. Thus, we could ensure that many of the sentences would reference the tables, but since we took entire paragraphs, we could also be sure that some sentences would not reference the tables. Table 4.1 summarizes the number of tables, paragraphs and (sentence, table) pairs we extracted for each dataset.

For comparison, we include a third dataset from Kong et al. [88] that contains (sentence, table) pairs from 18 general audience documents including news sources like the Economist [41] and the Guardian [61]. Together, the documents in our datasets cover a range of writing styles and table usages.

Dataset	# Docs	# Tables	# Paras	# (sentence, table) pairs
Pew	10	26	35	127
Academic	6	11	14	72
Kong [88]	18	35	49	139

Table 4.1: Summary of the three datasets we use to evaluate our pipeline. The *Pew* and *Kong* datasets are culled from documents written for general audiences while the *Academic* dataset is from computer science research papers.



Women are more likely than men to cite some reasons for not using the internet		
Major reasons	% of online men	% of online women
Don't need it	45	58*
Don't want it	43	58*
Worried about porn, theft, fraud	34	49*
Don't have time	29	29
Too expensive	25	34*
Too complex/hard to understand	22	30*

In the spring of 2002, we asked non-users about some of the reasons for not going online. Women were significantly more likely than men to cite many possibilities as "major reasons" they didn't use the internet: they didn't need it; didn't want it; were worried about online porn, credit card theft, and fraud; said it is too expensive; and too complicated and hard to understand. Equal numbers of men and women said they didn't have time.

(a) Correctly extracted reference

When men and women find it very hard to give up technology		
Things hard to give up	% of online men	% of online women
Computer	41	36
Internet	41	35
Email	32	38*
PDA	25	20

We also asked online adults who were also users of different technologies how difficult it would be for them to give them up. Men said slightly more than women that it would be very hard for them to give up computer, the internet, and PDAs. Significantly more women than men said it would be very hard to give up email.

(b) Correctly extracted reference

Women are more likely than men to cite some reasons for not using the internet		
Major reasons	% of online men	% of online women
Don't need it	45	58*
Don't want it	43	58*
Worried about porn, theft, fraud	34	49*
Don't have time	29	29
Too expensive	25	34*
Too complex/hard to understand	22	30*

In the spring of 2002, we asked non-users about some of the reasons for not going online. Women were significantly more likely than men to cite many possibilities as "major reasons" they didn't use the internet: they didn't need it; didn't want it; were worried about online porn, credit card theft, and fraud; said it is too expensive; and too complicated and hard to understand. Equal numbers of men and women said they didn't have time.

(c) Extracted reference containing FN errors

Men do more than women with computer maintenance			
Activities on home computer	% of online men	% of online women	Date of PIP survey
Responsible for maintenance	68*	45	Jun 05
Have Installed software	65*	47	Jun 05
Have changed homepage	50*	34	Oct 02
Set up virus protection, if have it	47*	26	Jun 05
Tried themselves to fix computer problem	45*	32	Jun 05
Use spam filters in personal acct.	37	36	Jun 03
Set up spam filters in work acct	21	15	Jun 03

In October 2002, significantly more women, 14%, than men, 8%,said they didn't know if the homepage that first appeared when they fire up the computer is one provided by their ISP or computer maker. More men than women said they had changed that page for their home computers at some point.

(d) Extracted reference containing FP errors

Figure 4.9: References extracted by our automatic reference extraction pipeline. (a) Correctly extracted reference for the sentence "*Equal numbers of men and women said they didn't have time.*" (b) Correctly extracted reference for the sentence "*Men said slightly more than women that it would be very hard for them to give up computer, the internet, and PDAs.*" (c) Reference containing false negative errors (missing cells) for the sentence "*Women were significantly more likely than men to cite many possibilities as "major reasons" they didn't use the internet: they didn't need it; didn't want it; were worried about online porn, credit card theft, and fraud; said it is too expensive; and too complicated and hard to understand.*" Because the pipeline removes the stop words 'do', 'not', 'need', 'want', and 'it', it misses the header rows "Don't need it" and "Don't want it." (d) Reference containing false positive errors (includes irrelevant cells) for the sentence "*More men than women said they had changed that page for their home computers at some point.*" Our pipeline detects an extra row because the word 'computer' appears in the sentence and in a single cell "Tried themselves to fix computer problem."

4.3.2 Gold Reference Set

We used an iterative process to create a gold reference set for the (sentence, table) pairs in the *Pew* and *Academic* datasets. First, two authors from our research team independently identified references between the (sentence, table) pairs following the reference annotation guidelines of Kong et al. [88]. They then resolved each inconsistency by explaining their logic in producing the reference. They then worked together to develop a consensus reference. Finally, a third author scrutinized the

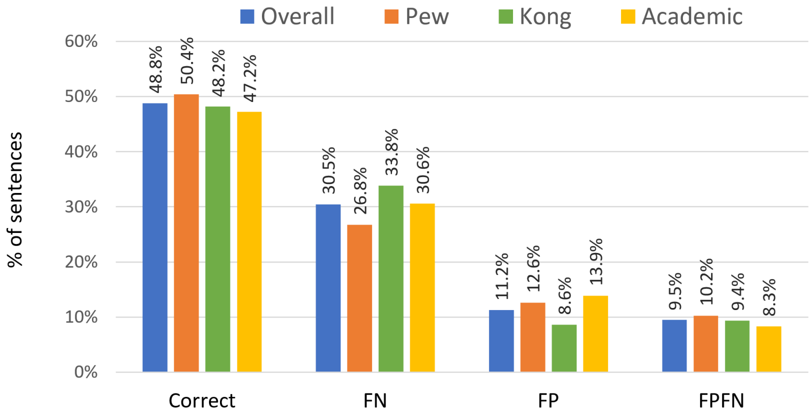


Figure 4.10: Comparison of correct, FN, FP and FPFN references produced by our complete pipeline for each of the datasets, Pew (orange), Kong (green) and Academic (yellow) as well as the overall combination of all three datasets (blue).

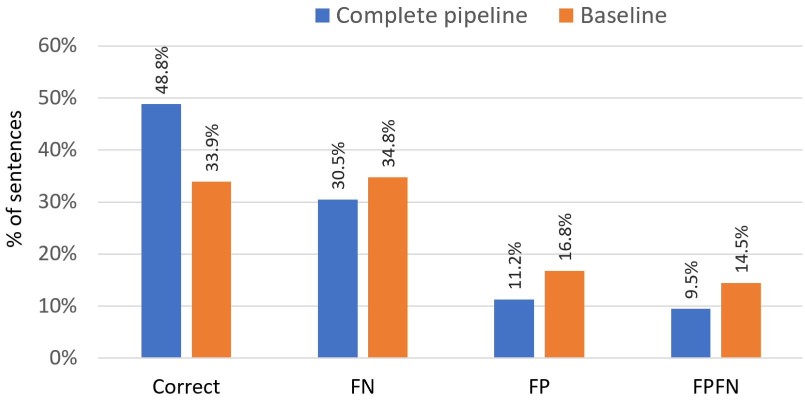


Figure 4.11: Comparison of correct, FN, FP and FPFN references produced by our complete pipeline and a baseline method using only the unique words matching strategy combining all three datasets.

resulting references and initiated a second round of debate for each reference that he disagreed with. After a thorough discussion between all three authors, they reached a final consensus about the set of cells to include as the gold reference set for each sentence. For the *Kong* dataset, we used the gold references provided by Kong et al. [88].

4.3.3 Pipeline Performance and Accuracy

Across all three datasets, our reference extraction pipeline took an average of 258.38 ms to process each (sentence, table) pair on a 2.5Ghz MacBook Pro with an Intel Core i7 processor and 16GB RAM. Stage 1 took an average of 233.89 ms per table, Stage 2 took 208.14 ms per sentence, and Stage 3 took 0.42 ms per sentence.

To compute the accuracy of our pipeline, we compare the results it generates to the gold references. Specifically, for each sentence, we compare our automatically generated reference A to the corresponding gold reference G and categorize the results as follows:

- **Correct reference:** our pipeline generates the exact same set of table cells as in the gold reference, i.e. $G = A$.
- **False negative (FN):** our pipeline generates a reference that is missing some cells that are included in the gold reference, i.e. $A \subset G$.
- **False positive (FP):** our pipeline generates a reference that includes extraneous cells that are not in the gold reference, i.e. $G \subset A$.
- **False positive + False negative (FPFN):** our pipeline generates references with both false negatives and false positives, i.e. $G \not\subset A$ and $A \not\subset G$.

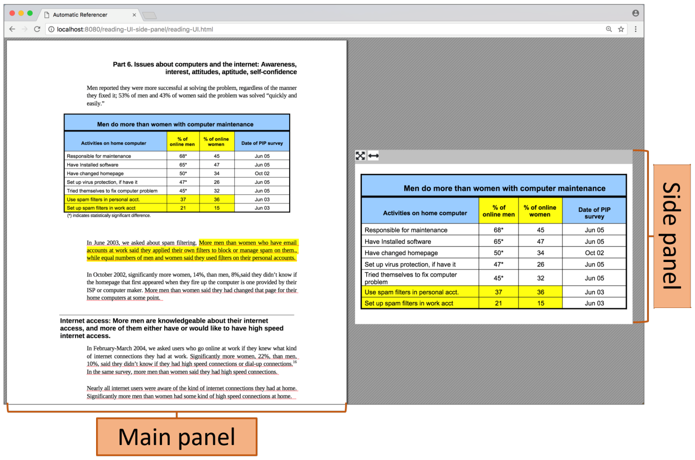
As shown in Figure 4.10, we find that overall (blue bars) across all three datasets, our complete pipeline generates 48.8% correct references, 30.5% references that contain only FN errors, 11.2% references that contain only FP errors, and 9.5% references that contain both FN and FP errors. Moreover, the accuracy numbers are similar across the three datasets despite the fact that they contain different kinds of writing meant for different audiences. This result suggests that the performance of our pipeline is somewhat independent of writing style.

For the *Kong* dataset, we also compare our pipeline with the crowdsourcing pipeline of Kong et al. [88] that combines references generated by multiple workers into a single set, using clustering and merging techniques. Their approach produces 71.2% correct references, 8.6% FN errors and 18.8% FP errors and 1.4% FPFN errors. While their crowdsourcing pipeline produces 22.4% more correct references than our automatic pipeline, their increase in accuracy comes at the cost of significantly more annotation effort as they require multiple crowd workers to independently generate references for each (sentence, table) pair.

Figure 4.11 compares the accuracy of our reference extraction pipeline to a baseline version of our pipeline that only includes the matching on unique words strategy and does not include other strategies in Stage 2 or the rule-based refinements of Stage 3. This comparison shows that the complete pipeline with the syntactic and semantic matching, as well as the rule-based refinement, provides a substantial improvement in the percentage of correct references over the baseline.

4.4 Interactive Document Reader

The goal of our interactive document reader (Figure 4.12) is to assist viewers by displaying references between the document text and the tables as they read the document. Given a PDF document with a set of such references, our reader underlines each sentence that references a table in red. Clicking



Part 6. Issues about computers and the internet: Awareness, interest, attitudes, aptitude, self-confidence

Men reported they were more successful at solving the problem, regardless of the manner they fixed it; 53% of men and 43% of women said the problem was solved "quickly and easily."

Men do more than women with computer maintenance			
Activities on home computer	% of online men	% of online women	Date of PIP survey
Responsible for maintenance	68*	45	Jun 05
Have Installed software	65*	47	Jun 05
Have changed homepage	50*	34	Oct 02
Set up virus protection, if have it	47*	26	Jun 05
Tried themselves to fix computer problem	45*	32	Jun 05
Use spam filters in personal acct.	37	36	Jun 03
Set up spam filters in work acct	21	15	Jun 03

(*) indicates statistically significant difference.

In June 2003, we asked about spam filtering. More men than women who have email accounts at work said they applied their own filters to block or manage spam on them., while equal numbers of men and women said they used filters on their personal accounts.

In October 2002, significantly more women, 14%, than men, 8%,said they didn't know if the homepage that first appeared when they fire up the computer is one provided by their ISP or computer maker. More men than women said they had changed that page for their home computers at some point.

Internet access: More men are knowledgeable about their internet access, and more of them either have or would like to have high speed internet access.

In February-March 2004, we asked users who go online at work if they knew what kind of internet connections they had at work. Significantly more women, 22%, than men, 10%, said they didn't know if they had high speed connections or dial-up connections.[16] In the same survey, more men than women said they had high speed connections.

Nearly all internet users were aware of the kind of internet connections they had at home. Significantly more men than women had some kind of high speed connections at home.

Men do more than women with computer maintenance			
Activities on home computer	% of online men	% of online women	Date of PIP survey
Responsible for maintenance	68*	45	Jun 05
Have Installed software	65*	47	Jun 05
Have changed homepage	50*	34	Oct 02
Set up virus protection, if have it	47*	26	Jun 05
Tried themselves to fix computer problem	45*	32	Jun 05
Use spam filters in personal acct.	37	36	Jun 03
Set up spam filters in work acct	21	15	Jun 03

Figure 4.12: Our interactive document reader contains a main panel showing the document and a side panel showing the table most relevant to the sentences at center of the main panel. Red underlines indicate sentences that refer to cells in a table. Clicking on such a sentence highlights it and the table cells it refers to in yellow. Clicking on a cell highlights all sentences that refer to it. Clicking anywhere else on the document removes the highlight.

on such a sentence highlights it and the table cells it refers to in yellow. Similarly, clicking on a cell highlights all the sentences that refer to it. Clicking anywhere else on the document removes the highlight.

To reduce the problem of split attention that occurs when a table is located relatively far away in the document from the referencing text, we include a side panel that replicates the table most relevant to the sentence at the center of the main panel. As the user scrolls through the pages in the main panel, the most relevant table in the side panel automatically updates. The table is scaled by default to fit in the panel, but clicking the expand button expands the table to full size. The table is a fully interactive copy of the table in the main panel and clicking on a cell in either table highlights the relevant sentences in the document and vice versa.

4.5 User study

We conducted a user study to compare our interactive document reader with automatic linking of sentences to table cells to a baseline reader (similar to Adobe Reader) that does not provide such links. We consider two main hypotheses:

H1: Despite the errors produced in our automatic reference extraction pipeline, our interactive document reader will help users locate table cells relevant to sentences in the text more accurately and quickly than the baseline reader.

H2: Since false positive (FP) errors can mislead readers by connecting sentences to incorrect table cells, they will cause more harm (lower speed and accuracy) than false negative (FN) errors which simply force readers to manually identify the connection between sentences and table cells.

4.5.1 Study Design

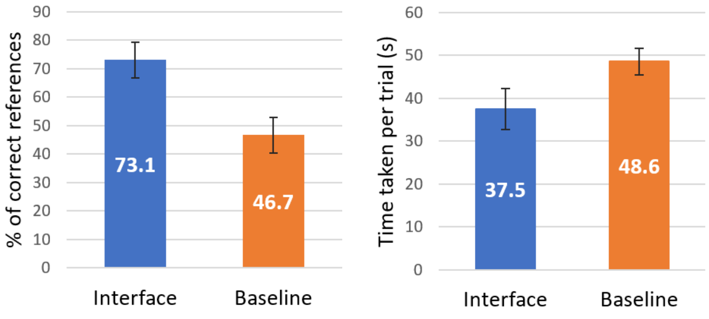
We used a within-subjects study design. We sampled two groups of 12 (sentence, table) pairs from our corpus such that the distribution of error types (correct, FN, FP, FPFN) in each group roughly matched the overall distribution produced by our automatic reference extraction pipeline (6 correct, 4 FN, 1 FP, 1 FPFN). We used the first group of references to generate 12 *interface condition* trials and the second group to generate 12 *baseline condition* trials. Each trial presented a single (sentence, table) reference pair, where the sentence was underlined in red and the paragraph containing the sentence was shown for context. The interface condition included the interactive reference highlighting of our interactive document reader, while the baseline condition did not include such highlighting. On each trial, the participant had to select the table cells referenced by the underlined sentence.

