UNIVERSITY OF CALIFORNIA SAN DIEGO

Towards understanding text-data connection in documents through the lens of data operations

A Thesis submitted in partial satisfaction of the requirements
for the degree Master of Science

in

Computer Science

by

Panayu Keelawat

Committee in charge:

Professor Haijun Xia, Chair
Professor Nadir Weibel, Co-Chair
Professor William Griswold

2022

The Thesis of Panayu Keelawat is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ML	Machine learning
LIDO	Language-Inferred Data Operation
DLIF	Data-Language Inference Framework

ACKNOWLEDGEMENTS

I would like to convey my sincere gratitude to Prof. Haijun Xia for his guidance during my time at UC San Diego. I learned a lot from conducting research at the lab. The experience has helped me become a better researcher. Without his support, this thesis would not have been possible. I would also like to thank my thesis committee members, Prof. Nadir Weibel and Prof. William Griswold, for their insights and advice.

Moreover, I would like to thank all of my collaborators, Tony Wang, Aditya Gunturu, and Jackie Kwok whom I am very fortunate to work closely with. My thanks also extend to Devamardeep Hayatpur, Zhutian Chen, Matthew Beaudouin-Lafon, Rima Cao, Jane E, Peiling Jiang and everyone I have met at the Creativity Lab for their ideas and feedback that helped shape the course of my research.

Beyond everything, I am also grateful to my family for their continuous support and understanding. I could never pursue my dreams without them. Lastly, I would like to thank all of my friends in San Diego who made my time in this beautiful city such an enjoyable experience.

This thesis, in full, is currently being prepared for submission for publication of the material as it may appear in a conference, 2023, Panayu Keelawat, Haijun Xia. The thesis author was the primary researcher and author of this material.

VITA

2019 Teaching Assistant, Department of Computer Engineering, Chulalongkorn University

2019 Bachelor of Engineering in Computer Engineering, Chulalongkorn University

2022 Teaching Assistant, Department of Psychology, University of California San Diego

2022 Master of Science in Computer Science, University of California San Diego

FIELD OF STUDY

Major Field: Computer Science

Studies in Computer Science and Engineering (Human-Computer Interaction)
Professor Haijun Xia (Cognitive Science)

ABSTRACT OF THE THESIS

Towards understanding text-data connection in documents through the lens of data operations

by

Panayu Keelawat

Master of Science in Computer Science

University of California San Diego, 2022

Professor Haijun Xia, Chair
Professor Nadir Weibel, Co-Chair

While the world is relying more on data, text descriptions of data inevitably become more prevalent. These descriptions synthesize data and highlight important things from the data for readers to better understand key takeaways. Obviously, there is a connection between data and text as it is a representation of information from the data. However, despite the surge of AI and data management research, studies on the connection between them are still lacking. Understanding the connection would not only streamline the work involving both components, but also introduce novel interaction techniques between them. Therefore, this work aims to develop a better comprehension of the connection by focusing on how each

phrase in the sentence is formed given clues from the rest of the sentence and the associated data. We collected data-rich documents and investigated how people describe data in natural language. We found that this problem is complicated because it incorporates two difficult subproblems - language and data management problems. Also, each phrase inference can be viewed as a series of data operations that can be traced back to language. Thus, we propose a taxonomy of Language-Inferred Data Operations (LIDOs) based on our collected dataset. In addition, we propose Data-Language Inference Framework (DLIF), a conceptual framework that eases the phrase prediction process by deconstructing this complex problem into five simpler steps. Examples of DLIF applications are shown with real datasets to illustrate how DLIF works.

Chapter 1

Introduction

We all live in an increasingly data-driven world. Data play a crucial role in computing, and they are collected or queried whenever we use our digital devices [51, 33]. Despite the negative effects, data in general improve our lives tremendously [19, 64]. However, managing data is a complex task that not everyone can easily cope with [9, 53, 58]. To be able to access large scale data, database management systems knowledge is required [42]. Sometimes users have to analyze data with complex operations, which can make things even more complicated. On the other hand, to be able to access small-scale data, users manage them with spreadsheets, like MS Excel [75] or Google Sheets [24]. Nevertheless, casual users typically know only the basic functionalities of these products since spreadsheet learning tends to be goal-driven and actually learning complex formulas can be daunting at times [53].

We regularly need to make sense of the data and report the findings with language, either written or spoken [61]. That is because it can be difficult for the audience to understand the data without any prior knowledge. Studies show that text descriptions of the data, or ***data sentences***, can describe data in multiple granularities [16, 40]. In a microscopic view, a data sentence reports just a tiny fraction of the entire data, but in a macroscopic view, a data sentence reports findings from the entire data. For example, the sentence "*The shop's best seller in April 2022*

was books" is a microscopic data sentence, because it focuses on one entry from the entire table. In contrast, "*Sells steadily increase each month with a 5% growth rate on average*" is a macroscopic data sentence because the sentence covers the overall trend of the data. These examples clearly show that language embeds rich information about the data regardless of the granularity level.