4.5.2 Study Procedure

We recruited 14 adult participants, all fluent in English, from three academic institutions. Each participant completed 24 trials, 12 in each condition. We counterbalanced the ordering of the conditions and randomized the ordering of trials within each condition for each participant to reduce ordering effects. Before running the experiment, participants went through a training session where they learned how to use both conditions and correctly complete the trial task. During the experiment, we measured the accuracy and speed of each trial. The participants were aware that we were measuring both time and accuracy, but we did not specifically ask them to prioritize speed or accuracy. After completing all 24 trials, we asked the participants to rate the helpfulness of our interface on a 5 point Likert scale and to express their opinions about the usefulness of the interface in a free-form text response. The experiment took about 45 minutes to complete and each participant received a $20.00 Amazon gift card for participating in the study.

4.5.3 Results

We find that our interactive reading interface significantly outperforms the baseline interface in terms of accuracy and speed (Figure 4.13). Accuracy in the interface condition ($\mu = 73.1\%$, $\sigma = 23.61$) was 26.4% higher than in the baseline condition ($\mu = 46.7\%$, $\sigma = 19.67$, $t(13) = 7.57$, $p < 0.001$). On average it took participants 22.9% (11.13 seconds) less time in the interface condition ($\mu = 37.5$s, $\sigma = 11.69$) than in the baseline condition ($\mu = 48.6$s, $\sigma = 17.96$, $t(13) = 4.12$, $p < 0.05$). In their subjective assessments of helpfulness of our interface for reading documents (on a 5 point scale with 5 = very helpful), participants were generally positive ($\mu = 4.1$, $\sigma = 0.17$). In the free-form response,



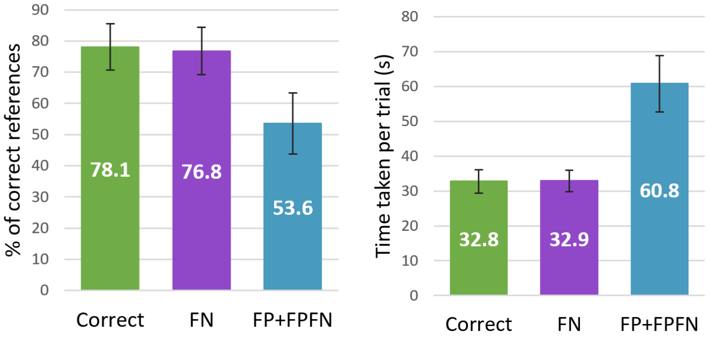
(a) Accuracy (b) Time

Figure 4.13: Users are significantly more accurate and faster at matching sentence text to table cells using our interactive reference highlighting interface compared to a baseline interface. These results indicate that despite errors introduced by our reference extraction pipeline, our interface facilitates document reading overall.

one of the participants who had the interface condition first commented that after the transition to the baseline condition, the increased amount of effort required to complete each trial was noticeable. Together, these results suggest that we can accept hypothesis H1.

We also find that presenting a reference containing FP errors in our interface harms accuracy and speed much more than presenting a reference containing only FN errors (Figure 4.14). Specifically, a one-way RM-ANOVA finds a significant effect for the types of references presented (Correct, FN – contains only FN errors, FP+FPFN – contains at least one FP error) on accuracy ($F(2, 26) = 4.89$, $p < 0.05$). Participants suffered a 24.5% hit in accuracy when the presented reference contained an FP error ($\mu = 53.6\%$, $\sigma = 36.50$) compared to when the presented reference was correct ($\mu = 78.1\%$, $\sigma = 27.57$, $t(13) = 3.061$, $p < 0.005$). Similarly with FP errors they suffered a 23.2% reduction in accuracy compared to when the reference contained only FN errors ($\mu = 76.8\%$, $\sigma = 28.53$, $t(13) = 2.061$, $p < 0.05$). These results suggest that FP errors are far more harmful to accuracy than FN errors. In fact, we found that when comparing accuracy for references containing only FN errors to correct references, there is no significant difference.

We find similar results for speed. A one-way RM-ANOVA finds a significant difference in time required to complete the trial for the three types of references presented ($F(2, 26) = 11.815$, $p < 0.001$). Participants took 28.0 seconds longer per trial when shown references containing FP errors ($\mu = 60.8$, $\sigma = 30.39$) than when shown correct references ($\mu = 32.8$, $\sigma = 12.59$, $t(13) = 3.372$, $p < 0.005$). Similarly, they took 27.8 seconds longer when shown references with FP errors than when shown references containing only FN errors ($\mu = 32.9$, $\sigma = 11.50$, $t(13) = 3.707$, $p < 0.005$). Moreover, we saw no significant drop in the speed between being shown correct references and being shown references with FN errors. Together these accuracy and speed results suggest that we can also accept hypothesis H2.



(a) Accuracy (b) Time

Figure 4.14: Comparison of user accuracy in matching sentence text to table cells when our interactive interface presents references that are correct, that contain FN errors only and that contain at least one FP error. We find that FP errors are significantly more harmful (reduce accuracy and speed) than FN only errors and that the difference between presenting correct references and FN only is not significant.

4.6 Discussion

Our main goal in developing our interactive document reader was to reduce split attention when reading documents containing tables. Our user study finds that users can match document text to table cells more accurately and quickly using our interface than they can using a standard baseline document reader. This result indicates that our interface does reduce the split attention problem.

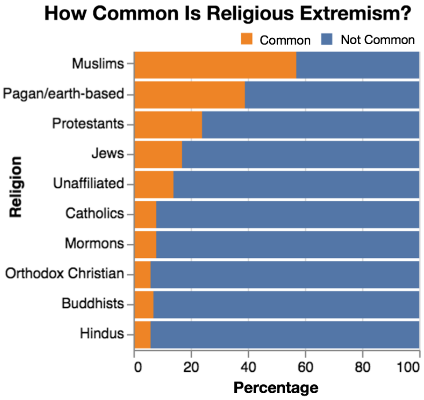
In addition to the controlled user study, we have observed a number of users as they interact with our document reader interface. They all agreed that our interface was easy to use and that they found the link connecting sentence text to table cells to be useful. One compared our interface to standard document viewers saying, "*The interface allows me to read the table while reading the text. Originally this was done in the text-then-table order, but this was parallelized, making it more efficient.*" Another said, "*In everyday life, if text includes tables, I would usually trust the text and not read the table too carefully, but this interface made me take time looking back and forth.*" These observations also suggest that our interface reduces split attention and facilitates document reading.

From the user study, we also found that presenting references containing FP errors is more harmful than presenting references containing only FN errors. We believe that this is because FP errors can actively mislead readers by matching text sentences to irrelevant table cells. In contrast, FN errors simply force readers to manually identify the connection between sentence text and table cells. In the free-form text response, one of the study participants wrote "*I would prefer to have missing information [than to have extra information] because I can always fill in the gaps.*" Being misinformed (FP errors) is much worse than being uninformed (FN errors).

Another implication of this finding is that while there is some room for improvement in the percentage of correct references produced by our pipeline, 48.8%, it may be best to focus future work on reducing the FP error rate of 20.7% before addressing the FN errors. Moreover, as our user study shows, extracting references between text and tables is challenging even for people. In the baseline condition, participants produced 46.7% correct references ($\sigma = 19.67$) on average, suggesting that our pipeline produces correct references at rates that are comparable to human performance.

Chapter 5

Visual Explanations for Chart Question Answering[1]



Q1: *What is the percentage of response ‘Common’ for Catholics?*

A(Sempre): ***92***
A(Ours): ***8****. I looked up the length of the orange bar for ‘Catholics’.*

Q2: *Which religion has the longest orange component?*

A(Sempre): ***Hindus***
A(Ours): ***Muslims****. I looked up ‘Religion’ of the longest orange bar.*

Q3: *What does the blue field represent?*

A(Sempre): ***24***
A(Ours): ***Not Common****. I looked up what blue represents by looking at the legend.*

Figure 5.1: Questions about a chart from a Pew research report [128]. Q1 requires a value lookup on the data in the chart and Q3 requires a lookup on the legend. Q2 is compositional as it requires multiple operations including value lookup and comparisons. Our automatic chart question answering pipeline answers all three questions correctly (marked in green) and gives correct explanations of how it obtained the answer, whereas Sempre [127, 187], a state-of-the-art table question answering system, gets all three wrong (marked in red).

People often use charts to analyze data, answer questions and explain their answers to others. In a formative study, we find that such human-generated questions and explanations commonly refer to visual features of charts. Based on this study, we developed an automatic chart question answering pipeline that generates visual explanations describing how the answer was obtained. Our pipeline

[1] The contents of this chapter has been adapted from Kim et al. [82] (`https://doi.org/10.1145/3313831.3376467`). The thesis author was the first author and a major contributor to the work.

first extracts the data and visual encodings from an input Vega-Lite chart. Then, given a natural language question about the chart, it transforms references to visual attributes into references to the data. It next applies a state-of-the-art machine learning algorithm to answer the transformed question. Finally, it uses a template-based approach to explain in natural language how the answer is determined from the chart's visual features. A user study finds that our pipeline-generated visual explanations significantly outperform in transparency and are comparable in usefulness and trust to human-generated explanations.

5.1 Introduction

Using visualizations to analyze data, answer questions, and explain how the answer was obtained, is at the heart of many decision-making tasks. However, performing such complex analytical tasks with visualizations is not always easy. Users often need to answer compositional questions that require combining multiple complex operations such as retrieving a value from the chart, finding extreme values, comparing and aggregating values, or calculating sums and differences of values. Consider the bar chart in Figure 5.1 and the question *"For which religion did the most chaplains think that religious extremism is common?"* To answer this question, users need to visually compare the values represented by orange bars, find the longest one and then lookup the corresponding religion; in this case it is *'Muslims'*.

As users are analyzing a chart, they regularly pose questions by referring to visual features of the chart including the graphical marks (e.g. bars) and their data encoding visual attributes (e.g. width) [78, 144, 146]. For example, in the course of analyzing the bar chart in Figure 5.1, a user might ask *"Which religion has the longest orange component?"* This is a visual version of our earlier question, and while it remains compositional, because it references visual features of the chart, it is shorter and more directly suggestive of the operations users must perform to answer it. Nevertheless, answering such compositional questions, whether they are visual or non-visual, can be time-consuming and mentally taxing as users must perform multiple complex operations.

To obtain a better insight into how people naturally ask questions about charts, we conducted a formative study in which we collected 629 human-generated questions for 52 real world charts along with 748 human-generated explanations. We then categorized the questions along two orthogonal dimensions; (1) *lookup* (i.e. requiring a single value retrieval), or *compositional* (i.e. requiring multiple operations) and (2) *visual* (i.e. referencing visual chart features) or *non-visual.* We find that people frequently ask compositional questions (70%), regularly ask visual questions (12%), and that visual explanations are especially common (51%).

Can we design a tool to automatically answer such natural language questions about charts? Automatic question answering would benefit users in several ways. It would significantly reduce the time and mental effort by performing complex operations such as retrieval, comparison and aggregation (sum, average) on behalf of users. Such a tool could quickly and accurately retrieve data values from visual attributes that are perceptually difficult to decode (e.g. size, brightness). More importantly, introducing a natural language interface into the data analysis workflow would lower the threshold of ability required to analyze data using charts and graphs. It could enable people who have not been formally trained in data analysis tools and visualization literacy to get answers to their questions. However, for users to rely on such an automated tool, it is critical that the tool be able to transparently explain how it obtains the answers [144]. Moreover, our formative study suggests that the most effective explanations are visual because they describe how the answer is extracted from the visual features of the chart. Yet, no previous work on automatic question answering for charts [26, 76, 77, 78, 133] has provided explanations for their answers.

In this chapter, we present an automatic pipeline for answering natural language questions about charts and generating such visual explanations. Our approach builds on Sempre [127, 187], a question-answering system for relational data tables that focuses on answering compositional, non-visual questions. We significantly extend Sempre to answer questions about charts and also generate corresponding visual explanations. Our pipeline works with both lookup and compositional questions as well as visual and non-visual questions. The key idea of our approach is to take advantage of the visual data encoding structure of an input chart in Vega-Lite [139] format – a programmatic representation that explicitly describes the encodings that map data to mark attributes – to accurately answer visual questions and to generate the visual explanations.

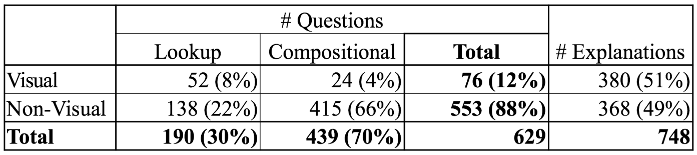
We evaluate our question-answering pipeline on the corpus of 629 chart-question pairs we gathered in our formative study. We find that our pipeline correctly answers 51% of all the questions in our corpus, while Sempre alone can only answer 39% of the questions correctly, a difference of 12%. For visual questions, our improvement is even larger at 53% and even for non-visual questions, our pipeline outperforms Sempre by 6%. Overall, these results suggest that information about the visual encoding structure of a chart is very useful for automatic chart question answering. Finally, we conduct a user study which finds that our pipeline-generated visual explanations are significantly more transparent than human-generated explanations while remaining comparable in usefulness and trust.

Our code and data are available at:

`https://github.com/dhkim16/VisQA-release`

5.2 Formative Study

To learn how people naturally ask questions, extract answers and explain their answers when they encounter charts, we conducted a formative study. We gathered a corpus of charts from multiple real-world sources, and asked crowdworkers to write natural language questions, provide answers



	# Questions			
	Lookup	Compositional	**Total**	# Explanations
Visual	52 (8%)	24 (4%)	**76 (12%)**	380 (51%)
Non-Visual	138 (22%)	415 (66%)	**553 (88%)**	368 (49%)
Total	**190 (30%)**	**439 (70%)**	**629**	**748**

Table 5.1: Counts and percentages of the types (lookup/compositional, visual/non-visual) of natural language questions and explanations crowdworkers generated for our set of 52 charts.

and explain their answers. We then manually analyzed the resulting data to understand (1) how often people ask lookup and compositional questions and (2) how often they refer to visual features for the charts in their questions and explanations.

5.2.1 Gathering Charts, Questions, Answers and Explanations

Our corpus includes 52 charts, gathered from four different sources; (1) the Vega-Lite Example Gallery [168], (2) charts in Pew Research Reports as collected by Kong et al. [88], (3) D3 charts we found across the Web, and (4) charts constructed from tables found in the WikiTableQuestions dataset [127]. In total, our corpus includes 47 bar charts (32 simple, 8 grouped, 7 stacked) and 5 line charts. We focus on these two chart types because, as Battle et al. [14] have shown, they are two of the most common types of charts available on the Web.

We asked crowdworkers from Amazon Mechanical Turk to consider a single chart, write 5 natural language questions about it, answer 10 questions about it including their own and provide explanations for their answers. We then manually reviewed the responses and removed questions that were not answerable from the chart, as well as explanations that carried no information about how the worker obtained the answer from the chart (e.g. *"I got it from the chart"*). This process generated a total of 629 questions, 866 answers and 748 explanations for the 52 charts.

5.2.2 Analysis

We analyzed the crowdworker responses to differentiate compositional questions from lookup questions as well as visual versus non-visual questions and explanations (Table 5.1).

We find that 70% of the questions are compositional, while the remaining 30% are lookups. Compositional questions often ask about extrema (38%), differences between two data values (22%), and the sum of multiple values (7%). An additional 12% of the compositional questions require performing multiple compositional operations to arrive at the answer (e.g. difference of the maximum and the minimum). People also regularly ask visual questions (12%) that refer to visual features of the chart. Visual questions tend to be lookups (68%) while non-visual questions tend to be compositional (75%).

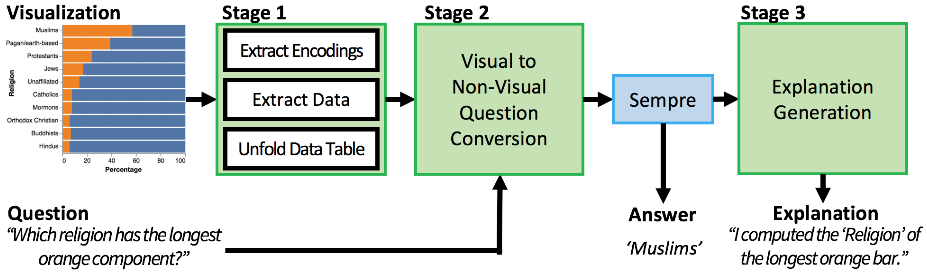


Figure 5.2: Our question answering pipeline for charts operates in three stages. In Stage 1, it extracts visual encodings and the data from the chart and then restructures the data table. In Stage 2, it transforms the input question, replacing any visual references to chart elements with non-visual references to data. Then, it passes the restructured data table and the transformed question to Sempre [127, 187], a state-of-the-art table question answering system which generates the text answer. Finally, in Stage 3, it generates a natuaral language explanation describing how the answer was generated from the chart.

Most importantly, we find that people frequently provide visual explanations (51%), which describe the process of extracting an answer from the visual features of a chart. Consider Q2 in Figure 5.1 where the correct answer is *'Muslims'*. A person might visually explain how they got the answer by reporting *"Muslims have the longest orange bar in the chart."* In contrast, a non-visual explanation such as *"Muslims are about 57% Common, more than any other Religion,"* only refers to the data and does not describe the process of extracting the data from the visual features of the chart. Thus, it is less thorough and this lack of completeness may explain why non-visual explanations are slightly less common than visual explanations.

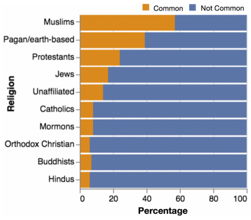
5.2.3 Additional Collection of Visual Questions

To better understand how visual questions are posed we also collected a set of 277 visual questions about charts from our collegues. We analyzed these questions to to identify the lexical and the syntactic structures that people typically use to refer to marks and visual attributes in visual questions. We use the results of this analysis for converting visual questions to non-visual questions in Stage 2 of our pipeline.

5.3 Method

Our question answering system takes a chart and a natural language question as input and outputs the answer to the question along with an explanation (Figure 5.2). Our approach is to adapt Sempre [127, 187], a question answering algorithm that works with relational data tables instead of charts. In Stage 1 of our pipeline, we extract the visual encodings that map data to the attributes of visual marks (e.g. height of a bar mark, color of lines, etc.). We also extract the data itself from the input chart. In Stage 2, we use the extracted encodings to transform the input question, replacing all references to visual marks and their attributes with references to data fields and data values. This transformation converts a *visual question* into a purely *non-visual question*. Next, we input the unfolded table and the transformed, non-visual question into Sempre to generate the answer. Sempre converts the input natural language question into a logical query called a *lambda expression*, and then executes the query on the data table to generate the answer. Finally, in Stage 3, we convert the lambda expression from Sempre into a visual explanation for the answer, using template-based translation.

5.3.1 Stage 1: Extract Data Table and Encodings



```
"data": {"url": "data/kong/data/3.csv"},
"transform": [
    {"filter": "datum.year == 2000"},
    {"filter": "datum.question == 'Extremism'"}],
"mark": "bar",
"encoding": {
    "x": {"field": "Percentage", "type": "quantitative"},
    "y": {"field": "Religion", "type": "nominal"},
    "color": {
        "field": "Response", "type": "nominal",
        "scale": {
            "domain": ["Common", "Not common"],
            "range": ["#EE8426", "#5376A7"]}}}
```

Figure 5.3: Vega-Lite specification for the chart in Figure 5.1. This specification includes a block of data *"transforms"* (orange keyword text) that filter the data to specific years and questions. The *"mark"* (blue keyword) is specified as *'bar'*, and the visual *"encodings"* (pink keyword) for x-position, y-position, and color of the marks are given explicitly.

Religion	Response	Percentage
Muslims	Common	57
Muslims	Not common	43
Pagan/earth-based	Common	39
Pagan/earth-based	Not Common	61
⋮	⋮	⋮
Hindus	Common	6
Hindus	Not Common	94

(a) Flat data table

Religion	Common	Not common
Muslims	57	43
Pagan/earth-based	39	61
Protestants	24	76
Jews	17	83
⋮	⋮	⋮
Buddhists	7	93
Hindus	6	94

(b) Unfolded data table

Figure 5.4: (a) Data extracted from the chart in Figure 5.1 is initially a flat relational data table. Each row represents one mark in the chart. (b) We unfold the table by choosing the *'Response'* column as a pivot, turning each of its data values into column headers and then re-aligning the data in the other columns. In (a), each *'Religion'* and *'Response'* value appears multiple times but only once in (b) reducing the size of the table by almost a factor of two. Moreover, in (b), looking up a specific (*'Religion'*, *'Response'*) pair such as (*'Hindus'*, *'Not common'*) requires looking for the value at the intersection of the pair rather than searching through all the rows corresponding to Hindus as in (a).

A chart is typically constructed by encoding (or mapping) the data to some visual attributes (e.g. position, area, color) of graphical marks (e.g. circles, rectangles) [117]. Vega-Lite [139] is a chart specification language that explicitly describes how input data should be transformed (e.g. aggregating it, re-scaling it) to make it suitable for visualization, and how the transformed data should be encoded using visual attributes of the marks (Figure 5.3).

In Stage 1 of our pipeline, we convert an input chart into a Vega-Lite specification and then extract encodings as well as the transformed data. Finally, we unfold the extracted data into a data table. We first describe how our extraction process works for a Vega-Lite chart and then explain how we convert other types of input charts into the Vega-Lite format.

Extraction from Vega-Lite Charts

Extract encodings. Given a Vega-Lite chart specification as in Figure 5.3, we can directly extract the encodings by looking for `encoding` keyword. In this example, the y-position attribute of a bar mark encodes a nominal data field named *'Religion'*, while the length attribute of the bar encodes a quantitative data field named *'Percentage'*. Similarly, the color attribute encodes a nominal data field named *'Response'* that takes the value 🟧 #EE8426 for the response *'Common'* or the value 🟦 #5376A7 for the response *'Not common'*.

Extract data. To extract the data from the chart, we first run the Vega-Lite interpreter and apply all data transformations in the chart specification. We then capture the transformed data from the Vega-Lite interpreter just before it is rendered into a chart. Specifically, we instrument the chart specification to write out the data immediately after the last transformation. The resulting data is in the form of a *flat* relational table where each row represents a single mark in the chart, or equivalently a single data tuple (Figure 5.4a).

Unfold data table. Table question answering systems like Sempre are trained using human readable tables which are often structured in an *unfolded* format in which a data tuple is comprised of a row header, a column header and the data value at their intersection (Figure 5.4b). Human readers typically prefer such unfolded tables to the corresponding flat relational tables because they are more compact and thereby reduce the cognitive effort required to retrieve information.

Sempre has been trained on a large set of tables from the WikiTableQuestions [127] dataset where manual inspection shows that many of the tables are unfolded. Therefore, we unfold our flat relational data tables into a form that is closer to Sempre's training data. Specifically, we implement Raman and Hellerstein's [132] `unfold` operation as follows. We first check that the extracted data table has a *pivot column* whose data values will be transformed into column headers. The pivot column must contain data values that repeat with the same frequency greater than one. In our example, the *'Religion'* column repeats each religion with a frequency of 2, while the *'Response'* column repeats each response with a frequency of 10 – each response appears once for each of the

10 religions in the dataset. We choose the column with the largest frequency as the pivot and realign the data values in the other columns to form the unfolded table (Figure 5.4b). The resulting unfolded data table is passed as input to Sempre.

Converting Charts into Vega-Lite

While our extraction procedures are designed for charts specified using Vega-Lite, we can handle other forms of charts by converting them into the Vega-Lite format. For visualizations created using D3.js [36], we apply the D3 deconstructor of Harper and Agrawala [63, 64] to automatically convert them to Vega-Lite. The most prevalent representation of charts today is a bitmap. For such chart images, we first use ReVision [140] to extract the data and the marks and then manually add the visual encodings to convert the chart into a complete Vega-Lite specification. We leave it to future work to incorporate alternate methods for extracting data, marks and visual encodings [30, 129] from bitmaps of charts.

5.3.2 Stage 2: Visual to Non-Visual Question Conversion

In Stage 2, we transform an input question which may refer to visual aspects of the chart such as its marks and visual attributes, into a non-visual question that only refers to the data depicted in the chart. For example, consider the chart in Figure 5.1 and visual question Q2, *"Which religion has the longest orange component?"* Our goal is to convert this visual question into the corresponding non-visual question, *"Which religion has the most Percentage of Common Response data?"*

The visual version of the question uses the word *'component'* to refer to marks (bar segments) and the word *'orange'* to refer to a value (orange) of the visual attribute (color) of the marks. The word *'longest'* refers to performing an `argmax` operation on the visual attribute (length) over the marks, and we call such operations on visual attributes *visual operations.* Our approach identifies references to marks, visual attributes and visual operations in the question and then converts them to references to the data and operations on the data to build the the non-visual question in a sequence of 6 steps (Figure 5.5).

Step 1: Mark detection. The first step is to detect all words referring to graphical marks in the chart. Our approach is to check whether each word in the question appears in a list of *mark words* that we manually built in a one-time pre-process from our analysis of the additional set of visual questions we collected in the formative study (Figure 5.6 cyan). For instance, one may refer to a bar in a stacked bar chart as a *'component'*, *'portion'* or *'segment'* and we include these words in the list for bar marks. Thus for the question *"Which religion has the longest orange component?"*, we detect the word *'component'*, as referring to the bar marks in the stacked bar chart (Figure 5.5).

Step 2: Dependency parsing. In the second step, we identify a set of words describing each graphical mark based on the grammatical structure of the question. We start by applying the

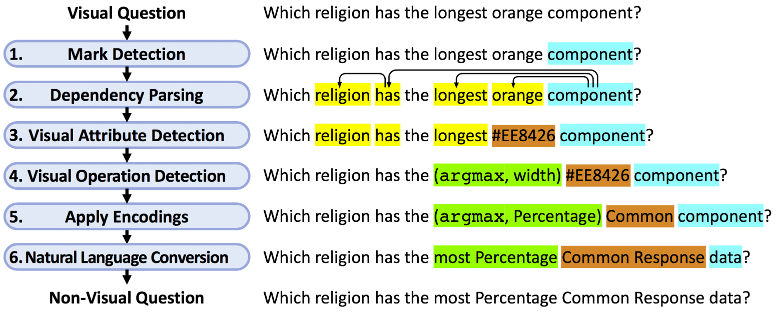

Visual Question
Which religion has the longest orange component?
1. Mark Detection
Which religion has the longest orange component?
2. Dependency Parsing
Which religion has the longest orange component?
3. Visual Attribute Detection
Which religion has the longest #EE8426 component?
4. Visual Operation Detection
Which religion has the (argmax, width) #EE8426 component?
5. Apply Encodings
Which religion has the (argmax, Percentage) Common component?
6. Natural Language Conversion
Which religion has the most Percentage Common Response data?
Non-Visual Question
Which religion has the most Percentage Common Response data?

Figure 5.5: Six steps used to convert the visual question (Q2 in Figure 5.1) into a non-visual question. The system detects *'component'* as a reference to the bar marks, it detects *'orange'* as a visual attribute word, and it detects *'longest'* as a visual operation word. It rewrites these words and outputs the rewritten non-visual question.

Stanford CoreNLP dependency parser [75, 106] to obtain a parse tree that encodes phrase-level dependency structure. For instance, in the dependency tree for our example question (Figure 5.7), the word *'orange'* is an adjectival modifier `amod` for mark word *'component'*.

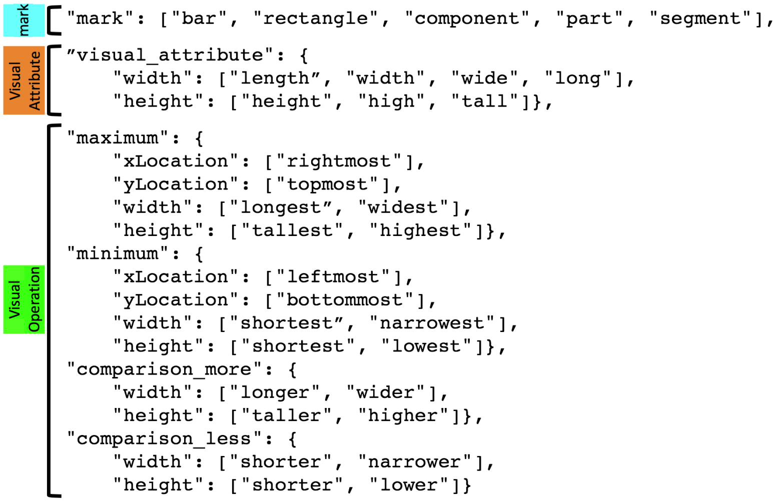
To obtain the words describing each mark, we find the tree node for each mark word in the question and traverse outwards following edges to its parents and children in breadth first order. We retain only the words corresponding to the following set of dependency labels: `acl`, `amod`, `compound`, `conj`, `dep`, `dobj`, `nmod`, and `nsubj`. We obtained this list by analyzing the complete set of dependency labels [166] along with our sample of additional visual questions collected in the formative study and noting that these labels best captured the visually descriptive words for each mark.

For our example (Figure 5.7), we traverse the tree starting at the mark word *'component'* and add the `amod` words *'longest'* and *'orange'* to the list of descriptive words, but we do not add the `det` word *'the'*. We would also traverse up the tree adding the parent word *'has'* with the relation `dobj` and then down through its children adding the `nsubj` word *'religion'*, but not the `det` word *'Which'*. Thus, for the mark word *'component'* we obtain the following set of descriptive words *'religion'*, *'has'*, *'longest'*, and *'orange'* as shown in Figure 5.5.

Questions often use color words to refer to marks without including a mark word. Thus, if the question contains a color word, we always add it to the list of descriptive words.

Step 3: Visual attribute detection. We next identify all the visual attribute words in the list of descriptive words. Our approach is similar to the approach in step 1 for mark detection.

As a one-time pre-process, we built a list of attribute words for each mark type by analyzing our sample of example visual questions. In this analysis, we noticed that the same word can refer to a different visual attribute depending on the mark type (e.g., the word *'height'* refers to the *'length'* of a bar in a vertical bar chart whereas it refers to the *'y-position'* of a point in a line chart), so we created a separate list of visual attribute words for each mark type. The field `visual_attribute` in Figure 5.6 (labeled in orange) shows an example of the alternatives word list for visual attributes of


mark
Visual Attribute
Visual Operation

```
"mark": ["bar", "rectangle", "component", "part", "segment"],
”visual_attribute": {
    "width": ["length”, "width", "wide", "long"],
    "height": ["height", "high", "tall"]},
"maximum": {
    "xLocation": ["rightmost"],
    "yLocation": ["topmost"],
    "width": ["longest”, "widest"],
    "height": ["tallest", "highest"]},
"minimum": {
    "xLocation": ["leftmost"],
    "yLocation": ["bottommost"],
    "width": ["shortest”, "narrowest"],
    "height": ["shortest", "lowest"]},
"comparison_more": {
    "width": ["longer", "wider"],
    "height": ["taller", "higher"]},
"comparison_less": {
    "width": ["shorter", "narrower"],
    "height": ["shorter", "lower"]}
```

Figure 5.6: Word lists used in our pipeline for marks of type *’bar’*. The list of alternative words for referring to *’bar’* marks (cyan). The list of alternative words to refer to visual attributes of *’bar’* marks (orange). The list of visual operations and the alternative natural language word (e.g. rightmost, bottommost, wider, narrower) corresponding to each one (green) In this case the list also includes a visual attribute (e.g. xLocation, height) that the visual operation applies to. The operations are given at the top level and the visual attributes they operate on are given in the second level. The word lists for line charts are included in Chapter A.

bar marks.

We next filter the complete visual attributes list to just the ones that appear in the visual encodings we extracted from our chart in Stage 1. Then to identify which words in our descriptive words list refer to visual attributes, we iterate over each descriptive word and find the closest match in our filtered visual attributes list. Since there are lots of ways to describe visual attributes, we use a *word2vec*-based synonym finding approach to detect a match. Specifically, for each descriptive word and each filtered visual attribute word, we lookup the 300-dimensional word2vec vector generated by the pre-trained model of Mikolov et al. [112, 113] trained on the Google News dataset [178]. We then compute the cosine similarity between their word2vec vectors and accept the best similarity match above a threshold τ (empirically set to 0.75).

Questions sometimes contain descriptive words referring to a color (e.g. *’orange’, ’red’, ’blue’*). Such color words are generally ambiguous as the word *’red’* may refer to a range of different RGB values. Thus, whenever we encounter a descriptive color word, we first lookup the descriptive word in the text color names of the X11 color list [181] to obtain the corresponding RGB hex code. We then consider any encoding involving the *color* attribute, and examine all the RGB values the attribute takes within the chart. Finally, we replace the color word in the question with the RGB hex code that appears in the chart and is closest (in Euclidean RGB distance) to the X11 RGB hex code. In our example, the descriptive word *‘orange’* yields the X11 hex code 🟧 #FFA500 and of the two

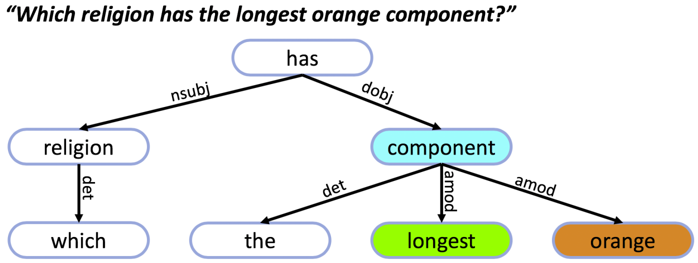


Figure 5.7: The dependency tree generated by the Stanford CoreNLP dependency parser for the question *"Which religion has the longest orange component?"*. Words comprising a noun phrase have the same parent noun in the tree and the tree provides a dependency relationship label for each edge (e.g. `amod` is an adjectival modifier, `det` is a determiner, `nsubj` is a nominal subject). In this case, the word for the visual attribute *'orange'* (orange) and the word for the visual operation *'longest'* (green) are *adjectival modifiers* (`amod`) of the mark word *'component'* (cyan)

colors 🟧 #EE8426 and 🟦 #5376A7 that appear in the chart (Figure 5.1), it is closest to the former. Therefore, we replace the word *'orange'* with 🟧 #EE8426 as shown in Figure 5.5.

Step 4: Visual operation detection. In step 4, we identify all the visual operation words in our remaining list of descriptive words. As in steps 1 and 3, we performed a one-time pre-process to build lists of alternative visual operation words (e.g. longest, narrowest, etc.). Each such visual operation word (e.g. tallest) implies performing an operation (e.g. `argmax`) on a specific visual attribute such as the *height* of a mark. Therefore, our visual operations word list maintains an (operation, attribute) pair for each visual operation word (Figure 5.6 green). To match a descriptive word to the visual operation words, we use the word2vec approach we used in step 3. In our example, we detect *'longest'* as a visual operation word and interpret it as the visual operation (`argmax`, width) as shown in Figure 5.5.

Note that we identify simple questions about the encoding (e.g. *"What is blue depicting?"*) by checking if the input question only refers to one visual feature or attribute of the chart and does not refer to a mark, data or visual operation. In such cases, we directly use the encodings we identified in Stage 1 to answer the question (e.g. *"Not Common"*), bypassing the rest of Stage 2.

Step 5: Apply encodings. In step 5, we use the encodings extracted in Stage 1 to replace the words corresponding to visual attributes and visual operations with words corresponding to data fields and data values. Specifically, we replace visual attribute words to the corresponding data field it encodes as given in the encoding. For specific visual attribute values like the orange color 🟧 #EE8426 we extracted in the step 3, we replace the attribute value with the corresponding data values – in this case the response *'Common'*. For visual operation words, we lookup the corresponding (operation, attribute) pair and replace the visual attribute with the corresponding data field based on the corresponding encoding. For example, given the visual operation word *'longest'*, we lookup the (`argmax`, width) pair, then find the encoding for the *width* attribute in the Vega-Lite specification and

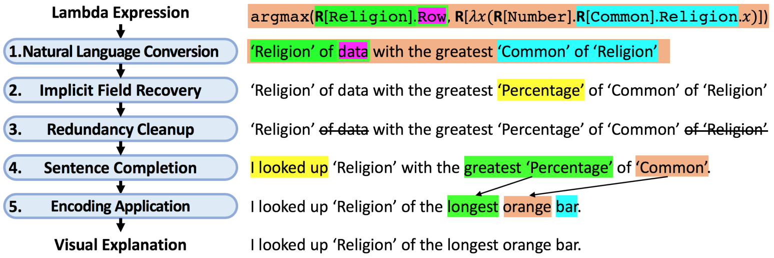

Lambda Expression
argmax(R[Religion].Row, R[λx(R[Number].R[Common].Religion.x)])
1.Natural Language Conversion
'Religion' of data with the greatest 'Common' of 'Religion'
2. Implicit Field Recovery
'Religion' of data with the greatest 'Percentage' of 'Common' of 'Religion'
3. Redundancy Cleanup
'Religion' ~~of data~~ with the greatest 'Percentage' of 'Common' ~~of 'Religion'~~
4. Sentence Completion
I looked up 'Religion' with the greatest 'Percentage' of 'Common'.
5. Encoding Application
I looked up 'Religion' of the longest orange bar.
Visual Explanation
I looked up 'Religion' of the longest orange bar.

Figure 5.8: Steps for generating an explanation from the lambda expression given by Sempre for the input question *"Which religion has the longest orange component?"* (Q2 in Figure 5.1). The system first converts the lambda expression generated by Sempre into a non-visual natural explanation. It then converts the non-visual explanation to a visual explanation by applying the visual encodings.

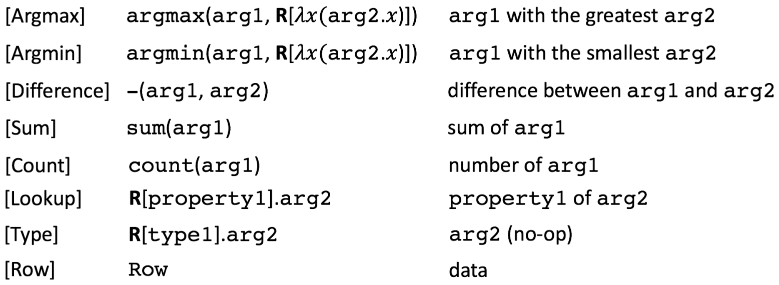
finally replace the attribute width with the corresponding data field *'Percentage'* from the encoding. Thus, we interpret the operation `argmax` as acting on the data field *'Percentage'* (Figure 5.5).

Step 6: Natural language conversion. In the final step, we convert our question into a non-visual natural language question suitable for input into Sempre, by rewriting words representing marks, visual attributes and visual operations using natural language equivalents. Because a mark represents a piece of data, we rewrite all mark words with the generic noun *'data'*. In Step 5, we converted the the visual attribute words into a corresponding data field or data value and we consider these as already in natural language. In our example, *'orange'* has already been converted into the data value *'Common'*. If as in this case the attribute word refers to a data value, we append the corresponding data field name to indicate the context in which the data value should be interpreted – in this case we append the data field name *'Response'* to the data value *'Common'*. Finally, if the attribute word is used as a noun, we add the word *'value'* to force the resulting conversion into a noun. For the visual operation words, we replace the operation word pairs e.g. (`argmax`, *'Percentage'*) with the natural language equivalent of the operation while removing pair notation e.g. *'most Percentage'*. Thus, the input question *"Which religion has the longest orange component?"* is rewritten as *"Which religion has the most Percentage Common Response data?"* While the non-visual question is not completely fluent, together with our unfolded data table it contains enough information for Sempre to answer it correctly: *"Muslims"*.

5.3.3 Stage 3: Explanation Generation

In Stage 3, our pipeline generates a visual explanation describing how the answer was extracted from the chart's visual features. Our approach takes the logical query *lambda expression* Sempre builds to answer the question and uses template-based natural language generation to produce the explanation.

Consider the example question *"Which religion has the longest orange component?"* (Q2 in



[Argmax]	argmax(arg1, **R**[λx(arg2.x)])	arg1 with the greatest arg2
[Argmin]	argmin(arg1, **R**[λx(arg2.x)])	arg1 with the smallest arg2
[Difference]	–(arg1, arg2)	difference between arg1 and arg2
[Sum]	sum(arg1)	sum of arg1
[Count]	count(arg1)	number of arg1
[Lookup]	**R**[property1].arg2	property1 of arg2
[Type]	**R**[type1].arg2	arg2 (no-op)
[Row]	Row	data

Figure 5.9: Rules for converting operations in lambda expressions to natural language for some of the common operations. First column shows the name of the rule, the second column shows the labmda expression and the third column shows the corresponding natural language expression. We include more rules in Chapter A.

Figure 5.1). In Stage 2, we generate the corresponding non-visual question *"Which religion has the most Percentage Common Response data?"*. Sempre then converts this question into the lambda expression

$$\texttt{argmax}(\mathbf{R}[\texttt{Religion}].\texttt{Row}, \mathbf{R}[\lambda x[\mathbf{R}[\texttt{Number}].\mathbf{R}[\texttt{Common}].\texttt{Religion}.x]]),$$

which it executes on the unfolded table we generated in Stage 1 to produce the correct answer *'Muslims'*. Our goal in Stage 3 is to convert this lambda expression to the natural language visual explanation *"I computed the 'Religion' of the longest orange bar."* We use a 5 step pipeline to generate the explanation (Figure 5.8).

For questions about the encoding that we detected in step 4 of Stage 2 (e.g. *"What is blue depicting?"*), we directly generate the explanation using the template *"I looked up what [encoding] represents by looking at the [label on the x-axis / label on the y-axis / legend],"* based on whether the encoding is specified by the x-axis, the y-axis or the legend. We do not process such questions through the steps in this stage.

Step 1: Natural language conversion. As presented by Liang [99], lambda expressions include a limited set of operations and generation rules. Thus, we build a small set of rules to convert lambda expression to natural language (a subset of our rules is shown in Figure 5.9). For our example, our pipeline applies the `argmax`, `type`, `lookup`, and `row` rules to convert the input lambda expression to *"'Religion' of data with the greatest 'Common' of 'Religion'."*

Step 2: Implicit field recovery. Sometimes, a field name becomes implicit during the table unfolding in Stage 1, and we maintain the field name as an auxiliary annotation to the table. For instance, during the unfolding process in Figure 5.4, we keep the field name *'Percentage'* as auxiliary annotation on each of the cells in the *'Common'* and *'Not common'* columns. In this step we add this implicit annotation to the reference to the value *'Common'* of the pivoted field in the explanation, resulting in *"'Religion' of data with the greatest 'Percentage' of 'Common' of 'Religion'."*

Step 3: Redundancy Cleanup. Our pipline next removes any redundant information using a series of regex rules. In our explanation, we see that the information about *'Religion'* is repeated twice at the beginning and end of the expression. Moreover, *"'Religion' of data"* does not carry more information than just *'Religion'*. Both of these issues make the explanation difficult to understand. The cleanup step removes these extraneous words and yields *"'Religion' with the greatest 'Percentage' of 'Common'."* We include the specific regex rules in Chapter A.

Step 4: Sentence Completion. Next, we generate a non-visual explanation by adding the pronoun *'I'* and a verb that describes the last operation performed by the system. For the verb, we use *'looked up'* for lookup operations, *'counted'* for counting operations, and *'computed'* for all other operations. In our example, we add *"I looked up"* to the beginning of the explanation to complete a non-visual explanation.

Step 5: Encoding application. To make the non-visual explanations visual, we apply the visual encodings obtained from Stage 1. For references to values of fields that are encoded as colors, we convert them to color words directly. For references to data fields encoded as other visual features, we check the surrounding words to see if there is an operation performed on the visual attribute, and convert it to a visual attribute word or a visual operation word using the word lists we used in Stage 2 (Figure 5.6). We add a mark word and position the converted visual words so that they modify the mark word. In our example, we convert the value *'Common'* to the color 'orange'. For the reference to the field *'Percentage'*, we use the neighboring word *'greatest'* and the visual encodings to recognize that this is an operation `argmax` on the visual attribute *width*, and use the visual operations word list to convert this to the visual operation word *'longest'*. Rearranging these words so that they modify the mark word *'bar'* yields *" I looked up 'Religion' of the longest orange bar."* Details about choice of color words and word rearrangements are in Chapter A.

5.4 Results

As shown in Figure 5.10, we find that across all 629 questions in our corpus, our pipeline answers 51% correctly. As a baseline, we compare this result to using Sempre with the flat relational tables initially extracted in Stage 1 in place of the charts and find that it only answers 39% of the questions correctly. Our pipeline greatly outperforms Sempre on visual questions with improvements of 53% for all visual questions, 74% on visual lookup questions and 8% on visual compositional questions. We find that for even for non-visual questions, our system outperforms Sempre, by 6% overall, 19% on non-visual lookup questions and 2% on non-visual compositional questions.

Figure 5.11 compares the accuracy of our complete pipeline to a pipeline in which we only retain Stage 1 (and eliminate Stage 2—visual to non-visual conversion) and to a pipeline in which we only retain Stage 2 (more specifically, we include data and encoding extraction from Stage 1 but eliminate data table unfolding). Although both stages contribute significantly to the overall success

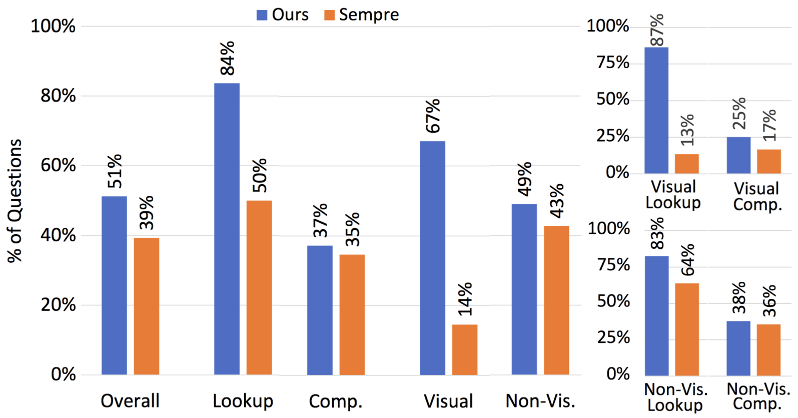


Figure 5.10: Accuracy of our pipeline (blue) compared to a baseline version of Sempre (orange) for questions of each type (visual/non-visual and lookup/compositional).

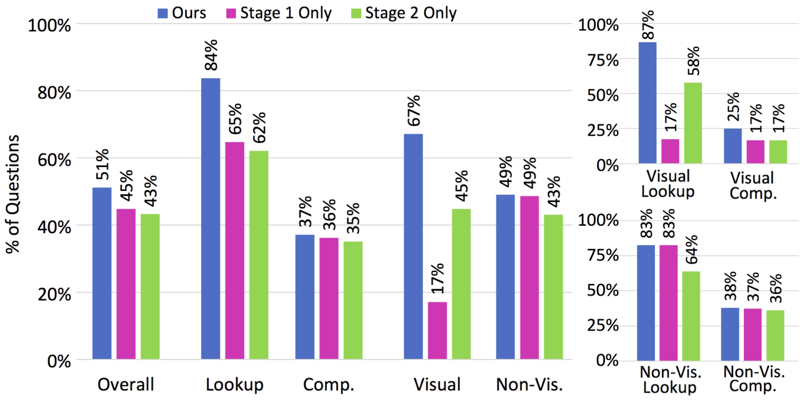
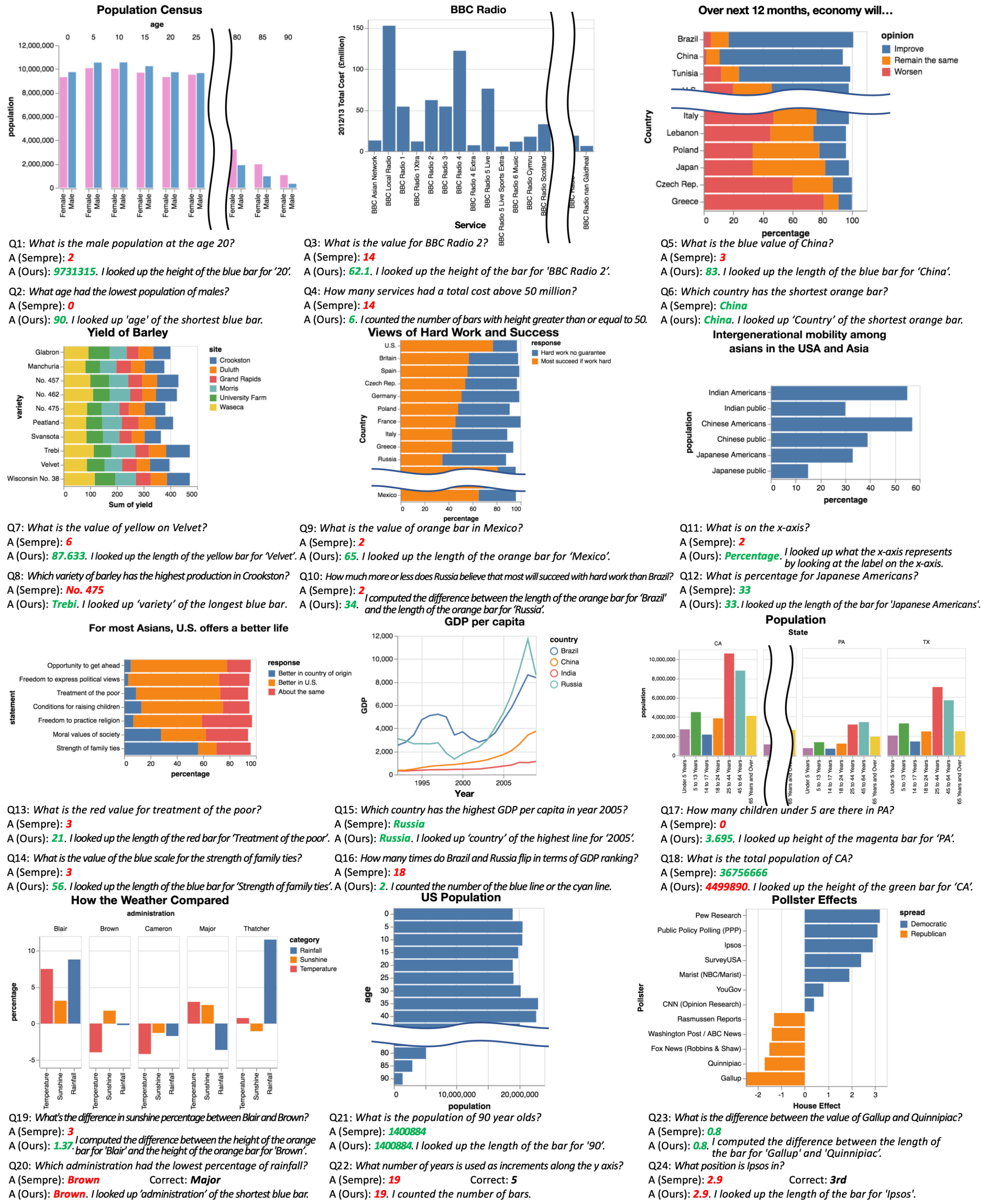


Figure 5.11: Accuracy of our complete pipeline (blue) compared to the pipeline with the data table unfolding of Stage 1 only (purple) and the question transformation of Stage 2 only (green).

of our pipeline, we see a major improvement in answering visual questions from Stage 2, which is not surprising as Stage 2 is responsible for converting visual questions into the non-visual form necessary for Sempre.

Figure 5.12 shows a variety of charts and questions with answers and explanations generated by our pipeline, as well as the answers generated by the baseline version of Sempre. We see that our system generates correct answers and explanations for many questions that Sempre cannot answer correctly. In particular, Stage 2 of our pipeline handles visual features and allows our pipeline to correctly answers visual questions (Q5, Q7, Q9, Q11, Q13, Q14). It even correctly answers non-visual questions both lookup (Q1, Q3, Q17) and compositional (Q2, Q4, Q8, Q10, Q19). However, our pipeline sometimes outputs a wrong answer for a question Sempre gets correct, as in Q18. In this case the error is due to a change in table structure from table unfolding in Stage 1 of our pipeline.



Q1: *What is the male population at the age 20?*
A (Sempre): ***2***
A (Ours): ***9731315***. *I looked up the height of the blue bar for '20'.*

Q2: *What age had the lowest population of males?*
A (Sempre): ***0***
A (Ours): ***90***. *I looked up 'age' of the shortest blue bar.*

Q3: *What is the value for BBC Radio 2?*
A (Sempre): ***14***
A (Ours): ***62.1***. *I looked up the height of the bar for 'BBC Radio 2'.*

Q4: *How many services had a total cost above 50 million?*
A (Sempre): ***14***
A (Ours): ***6***. *I counted the number of bars with height greater than or equal to 50.*

Q5: *What is the blue value of China?*
A (Sempre): ***3***
A (Ours): ***83***. *I looked up the length of the blue bar for 'China'.*

Q6: *Which country has the shortest orange bar?*
A (Sempre): ***China***
A (ours): ***China***. *I looked up 'Country' of the shortest orange bar.*

Q7: *What is the value of yellow on Velvet?*
A (Sempre): ***6***
A (Ours): ***87.633***. *I looked up the length of the yellow bar for 'Velvet'.*

Q8: *Which variety of barley has the highest production in Crookston?*
A (Sempre): ***No. 475***
A (Ours): ***Trebi***. *I looked up 'variety' of the longest blue bar.*

Q9: *What is the value of orange bar in Mexico?*
A (Sempre): ***2***
A (Ours): ***65***. *I looked up the length of the orange bar for 'Mexico'.*

Q10: *How much more or less does Russia believe that most will succeed with hard work than Brazil?*
A (Sempre): ***2***
A (Ours): ***34***. *I computed the difference between the length of the orange bar for 'Brazil' and the length of the orange bar for 'Russia'.*

Q11: *What is on the x-axis?*
A (Sempre): ***2***
A (Ours): ***Percentage***. *I looked up what the x-axis represents by looking at the label on the x-axis.*

Q12: *What is percentage for Japanese Americans?*
A (Sempre): ***33***
A (Ours): ***33***. *I looked up the length of the bar for 'Japanese Americans'.*

Q13: *What is the red value for treatment of the poor?*
A (Sempre): ***3***
A (Ours): ***21***. *I looked up the length of the red bar for 'Treatment of the poor'.*

Q14: *What is the value of the blue scale for the strength of family ties?*
A (Sempre): ***3***
A (Ours): ***56***. *I looked up the length of the blue bar for 'Strength of family ties'.*

Q15: *Which country has the highest GDP per capita in year 2005?*
A (Sempre): ***Russia***
A (Ours): ***Russia***. *I looked up 'country' of the highest line for '2005'.*

Q16: *How many times do Brazil and Russia flip in terms of GDP ranking?*
A (Sempre): ***18***
A (Ours): ***2***. *I counted the number of the blue line or the cyan line.*

Q17: *How many children under 5 are there in PA?*
A (Sempre): ***0***
A (Ours): ***3.695***. *I looked up height of the magenta bar for 'PA'.*

Q18: *What is the total population of CA?*
A (Sempre): ***36756666***
A (Ours): ***4499890***. *I looked up the height of the green bar for 'CA'.*

Q19: *What's the difference in sunshine percentage between Blair and Brown?*
A (Sempre): ***3***
A (Ours): ***1.37***. *I computed the difference between the height of the orange bar for 'Blair' and the height of the orange bar for 'Brown'.*

Q20: *Which administration had the lowest percentage of rainfall?*
A (Sempre): ***Brown*** Correct: ***Major***
A (Ours): ***Brown***. *I looked up 'administration' of the shortest blue bar.*

Q21: *What is the population of 90 year olds?*
A (Sempre): ***1400884***
A (Ours): ***1400884***. *I looked up the length of the bar for '90'.*

Q22: *What number of years is used as increments along the y axis?*
A (Sempre): ***19*** Correct: ***5***
A (Ours): ***19***. *I counted the number of bars.*

Q23: *What is the difference between the value of Gallup and Quinnipiac?*
A (Sempre): ***0.8***
A (Ours): ***0.8***. *I computed the difference between the length of the bar for 'Gallup' and 'Quinnipiac'.*

Q24: *What position is Ipsos in?*
A (Sempre): ***2.9*** Correct: ***3rd***
A (Ours): ***2.9***. *I looked up the length of the bar for 'Ipsos'.*

Figure 5.12: Sample questions from our corpus with answers generated by our pipeline as well as a baseline version of Sempre. Answers in green are correct and answers in red are incorrect. If neither of pipeline generated a correct answer, we also report the correct answer as in Q20, Q22, and Q24. We encourage readers to zoom in to the figure to read the text.

Explanation	Transparency	Trust	Usefulness	Accuracy(%)	Time(s)
None	1.3 (±0.8)	3.0 (±1.1)	-	87.5	26.2 (±28.3)
Human	3.3 (±0.8)	**3.3** (±0.9)	3.4 (±1.5)	91.3	26.0 (±27.5)
Non-Visual	3.4 (±1.3)	3.1 (±0.9)	3.6 (±1.4)	95.0	**23.7** (±21.2)
Visual	**3.9** (±1.1)	**3.3** (±0.9)	**3.7** (±1.5)	**98.8**	26.7 (±30.5)

Table 5.2: Results from the user study (each result is represented as $avg(\pm stdev)$). We see that for most of the measures, the visual explanations generated by our system achieves the best.

Nevertheless, analyzing the wrong answers produced by our system, we find that 92% are due to Sempre, and 5% are due to incorrect conversions of visual questions. The remaining 4% are because of changes in the table structure from table unfolding. Further analyzing the Sempre errors, 12% are caused by Sempre not including the operation involved in the question. For example, Sempre does not include operations with binary output, making it unable to answer Y/N questions, which accounts for 1.5% of all the questions. We refer to the analysis in the original papers [127, 187] for more details about errors by Sempre.

Our system gets the correct answer for Q16, but from the explanation, we see that it counted the number of lines corresponding to the countries that appeared in the question (i.e. Brazil and Russia) instead of counting the number of flips in the GDP ranking of the two countries; it accidentally got the answer correct. On the other hand, since Sempre does not give explanations, it is unclear how many of the correct answers it gives are obtained through an incorrect process because the model is opaque. Even for questions our system gets wrong (Q18, Q20, Q22, Q24), we see that our system transparently explains how it arrived at the wrong answer.

5.5 User Study

To see how the visual explanations generated by our pipeline do on the measures of transparency, trust, and usefulness, we conducted a user study with four different conditions: (1) the *no-explanation* condition in which we only show the answer to a question, (2) the *human explanation* condition in which we show the answers and explanations generated by humans from our formative study, (3) the *non-visual explanation* condition in which we show the answers and the non-visual explanations generated by our pipeline (at the end of step 4 of Stage 3), and (4) the *visual-explanation* condition in which we show the answers and visual explanations generated by our pipeline. We consider three hypotheses:

H1: Users will find the visual explanation condition more transparent and trustworthy than the no-explanation condition.

H2: Users will find the visual explanation condition better than or at least as good as human explanation condition based on transparency, trust, and usefulness.

H3: Users will find the visual explanation condition better than the non-visual explanation condition based on transparency, trust, and usefulness.

5.5.1 Study Design

We designed a within-subjects study with sixteen participants, all fluent in English. To set up the study we gathered 20 unique charts-question pairs from our corpus, and divided them into four groups of five. We counterbalanced the mapping between the four conditions and the four chart-question groups. We then ran the study in two stages. In the first stage, we randomly shuffled the questions. Along with the chart and the question, we showed the participants answers and explanations (if any) of the condition the question was mapped to. For each question, the participants first determined whether the presented answer was correct, and then rated the usefulness of the explanation on a 5-point Likert scale. We timed how long it took them to determine correctness. In the second stage, we showed each group of questions in a counterbalanced order and asked the participants to rate the transparency and the trustworthiness of the condition on a 5-point Likert scale. Afterwards, we collected free form responses about what the participants considered relevant to the transparency, trustworthiness, and usefulness. The study took about 30 minutes and each participant received a $15.00 Amazon gift card.

5.5.2 Results and Discussion

Assessing H1. Figure 5.2 shows the results of the study. We find that the visual explanations generated by our pipeline significantly increased the transparency of the pipeline compared to the no-explanation condition (Mann-Whitney U = 245.5, $p < 0.001$). Trust towards the visual explanation condition was higher than the no-explanation condition, but the difference was not significant (U = 149.0, $p = 0.21$).

Assessing H2. Participants also found the visual explanations generated by our pipeline significantly more transparent than the human-generated explanations (U = 178.0, $p < 0.05$). We hypothesize that this result is due to the systematic way our pipeline generates the explanations, as one participant put it, *"I like that in some system the explanation is more consistent than others. It guarantees me that it will provide certain information."* Finally, we saw that the trust towards visual explanations generated by our pipeline is very close to that towards human-generated explanations (U = 130.5, $p = 0.46$).

Assessing H3. When we compare between the visual explanations to non-visual explanations generated by our pipeline, we find that the measures of transparency, trust, and usefulness are all higher for the visual explanations, but none of the improvements are significant (U = 153.5, $p = 0.16$; U = 143.5, $p = 0.27$; U = 3356.5, $p = 0.29$, respectively).

In the free form response, most participants (12 of 16) reported explanations as relevant to transparency (e.g., *"I appreciated being able to see from the explanations what caused the system to make errors. Providing no explanation at all made the system seem like a complete black box."*). For trust, participants reported the accuracy of the answer (10 of 16) and whether the explanation matches the answer (4 of 16) as relevant. Finally, for usefulness, participants reported that explanations are more useful if they refer to visual features (7 of 16).

In sum, we find the visual explanations generated by our system are significantly more transparent than human-generated explanations and are comparable in usefulness and trust.

Accuracy and Time. We also measured the accuracy and speed with which participants could confirm the correctness of answers with and without explanations. While these initial measurements show improvements in accuracy for and speed when people have access to explanations, the relatively small differences combined with large variance in timing, suggest that further study is needed to understand the causes of these improvements. We provide more details about these measurements in Chapter A.

Chapter 6

Limitations and Future Work[1]

Although this thesis makes an important step towards extending our knowledge about the relationship between text and visualizations, there are several limitations that can be addressed in future work.

6.1 Generalization of Findings and Algorithms

In this thesis, we have focused on certain types of text and visualizations that best represent the variables of interest while controlling the variations in less relevant variables. While we believe that our findings are generalizable, future work can confirm this and extend our algorithms to other types of text and visualizations.

In Chapter 3, we explored how readers take away information when presented with univariate line charts and captions. Visualizations have prominent features (e.g., extrema in bar charts, outliers in scatterplots) and less prominent features (e.g., a point in a cluster in scatterplots) and we expect similar findings would hold in general. We leave it to future work to confirm this intuition.

In addition, we used a template-based approach for generating captions to minimize the effect of the variation of natural language and to keep the experiment size reasonable. Simultaneously, we carefully varied the visual feature described in the caption and the presence of external information to best understand how people read captions and charts together to form their takeaways. Future work could study captions with various natural language expressions and different ways of emphasis. It would be useful to understand whether the relationship between multiple features in a caption (e.g., a simple list - *“There were major dips in employment in 2008 and 2020.”* or a comparison - *“The dip in 2020 was greater than the dip in 2008.”*) has an effect on what readers take away. Studying how our findings generalize to other types of external information (e.g., extrapolation,

[1] The contents of this chapter has been adapted from the thesis authors works presented in the previous chapters [82, 83, 84].

breakdown into subcategories) would be an interesting direction to pursue.

In Chapter 4, we confirmed that our reference extraction pipeline handles tables and any chart with tabular data [88] and generated a document reader for tables. Similar interfaces showing references between some basic chart types and text [170] and proof statements and text [34] suggest the benefits of reading interfaces that link other types of visualizations and text. The task of reference extraction that could support these interfaces still remain mostly an open problem for many types of visualizations other than some basic chart types [92, 94].

Furthermore, while our work provides evidence that our pipeline generalizes across different document styles, a larger corpus containing additional types of documents (e.g. textbooks, news articles, etc.) would allow us to verify the generalizability of our pipeline across document types. Generating labeled data from the larger corpus would also pave the way for developing modern machine-learning based methods for this problem.

In Chapter 5, we mainly focused on generating visual explanations about questions about charts with tabular underlying data to leverage Sempre [127, 187]. We believe that the principles of transparency and trust apply to question answering on all types of visualizations (e.g., diagrams [80], geometric figures [141, 142]) and future work can seek ways to best explain how a system obtained an answer based on a visualization.

6.2 Implementation and Improvements of Algorithms

In this thesis, we have introduced design guidelines as well as algorithms for helping people see the connections between visualizations and text. Future work can implement tools to help follow the guidelines and improve the algorithms we present in this thesis.

First, we would like to explore how the work in Chapter 3 can provide interesting implications for *both* chart and caption design to help the author effectively convey a specific point of view. Enhancements to visualization authoring tools could suggest chart design alternatives given a feature that the author would like to emphasize. Specifically, the system could go further by emphasizing features in the chart according to the main message the author wants to convey by automatically adding annotations to the chart, adding highlights, and adjusting levels of detail so that the chart and the caption deliver a concerted message. This will require formulating a high-level language specification that the authors can use to communicate to the system about their intents or a natural language processing module that can infer the authors' intents based on the captions they write. Coordinating interaction between the chart and the caption such that hovering over the text in the caption would highlight the corresponding visual feature in the chart and vice-versa, is another interesting direction to pursue to help the reader. The resulting system would be a significant extension of the interactive document reader presented by Kong et al. [88] and Chapter 4. On the captioning side, a system could classify basic captions, captions about high-prominence features, and captions about low-prominence features. Based on the classification, the system could suggest external information to further emphasize the information presented.

In Chapter 4, we verified that readers are better off with our interactive document reader although the accuracy was only 48.8%. This accuracy shows that there is much room for improvement. While our work focuses on references within a single sentence, anaphora resolution techniques [115] and/or a document-level discourse parser [74] could be used to identify references that span multiple sentences. Some references also require external knowledge (e.g. table headers list the countries in the world and the text refers to 'Asian countries'). Knowledge bases such as DBPedia [9], Freebase [18] or Wolfram Alpha [177] could be used to resolve such references. References sometimes include clauses such as 'all but ...' (exclusion) or 'the second most common' (ranking) to indicate specific table cells. Applying compositional semantic parsing [127] may be one approach for resolving references that involve such logical operations.

Our question answering system in Chapter 5 achieves an accuracy of 51% and can be improved further. For example, people refer to visual features of a chart using a variety of words. Our rule-based approach sometimes fails to detect synonyms for these features. A supervised method that learns to detect such visual references and convert visual questions to non-visual questions could provide a more generalizable model. We have also found classes of questions that our system cannot handle, some because Sempre [127, 187] cannot handle them (e.g., yes/no questions), and some compositional questions about visual encodings (e.g., Q22 of Figure 5.12). We hope to extend our pipeline to handle such questions.

Moreover, while our template-based approach generates explanations that can convey how our pipeline obtained an answer, the resulting explanation may lack fluency and offer little variations in style. Applying data-driven neural models for natural language generation [108] may help address such limitations. Moreover, as Kong et al. [88] and Chapter 4 have suggested, highlighting parts of the charts relevant to the explanations might also improve their effectiveness.

Chapter 7

Conclusion[1]

Visualizations and text oftentimes accompany each other. In this thesis, we noted that understanding the relationship between the two representations is challenging. Based on the observation, we introduced tools and design principles that can guide people towards the intended messages by making the relationship clear.

In Chapter 3, we examine what readers take away from both a chart and its caption. Our results suggest that when the caption mentions visual features of differing prominence levels, the takeaways differ. When the caption mentions a specific feature, the takeaways also tend to mention that feature. We also observed that when a caption mentions a visually prominent feature, the takeaways more consistently mention that feature. On the other hand, when the caption mentions a less prominent feature, the readers' takeaways are more likely to mention the most prominent prominence features than the feature described in the caption. We also find that including external information in the caption makes the readers more likely to form their takeaways based on the feature described in the caption. From the results of our study, we propose guidelines to better design charts and captions together; using visual cues and alternative chart representations, visual features can be made more prominent and be further emphasized by their descriptions in the caption. Design implications from this work provide opportunities for the authoring of chart and caption pairs in visual analysis tools to effectively convey a specific point of view to the reader.

Then, in Chapter 4, we have presented a fully automatic pipeline for extracting references between the text and tables in a document. Our pipeline includes three main stages that analyze the structure of the table, apply natural language processing techniques to match sentence text to table cells and refine the matches using the table structure. While our results are not perfectly accurate, the majority of errors are due to false negatives (missing cells), which we have found to be less harmful than false positives (misleading cells) in the user study. We believe that this pipeline is an initial

[1] The contents of this chapter has been adapted from the thesis authors works presented in the previous chapters [82, 83, 84].

step towards more interactive documents that assist readers in absorbing their content by linking and presenting multiple sources of relevant information.

Finally, in Chapter 5, we have presented an automatic pipeline for answering questions about charts and generating visual explanations. In a formative study, we find that people regularly ask visual questions and that visual explanations are both common and effective. Our automatic question-answering pipeline achieves an overall accuracy of 51% on a corpus of real-world chart with human-generated questions. Finally, user study confirms that our system is significantly more transparent than the answers and explanations generated by humans, and that it is on par with the human-generated answers and explanations for trust and usefulness.

Appendix A

Visual Explanations for Chart Question Answering: Additional Information[1]

A.1 Formative Study

A.1.1 Words for Referring to Visual Features

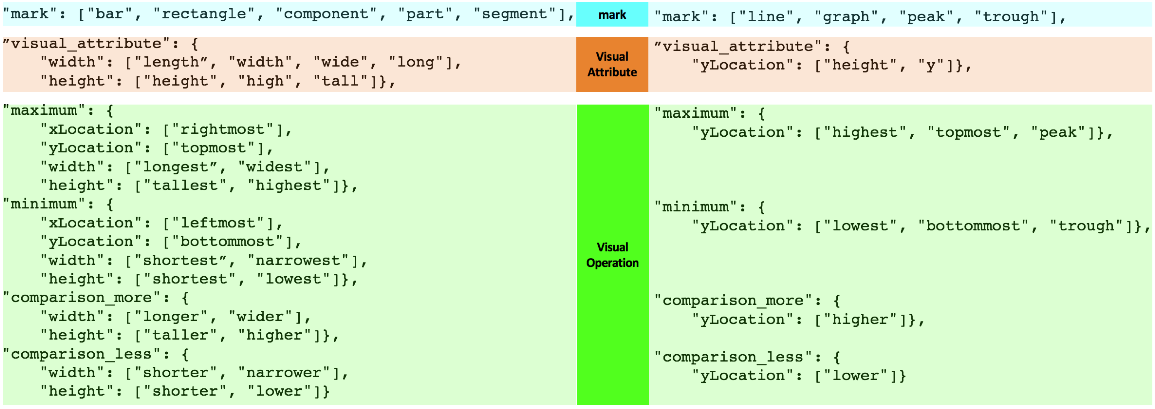
From the 277 visual questions we collected prior to the formative study, we identified the words that people use to refer to the visual features of the charts. We complied these into word lists and use them to detect mark words, visual attribute words and visual operation words in Stage 2 of our pipeline. Here, we include the word lists that we compiled (Figure A.1).

A.1.2 Additional Analysis

In addition to the analysis of how often people ask visual/non-visual or lookup/compositional questions, and provide visual/non-visual explanations, we further analyzed the questions from the formative study to determine which visual elements of the charts people referred to when asking questions or explaining their answers visually. Furthermore, we analyzed if people provide visual explanations when answering visual questions.

Visual Questions 43% of the visual questions included mark words (e.g. ‘bar’, ‘line’). More visual questions referred to the color attributes of the marks (54%) than the length attributes of the marks (22%). 22% of the questions referred to the elements on the axes (e.g. the axis itself, label, ticks).

[1] The contents of this chapter has been adapted from the Supplemental Materials of Kim et al. [82]. This section provides additional information to Chapter 5.



mark
Visual Attribute
Visual Operation

```
"mark": ["bar", "rectangle", "component", "part", "segment"],
”visual_attribute": {
    "width": ["length”, "width", "wide", "long"],
    "height": ["height", "high", "tall"]},
"maximum": {
    "xLocation": ["rightmost"],
    "yLocation": ["topmost"],
    "width": ["longest”, "widest"],
    "height": ["tallest", "highest"]},
"minimum": {
    "xLocation": ["leftmost"],
    "yLocation": ["bottommost"],
    "width": ["shortest”, "narrowest"],
    "height": ["shortest", "lowest"]},
"comparison_more": {
    "width": ["longer", "wider"],
    "height": ["taller", "higher"]},
"comparison_less": {
    "width": ["shorter", "narrower"],
    "height": ["shorter", "lower"]}
```

(a) Word lists for bar charts

```
"mark": ["line", "graph", "peak", "trough"],
”visual_attribute": {
    "yLocation": ["height", "y"]},
"maximum": {
    "yLocation": ["highest", "topmost", "peak"]},
"minimum": {
    "yLocation": ["lowest", "bottommost", "trough"]},
"comparison_more": {
    "yLocation": ["higher"]},
"comparison_less": {
    "yLocation": ["lower"]}
```

(b) Word lists for line graphs

Figure A.1: Word lists used in Stage 2 of our pipeline for marks of type ‘*bar*’ and ‘*line*’ (Extension of Figure 6 in the main paper). The list of words referring to these marks (cyan), the list of words for referring to the visual attributes of the marks (orange) and the list of words for representing operations on the marks (green).

Visual Explanations 87% of the visual explanations included mark words. Unlike for visual questions, more visual explanations referred to the dimension of the marks (42%) than to the color attributes of the marks (30%). 13% of the questions referred to the elements on the axes.

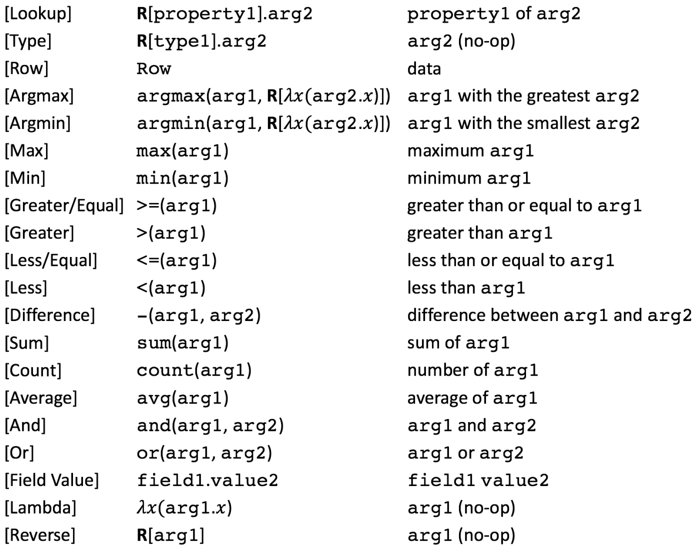
Explanations to Visual/Non-Visual Questions 60% of the explanations to the visual questions were visual, whereas 50% of the explanations to the non-visual questions were non-visual. Visual explanations were slightly more common when the questions were visual, and for both visual and non-visual questions, people provided visual explanations at least half of the time.

A.2 Additional Details for Explanation Generation

We generate the visual explanations from the lambda expressions via using a series of regex rules in Stage 3. Here, we provide more information about the rules used for explanation generation. For specific implementation details, please refer to the released code.

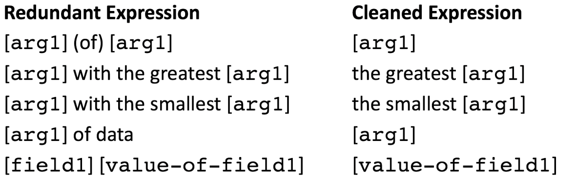
A.2.1 Natural Language Conversion Rules

In Stage 3 step 1, our pipeline converts lambda expressions to natural language using a small set of rules. Figure A.2 shows a set of rules for this process. Please refer to the released code for specifics and precedence.



[Lookup]	**R**[property1].arg2	property1 of arg2
[Type]	**R**[type1].arg2	arg2 (no-op)
[Row]	Row	data
[Argmax]	argmax(arg1, **R**[λx(arg2.x)])	arg1 with the greatest arg2
[Argmin]	argmin(arg1, **R**[λx(arg2.x)])	arg1 with the smallest arg2
[Max]	max(arg1)	maximum arg1
[Min]	min(arg1)	minimum arg1
[Greater/Equal]	>=(arg1)	greater than or equal to arg1
[Greater]	>(arg1)	greater than arg1
[Less/Equal]	<=(arg1)	less than or equal to arg1
[Less]	<(arg1)	less than arg1
[Difference]	-(arg1, arg2)	difference between arg1 and arg2
[Sum]	sum(arg1)	sum of arg1
[Count]	count(arg1)	number of arg1
[Average]	avg(arg1)	average of arg1
[And]	and(arg1, arg2)	arg1 and arg2
[Or]	or(arg1, arg2)	arg1 or arg2
[Field Value]	field1.value2	field1 value2
[Lambda]	λx(arg1.x)	arg1 (no-op)
[Reverse]	**R**[arg1]	arg1 (no-op)

Figure A.2: Conversion rules from lambda expressions to natural language (Extended version of Figure 9 in the main paper). The first column shows the name of the rule, the second column shows the lambda expression and the third column shows the corresponding natural language expression.



Redundant Expression	**Cleaned Expression**
[arg1] (of) [arg1]	[arg1]
[arg1] with the greatest [arg1]	the greatest [arg1]
[arg1] with the smallest [arg1]	the smallest [arg1]
[arg1] of data	[arg1]
[field1] [value-of-field1]	[value-of-field1]

Figure A.3: Redundancy cleanup rules for explanations. The first column shows the original redundant expression, and the second column shows the cleaned-up result.

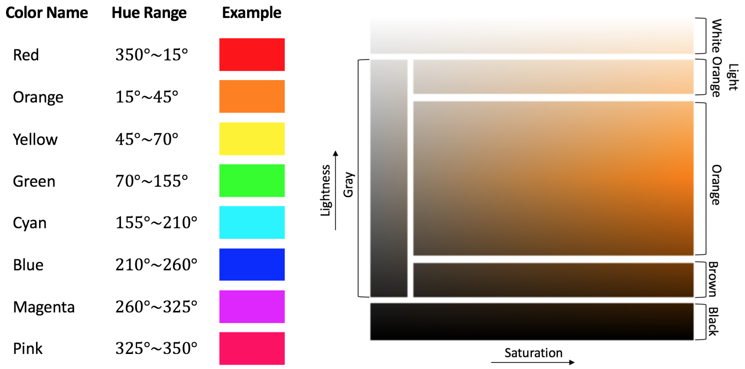
A.2.2 Redundancy Cleanup Rules

We remove two types of redundancies during redundancy cleanup (Stage 3 step 3): (1) repeated mentions of field names or values (e.g. *"'age' of the greatest 'age"'*) or (2) unnecessary mention of field or the word *'data'* (e.g. *"'Country' 'China"'*). Figure A.3 lists some of the regex used in this process. For the specific regex we use, please refer to the released code.

A.2.3 Encoding Application

In Stage 3 step 5, our pipeline applies the encodings to convert references to field names and field values in explanations into the visual attributes of the marks to generate visual explanations.

Choosing Color Words Whereas people may use various color names to describe a color they see, explanations need to be clear and a small set of common color names is all that is needed to



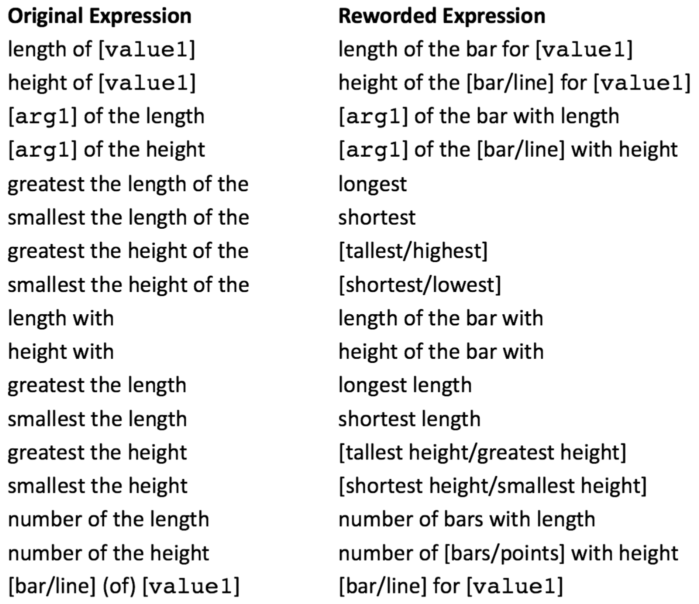
Color Name	Hue Range	Example
Red	350°~15°	🟥
Orange	15°~45°	🟧
Yellow	45°~70°	🟨
Green	70°~155°	🟩
Cyan	155°~210°	🟦
Blue	210°~260°	🟦
Magenta	260°~325°	🟪
Pink	325°~350°	🟥

(a) Hue ranges (b) Saturation and lightness

Figure A.4: How the colors are named in the HSL space. (a) We split the hue space into eight color ranges. The example colors represent colors in the center of the range with 100% saturation and 50% lightness. (b) For each hue value, we split the saturation and lightness space into black, white, light and dark versions of the color, and the color itself. Here, we exemplify this with the color orange (with hue value 30°). We use *'brown'* instead of *'dark orange'* because it is a more commonly used color name.

distinguish the marks. Because the common color names are better spread out throughout the hue-space than in the RGB space, we use the HSL color space to assign names to colors. In comparison, we use RGB color space for matching color words to colors in the chart because RGB color space has a naturally defined metric that allows distance comparisons to different colors used in the chart, whereas the HSL color space does not. We split the hue space into smaller slices according to the color names given by WorkWithColor.com [179]. We split the color ranges of the half-colors (e.g. red-orange, yellow-green) into two halves and merged them with the closest hue range, resulting in a total of eight colors (Figure A.4a). For lightness, we named colors with lightness greater than 87.5% as *'white'* and colors with lightness less than 12.5% as *'black'*. We further split the lightness space and add the adjective *'light'* when the lightness is between 75% and 87.5%, and *'dark'* if it is between 12.5% and 25%. For saturation, we name colors *'gray'* if it has saturation less than 12.5%. If the light or dark shade of the color is often interpreted as a different color, we specially defined the color name (e.g. *'brown'* for *'dark orange'*). Figure A.4b shows the split for orange.

Word Rearrangement Simply applying the encodings may result in natural language expressions that could be made smoother by rearranging the words. For example, *"length of 'China"'* can be smoothed by adding the mark word into *"length of the bar for 'China"'*. In order to do so, we apply a series of regex rules to the resulting explanations (Figure A.5). Please refer to our code for the implementation details as well as the exact precedence.



Original Expression	Reworded Expression
length of [value1]	length of the bar for [value1]
height of [value1]	height of the [bar/line] for [value1]
[arg1] of the length	[arg1] of the bar with length
[arg1] of the height	[arg1] of the [bar/line] with height
greatest the length of the	longest
smallest the length of the	shortest
greatest the height of the	[tallest/highest]
smallest the height of the	[shortest/lowest]
length with	length of the bar with
height with	height of the bar with
greatest the length	longest length
smallest the length	shortest length
greatest the height	[tallest height/greatest height]
smallest the height	[shortest height/smallest height]
number of the length	number of bars with length
number of the height	number of [bars/points] with height
[bar/line] (of) [value1]	[bar/line] for [value1]

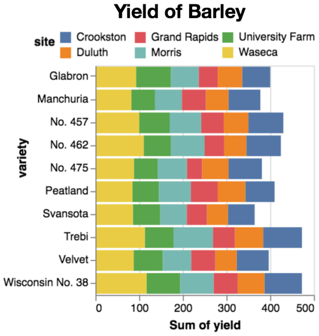
Figure A.5: Word rearrangement rules for visual explanations. First column shows the original expression that can appear in the visual explanations and the second column shows the reworded result. The bracketed expressions with two options indicate word choice when the chart is a bar chart (left) and when the chart is a line graph (right).

A.3 Additional Results: Explanations

Because the explanations generated by our system in Stage 3 are templated conversions of Sempre [127, 187]'s lambda expressions [99], the generated explanations are reasonable as long as the lambda expression output corresponds to meaningful operations. Here, we share some less meaningful explanations generated because the original lambda expression did not represent meaningful operations on charts (Figure A.6).

For Q1, our pipeline generates an explanation that simply states the answer *'Glabron'* without any operations on it. Observing the lambda expressiom, our system finds the row of the underlying data table with the *'variety'* value equal to *'Glabron'*, and obtains the *'variety'* value of that row, which is equivalent to just reporting *'Glabron'*. Because the operations in the lambda expression are very redundant, our system results in removing all the redundant operations and ends up giving the meaningless explanation.

For Q2, our pipeline generates an explanation with the word *'index'*, which is not defined with respect to the chart. This is because the lambda expression operates on the underlying data table and not the chart itself. The lambda expression indicates that it read the variety value of the last row of the table, which has no correspondence in terms of the chart because the ordering of rows in the table does not necessarily match that of the ordering of the *'varieties'* We leave better incorporation



Q1: *Which site had the lowest production of the Glabron variety?*
A: ***Glabron*** Correct: ***Grand Rapids***
λ: **R**[Variety](and(Variety.Glabron, Row))
Explanation: *I computed the 'Glabron'.*

Q2: *On average, which site produces the greatest yield across all varieties?*
A: ***Wisconsin No. 38*** Correct: ***Waseca***
λ: **R**[Variety](and(argmax(Row, Index), Row))
Explanation: *I looked up 'variety' of the greatest index.*

Figure A.6: Examples of less meaningful explanations generated by our pipeline. The first row shows the question, and the second row shows the answers generated by our system (red indicates incorrect) and the correct answer. The third row shows the lambda expression generated by Sempre and the last row shows the explanation generated by our system.

of such table-specific operations as future work.

A.4 Additional Results for User Study

In addition to the Likert scale measurements of transparency, trust and usefulness, we also measured how accurately participants determined the correctness of the provided answers, and how quickly the participants so (Table 5.2).

A.4.1 Accuracy

We did see higher accuracy when we provided answers and explanations generated by our system (98.8% with visual explanations and 95.0% with non-visual explanations) than when we provided answers generated by humans (91.3% with explanations and 87.5% without explanations). While this could be due to our explanations, this could also be due to the wrong answers by our system being more conspicuous than wrong answers generated by humans. Further study is required to determine the contributions of these factors.

A.4.2 Time Measurements

Although we measured time taken to determine the correctness of the provided answers, we did not see a significant improvement in completion times when we presented our visual explanations (μ = 26.7s, σ = 30.5s) compared to when we presented no explanation (μ = 26.2s, σ = 28.3s, $t(157)$ = 0.11, p = 0.46), human-generated explanations (μ = 26.0s, σ = 27.5s, $t(157)$ = 0.16, p = 0.44), or our non-visual explanations (μ = 23.7s, σ = 21.2s, $t(157)$ = -0.72, p = 0.76). Instead, we saw large variations in completion times in all conditions. This is probably because we did not instruct the participants to optimize for time. Other factors could be because the time required to perform the operations to confirm the answers was much greater than the time required to parse the provided answers and explanations with respect to the provided charts. Additional studies could help understand how explanations affect the speed at which people parse information.

Bibliography

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[2] Adobe Acrobat Reader. `https://www.adobe.com/acrobat/acrobat-pro.html`, 2021.

[3] Rishabh Agarwal, Chen Liang, Dale Schuurmans, and Mohammad Norouzi. Learning to generalize from sparse and underspecified rewards. In *International Conference on Machine Learning*, pages 130–140. PMLR, 2019.

[4] Bilal Alsallakh, Allan Hanbury, Helwig Hauser, Silvia Miksch, and Andreas Rauber. Visual methods for analyzing probabilistic classification data. *IEEE transactions on visualization and computer graphics*, 20(12):1703–1712, 2014.

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

[6] Amazon Mechanical Turk. `https://www.mturk.com/`, 2021.

[7] Amazon QuickSight. `https://aws.amazon.com/quicksight/`, 2021.

[8] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[9] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, pages 722–735, 2007.

[10] Paul Ayres and Gabriele Cierniak. Split-Attention Effect. In Norbert M. Seel, editor, *Encyclopedia of the Sciences of Learning*, pages 3172–3175. Springer US, Boston, MA, 2012.

[11] 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.

[12] World Bank. *World Development Report 2020: Trading for Development in the Age of Global Value Chains*. World Bank, Washington, DC, USA, 2020.

[13] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504*, 2017.

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

[15] Fabian Beck and Daniel Weiskopf. Word-sized graphics for scientific texts. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1576–1587, 2017.

[16] Albert D Biderman. The graph as a victim of adverse discrimination and segregation: Comment occasioned by the first issue of information design journal. *Information Design Journal*, 1(4):232–241, 1979.

[17] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342, 2010.

[18] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[19] Michelle A Borkin, Zoya Bylinskii, Nam Wook Kim, Constance May Bainbridge, Chelsea S Yeh, Daniel Borkin, Hanspeter Pfister, and Aude Oliva. Beyond memorability: Visualization recognition and recall. *IEEE transactions on visualization and computer graphics*, 22(1):519–528, 2015.

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

[21] John Bransford. *Human Cognition: Learning, Understanding, and Remembering*. Wadsworth Publishing Company, Belmont, CA, USA, 1979.

[22] Gary L Brase. Pictorial representations in statistical reasoning. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 23(3):369–381, 2009.

[23] Sandra Carberry, Stephanie Elzer, and Seniz Demir. Information graphics: an untapped resource for digital libraries. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 581–588, 2006.

[24] Stuart K Card, Jock D Mackinlay, and Ben Scheiderman. *Readings in Information Visualization, using vision to think*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[25] Bay-Wei Chang, Jock Mackinlay, and Polle T Zellweger. Fluidly revealing information in fluid documents. In *Proceedings of Smart Graphics 2000 AAAI Spring Symposium*, 2000.

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

[27] Charles Chen, Ruiyi Zhang, Sungchul Kim, Scott Cohen, Tong Yu, Ryan Rossi, and Razvan Bunescu. Neural caption generation over figures. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pages 482–485, 2019.

[28] Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, Tong Yu, Ryan Rossi, and Razvan Bunescu. Figure captioning with reasoning and sequence-level training. *arXiv preprint arXiv:1906.02850*, 2019.

[29] Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. Learning structured natural language representations for semantic parsing. In *55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 44–55. Association for Computational Linguistics (ACL), 2017.

[30] Jinho Choi, Sanghun Jung, Deok Gun Park, Jaegul Choo, and Niklas Elmqvist. Visualizing for the non-visual: Enabling the visually impaired to use visualization. In *Computer Graphics Forum*, volume 38.3, pages 249–260. Wiley Online Library, 2019.

[31] Imran Chowdhury, Abdul Moeid, Enamul Hoque, Muhammad Ashad Kabir, Md Sabir Hossain, and Mohammad Mainul Islam. Designing and evaluating multimodal interactions for facilitating visual analysis with dashboards. *IEEE Access*, 9:60–71, 2020.

[32] Matthew Conlen and Jeffrey Heer. Idyll: A markup language for authoring and publishing interactive articles on the web. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 977–989, 2018.

[33] Kenneth Cox, Rebecca E Grinter, Stacie L Hibino, Lalita Jategaonkar Jagadeesan, and David Mantilla. A multi-modal natural language interface to an information visualization environment. *International Journal of Speech Technology*, 4(3):297–314, 2001.

[34] Will Crichton. A new medium for communicating research on programming languages. In *Proceedings of the 1st Workshop on Human Aspects of Types and Reasoning Assistants*, 2021.

[35] Zhe Cui, Sriram Karthik Badam, M Adil Yalçin, and Niklas Elmqvist. Datasite: Proactive visual data exploration with computation of insight-based recommendations. *Information Visualization*, 18(2):251–267, 2019.

[36] D3 JavaScript Library. `https://d3js.org/`, 2021.

[37] Çağatay Demiralp, Peter J Haas, Srinivasan Parthasarathy, and Tejaswini Pedapati. Foresight: Rapid data exploration through guideposts. *arXiv preprint arXiv:1709.10513*, 2017.

[38] Kedar Dhamdhere, Kevin S McCurley, Ralfi Nahmias, Mukund Sundararajan, and Qiqi Yan. Analyza: Exploring data with conversation. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 493–504, 2017.

[39] Distill. `https://distill.pub/`, 2021.

[40] Pierre Dragicevic, Yvonne Jansen, Abhraneel Sarma, Matthew Kay, and Fanny Chevalier. Increasing the transparency of research papers with explorable multiverse analyses. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2019.

[41] Economist Graphic Detail. `https://www.economist.com/graphic-detail`, 2021.

[42] Howard E Egeth and Steven Yantis. Visual attention: Control, representation, and time course. *Annual review of psychology*, 48(1):269–297, 1997.

[43] Stephanie Elzer, Sandra Carberry, Daniel Chester, Seniz Demir, Nancy Green, Ingrid Zukerman, and Keith Trnka. Exploring and exploiting the limited utility of captions in recognizing intention in information graphics. In *Proceedings of the 43rd Annual meeting of the Association for Computational Linguistics (ACL'05)*, pages 223–230, 2005.

[44] Stephanie Elzer, Sandra Carberry, and Ingrid Zukerman. The automated understanding of simple bar charts. *Artificial Intelligence*, 175(2):526–555, 2011.

[45] Jing Fang, Prasenjit Mitra, Zhi Tang, and C Lee Giles. Table header detection and classification. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[46] Massimo Fasciano and Guy Lapalme. Postgraphe: a system for the generation of statistical graphics and text. In *Eighth International Natural Language Generation Workshop*, 1996.

[47] Leo Ferres, Petro Verkhogliad, Gitte Lindgaard, Louis Boucher, Antoine Chretien, and Martin Lachance. Improving accessibility to statistical graphs: the igraph-lite system. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 67–74, 2007.

[48] Jenny Rose Finkel, Trond Grenager, and Christopher D Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, 2005.

[49] Mareike Florax and Rolf Ploetzner. What contributes to the split-attention effect? the role of text segmentation, picture labelling, and spatial proximity. *Learning and instruction*, 20(3):216–224, 2010.

[50] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

[51] 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.

[52] Rocio Garcia-Retamero and Edward T Cokely. Designing visual aids that promote risk literacy: a systematic review of health research and evidence-based design heuristics. *Human factors*, 59(4):582–627, 2017.

[53] Rocio Garcia-Retamero and Ulrich Hoffrage. Visual representation of statistical information improves diagnostic inferences in doctors and their patients. *Social Science & Medicine*, 83:27–33, 2013.

[54] Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 123–135, 2017.

[55] Pascal Goffin, Wesley Willett, Jean-Daniel Fekete, and Petra Isenberg. Design considerations for enhancing word-scale visualizations with interaction. In *Posters of the Conference on Information Visualization (InfoVis)*. IEEE, 2015.

[56] Google sheets. `https://www.google.com/sheets/about/`, 2021.

[57] John D Gould. Looking at pictures. *Eye movements and psychological processes*, pages 323–345, 1976.

[58] Vidhya Govindaraju, Ce Zhang, and Christopher Ré. Understanding tables in context using standard nlp toolkits. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 658–664, 2013.

[59] Lars Grammel, Melanie Tory, and Margaret-Anne Storey. How information visualization novices construct visualizations. *IEEE transactions on visualization and computer graphics*, 16(6):943–952, 2010.

[60] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11287–11297, 2021.

[61] Guardian DataBlog. `https://www.theguardian.com/data`, 2021.

[62] E Dario Gutierrez, Ekaterina Shutova, Tyler Marghetis, and Benjamin Bergen. Literal and metaphorical senses in compositional distributional semantic models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 183–193, 2016.

[63] Jonathan Harper and Maneesh Agrawala. Deconstructing and restyling d3 visualizations. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 253–262, 2014.

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

[65] Marti Hearst, Melanie Tory, and Vidya Setlur. Toward interface defaults for vague modifiers in natural language interfaces for visual analysis. In *2019 IEEE Visualization Conference (VIS)*, pages 21–25. IEEE, 2019.

[66] Mary Hegarty and Marcel-Adam Just. Constructing mental models of machines from text and diagrams. *Journal of memory and language*, 32(6):717–742, 1993.

[67] Lynette Hirschman, Marc Light, Eric Breck, and John D Burger. Deep read: A reading comprehension system. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 325–332, 1999.

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

[69] Ting-Yao Hsu, C Lee Giles, and Ting-Hao'Kenneth' Huang. Scicap: Generating captions for scientific figures. *arXiv preprint arXiv:2110.11624*, 2021.

[70] Kevin Hu, Diana Orghian, and César Hidalgo. Dive: A mixed-initiative system supporting integrated data exploration workflows. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pages 1–7, 2018.

[71] 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.

[72] IBM Watson Analytics. `https://www.ibm.com/products/watson-assistant/analytics`, 2021.

[73] Harsh Jhamtani and Taylor Berg-Kirkpatrick. Truth-conditional captions for time series data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 719–733, 2021.

[74] Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3):385–435, 2015.

[75] Daniel Jurafsky and James H Martin. Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing. *Upper Saddle River, NJ: Prentice Hall*, 2008.

[76] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2018.

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

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

[79] Jan-Frederik Kassel and Michael Rohs. Valletto: A multimodal interface for ubiquitous visual analytics. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2018.

[80] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016.

[81] Azam Khan, Simon Breslav, Michael Glueck, and Kasper Hornbæk. Benefits of visualization in the mammography problem. *International Journal of Human-Computer Studies*, 83:94–113, 2015.

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

[83] 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.

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

[85] Ha-Kyung Kong, Zhicheng Liu, and Karrie Karahalios. Frames and slants in titles of visualizations on controversial topics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[86] Ha-Kyung Kong, Zhicheng Liu, and Karrie Karahalios. Trust and recall of information across varying degrees of title-visualization misalignment. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[87] Nicholas Kong and Maneesh Agrawala. Graphical overlays: Using layered elements to aid chart reading. *IEEE transactions on visualization and computer graphics*, 18(12):2631–2638, 2012.

[88] 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.

[89] Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. Neural amr: Sequence-to-sequence models for parsing and generation. In *55th Annual Meeting of the Association for Computational Linguistics 2017*, pages 146–157. Association for Computational Linguistics, 2017.

[90] Abhinav Kumar, Jillian Aurisano, Barbara Di Eugenio, Andrew Johnson, Alberto Gonzalez, and Jason Leigh. Towards a dialogue system that supports rich visualizations of data. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 304–309, 2016.

[91] Abhinav Kumar, Barbara Di Eugenio, Jillian Aurisano, Andrew Johnson, Abeer Alsaiari, Nigel Flowers, Alberto Gonzalez, and Jason Leigh. Towards multimodal coreference resolution for exploratory data visualization dialogue: Context-based annotation and gesture identification. In *The 21st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2017–SaarDial)(August 2017)*, volume 48, 2017.

[92] 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.

[93] Andrew Large, Jamshid Beheshti, Alain Breuleux, and Andre Renaud. Multimedia and comprehension: The relationship among text, animation, and captions. *Journal of the American society for information science*, 46(5):340–347, 1995.

[94] Shahid Latif, Diao Liu, and Fabian Beck. Exploring interactive linking between text and visualization. In *EuroVis (Short Papers)*, pages 91–94, 2018.

[95] 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*, 2021.

[96] Pengyuan Li, Xiangying Jiang, and Hagit Shatkay. Extracting figures and captions from scientific publications. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1595–1598, 2018.

[97] Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc Le, and Ni Lao. Memory augmented policy optimization for program synthesis and semantic parsing. *arXiv preprint arXiv:1807.02322*, 2018.

[98] Jie Liang and Mao Lin Huang. Highlighting in information visualization: A survey. In *2010 14th International Conference Information Visualisation*, pages 79–85. IEEE, 2010.

[99] Percy Liang. Lambda dependency-based compositional semantics. *arXiv preprint arXiv:1309.4408*, 2013.

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

[101] Pengfei Liu, Xipeng Qiu, and Xuan-Jing Huang. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, 2017.

[102] Shixia Liu, Michelle X Zhou, Shimei Pan, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 543–552, 2009.

[103] Shixia Liu, Michelle X Zhou, Shimei Pan, Yangqiu Song, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. Tiara: Interactive, topic-based visual text summarization and analysis. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):1–28, 2012.

[104] Alan Lundgard and Arvind Satyanarayan. Accessible visualization via natural language descriptions: A four-level model of semantic content. *IEEE transactions on visualization and computer graphics*, 2021.

[105] Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. Natural language to visualization by neural machine translation. *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[106] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[107] Laura E Matzen, Michael J Haass, Kristin M Divis, Zhiyuan Wang, and Andrew T Wilson. Data visualization saliency model: A tool for evaluating abstract data visualizations. *IEEE transactions on visualization and computer graphics*, 24(1):563–573, 2017.

[108] Hongyuan Mei, Mohit Bansal, and Matthew R Walter. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 720–730, 2016.

[109] Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1527–1536, 2020.

[110] Luana Micallef, Pierre Dragicevic, and Jean-Daniel Fekete. Assessing the effect of visualizations on bayesian reasoning through crowdsourcing. *IEEE transactions on visualization and computer graphics*, 18(12):2536–2545, 2012.

[111] Microsoft power BI. `https://powerbi.microsoft.com/`, 2021.

[112] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[113] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[114] Yao Ming, Huamin Qu, and Enrico Bertini. Rulematrix: Visualizing and understanding classifiers with rules. *IEEE transactions on visualization and computer graphics*, 25(1):342–352, 2018.

[115] Ruslan Mitkov. *Anaphora resolution.* Routledge, New York, NY, USA, 2014.

[116] Vibhu Mittal, Steven Roth, Johanna D Moore, Joe Mattis, and Giuseppe Carenini. Generating explanatory captions for information graphics. In *Proceeedings of the International Joint Conference on Artificial Intelligence*, pages 1276–1283, 1995.

[117] Tamara Munzner. *Visualization analysis and design.* CRC press, Boca Raton, FL, USA, 2014.

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

[119] Gwen C Nugent. Deaf students' learning from captioned instruction: The relationship between the visual and caption display. *The Journal of Special Education*, 17(2):227–234, 1983.

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

[121] Shaun O'Brien and Claire Lauer. Testing the susceptibility of users to deceptive data visualizations when paired with explanatory text. In *Proceedings of the 36th ACM International Conference on the Design of Communication*, pages 1–8, 2018.

[122] Alvitta Ottley, Aleksandra Kaszowska, R Jordan Crouser, and Evan M Peck. The curious case of combining text and visualization. In *EuroVis (Short Papers)*, pages 121–125, 2019.

[123] Alvitta Ottley, Blossom Metevier, PK Han, and Remco Chang. Visually communicating bayesian statistics to laypersons. In *Technical Report.* Citeseer, 2012.

[124] Alvitta Ottley, Evan M Peck, Lane T Harrison, Daniel Afergan, Caroline Ziemkiewicz, Holly A Taylor, Paul KJ Han, and Remco Chang. Improving bayesian reasoning: The effects of phrasing, visualization, and spatial ability. *IEEE transactions on visualization and computer graphics*, 22(1):529–538, 2015.

[125] Anshul Vikram Pandey, Katharina Rall, Margaret L Satterthwaite, Oded Nov, and Enrico Bertini. How deceptive are deceptive visualizations? an empirical analysis of common distortion techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1469–1478, 2015.

[126] Devi Parikh, Phillip Isola, Antonio Torralba, and Aude Oliva. Understanding the intrinsic memorability of images. *Journal of Vision*, 12(9):1082–1082, 2012.

[127] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*, 2015.

[128] Pew Research. `https://www.pewresearch.org/`, 2021.

[129] Jorge Poco and Jeffrey Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. In *Computer Graphics Forum*, volume 36. 3, pages 353–363. Wiley Online Library, 2017.

[130] Xin Qian, Eunyee Koh, Fan Du, Sungchul Kim, and Joel Chan. A formative study on designing accurate and natural figure captioning systems. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2020.

[131] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[132] Vijayshankar Raman and Joseph M Hellerstein. Potter's wheel: An interactive data cleaning system. In *VLDB*, volume 1, pages 381–390, 2001.

[133] Revanth Reddy, Rahul Ramesh, Ameet Deshpande, and Mitesh M Khapra. Figurenet: A deep learning model for question-answering on scientific plots. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[134] Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

[135] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[136] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.1, 2018.

[137] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 193–203, 2013.

[138] Steven F Roth, John Kolojejchick, Joe Mattis, and Jade Goldstein. Interactive graphic design using automatic presentation knowledge. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 112–117, 1994.

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

[140] 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.

[141] Min Joon Seo, Hannaneh Hajishirzi, Ali Farhadi, and Oren Etzioni. Diagram understanding in geometry questions. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[142] Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1476, 2015.

[143] Vidya Setlur, Sarah Battersby, and Tracy Wong. Geosneakpique: Visual autocompletion for geospatial queries. *arXiv preprint arXiv:2110.12596*, 2021.

[144] 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.

[145] 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.

[146] Vidya Setlur and Melanie Tory. Exploring synergies between visual analytical flow and language pragmatics. In *2017 AAAI Spring Symposium Series*, 2017.

[147] Vidya Setlur, Melanie Tory, and Alex Djalali. Inferencing underspecified natural language utterances in visual analysis. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 40–51, 2019.

[148] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055, 2017.

[149] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[150] Hrituraj Singh and Sumit Shekhar. Stl-cqa: Structure-based transformers with localization and encoding for chart question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3275–3284, 2020.

[151] Arjun Srinivasan, Bongshin Lee, Nathalie Henry Riche, Steven M Drucker, and Ken Hinckley. Inchorus: Designing consistent multimodal interactions for data visualization on tablet devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[152] Arjun Srinivasan, Bongshin Lee, and John T Stasko. Interweaving multimodal interaction with flexible unit visualizations for data exploration. *IEEE transactions on visualization and computer graphics*, 2020.

[153] Arjun Srinivasan and Vidya Setlur. Snowy: Recommending utterances for conversational visual analysis. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 864–880, 2021.

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

[155] Yiwen Sun, Jason Leigh, Andrew Johnson, and Barbara Di Eugenio. Articulate: Creating meaningful visualizations from natural language. In *Innovative Approaches of Data Visualization and Visual Analytics*, pages 218–235. IGI Global, 2014.

[156] Yiwen Sun, Jason Leigh, Andrew Johnson, and Sangyoon Lee. Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. In *International Symposium on Smart Graphics*, pages 184–195. Springer, 2010.

[157] John Sweller, Paul Ayres, and Slava Kalyuga. The split-attention effect. In *Cognitive load theory*, pages 111–128. Springer, 2011.

[158] Tableau Public. `https://public.tableau.com/`, 2021.

[159] Tableau Software. `http://www.tableau.com`, 2021.

[160] Editorial Team. Distill hiatus. *Distill*, 2021. https://distill.pub/2021/distill-hiatus.

[161] ThoughtSpot. `https://www.thoughtspot.com/`, 2021.

[162] Rubèn Tito, Minesh Mathew, CV Jawahar, Ernest Valveny, and Dimosthenis Karatzas. Icdar 2021 competition on document visual question answering. In *International Conference on Document Analysis and Recognition*, pages 635–649. Springer, 2021.

[163] Edward R. Tufte. *Envisioning Information.* Graphics Press, Cheshire, CT, USA, 1990.

[164] Edward R. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, CT, USA, 2001.

[165] Edward R Tufte. *Beautiful evidence*, volume 1. Graphics Press Cheshire, CT, 2006.

[166] Universal Dependencies. `https://universaldependencies.org/`, 2021.

[167] Manasi Vartak, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. Seedb: supporting visual analytics with data-driven recommendations. *Proceedings of the VLDB Endowment*, 8(13):2015, 2015.

[168] Vega-Lite Example Gallery. `https://vega.github.io/vega-lite/examples/`, 2021.

[169] Bret Victor. Explorable explanations. `http://worrydream.com/ExplorableExplanations/`, 2011.

[170] Ken Wakita and Kohei Arimoto. Guiding readers to the focus and context of industrial statistical reports. *OSF Preprints*, 2019.

[171] Mingxuan Wang, Zhengdong Lu, Jie Zhou, and Qun Liu. Deep neural machine translation with linear associative unit. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145, 2017.

[172] Washington Post. `https://www.washingtonpost.com/`, 2021.

[173] Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. Tiara: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162, 2010.

[174] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics*, 26(1):56–65, 2019.

[175] Wikipedia. `https://www.wikipedia.org/`, 2021.

[176] Graham Wills and Leland Wilkinson. Autovis: automatic visualization. *Information Visualization*, 9(1):47–69, 2010.

[177] WolframAlpha. `https://www.wolframalpha.com/`, 2021.

[178] Word embedding trained on Google News. `https://code.google.com/archive/p/word2vec/`, 2021.

[179] WorkWithColor.com color names. `http://www.workwithcolor.com/color-names-01.htm`, 2021.

[180] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 747–763, 2019.

[181] X11 color names. `https://en.wikipedia.org/wiki/X11_color_names`, 2021.

[182] Cindy Xiong, Lisanne Van Weelden, and Steven Franconeri. The curse of knowledge in visual data communication. *IEEE transactions on visualization and computer graphics*, 26(10):3051–3062, 2019.

[183] Xiaojun Xu, Chang Liu, and Dawn Song. Sqlnet: Generating structured queries from natural language without reinforcement learning. *arXiv preprint arXiv:1711.04436*, 2017.

[184] Hui Yang, Lekha Chaisorn, Yunlong Zhao, Shi-Yong Neo, and Tat-Seng Chua. Videoqa: question answering on news video. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 632–641, 2003.

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

[186] Jin Yu, Ehud Reiter, Jim Hunter, and Somayajulu Sripada. Sumtime-turbine: a knowledge-based system to communicate gas turbine time-series data. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 379–384. Springer, 2003.

[187] Yuchen Zhang, Panupong Pasupat, and Percy Liang. Macro grammars and holistic triggering for efficient semantic parsing. *arXiv preprint arXiv:1707.07806*, 2017.

[188] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.