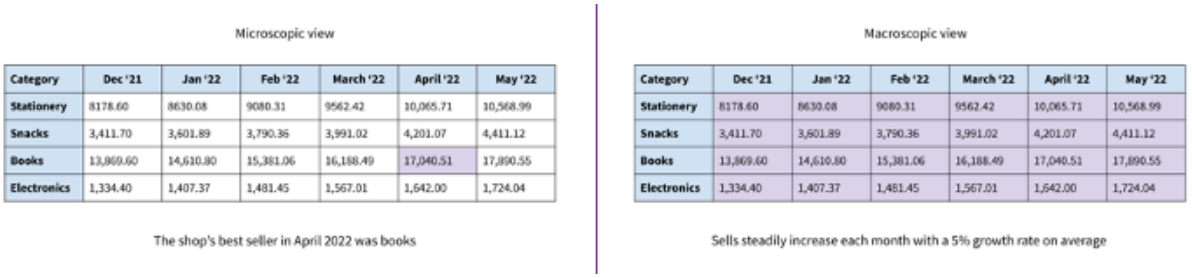

Figure 1.1. Focused area comparison between micro- and macroscopic data sentences.

Researchers have explored the usability of the bond between text and data. In particular, ***data phrases*** in a sentence are connected to the sentence's associated data. Problems such as caption generation, visualization generation, label annotation, etc. have been studied in many works [68]. While these problems are important, ***data phrase inference*** is a relatively new problem in interactive writing [12] and is highly underexplored [6]. Unlike data-agnostic inference [38], data-driven inference uses cues from the associated data together with other linguistic components in the sentence to make a prediction. We may view the result computation process as a series of data operations [12]. The problem is challenging because the task involves both linguistic and data management problems. If we look deeply into the task, we will see that we need to infer the information we need from the sentence as well as essential data operations to produce the target phrase. To our knowledge, there have been no studies that extensively scrutinize how people write data sentences and their connections to data, especially through the lens of data operations.

Therefore, in this work, we focus on building a foundational understanding of data sentences and their links to data. We collected eighteen data-rich documents from various fields and analyzed the patterns of the associated data sentences, reaching distilled break down of the task. Furthermore, we proposed a language-inferred data operation task taxonomy and potential ways to identify each operation. The taxonomy is crucial because it reflects operation intents inferred solely from language, which has not been explored thoroughly before. This work helps decouple subproblems of the task with more understanding of data phrase inference. The proposed taxonomy provides common vocabularies for researchers as well.

This work thus contributes:

1. A foundational theory of data operations behind the text-data bridging
2. A task taxonomy for language-inferred data operations
3. A conceptual framework for phrase inference based on the rest of the sentence and the corresponding data

Chapter 2

Related Work

We reviewed prior work on operations on data, text-data interactions, and natural language interfaces for data operations, as this work aims to contribute to these areas.

2.1 Operations on Data

Data play a central role in computing, so there are countless studies on data operations [15, 28]. Typical computer users may be familiar with spreadsheets as they use them to manage tabular data. Microsoft Excel [75] and Google Sheets [24] are popular choices since many worksheet functions are offered. SQL is widely adopted by technical people for managing relational databases [42]. Since there are many operations we can apply to data, researchers have come up with classifications of data operations. Wehrend and Lewis proposed eleven classes of data operations aiming to solve the so-called "cognitive tasks" [66]. Zhou and Feiner characterized data and proposed a taxonomy for visual tasks [76]. Amar et al. later put forward ten low-level analytic tasks to be a taxonomy for visualized data, which includes retrieve value, filter, compute derived value, find extremum, sort, determine range, characterize distribution, find anomalies, cluster, and correlate [3]. Brath and Hagerman reviewed quantitative sentences in various articles, and found common patterns that require computation, i.e., Comparative, Descriptive, Ranking, Savings, and Others [6]. Authors of [71] utilized these taxonomies [3, 66,

76] to construct facts that are derived from data insights. Brehmer and Munzner addressed the gap between low-level and high-level tasks by proposing a typology for multi-level tasks [7]. Very recently, possible data operations across multiple visualizations were explored in ComputableViz [67].

2.2 Text-Data Interactions

Understanding data is difficult, and it becomes more and more challenging when the data is extensive and complex [2]. Thus, when we present data in whatever form, such as tables, graphs, etc., we usually provide essential information in text description format for the readers. The conveyed message may describe the data in different levels of abstraction. Low-level descriptions correspond to a portion of the data, while high-level descriptions correspond to the entire data. Demir et al. proposed six syntactic complexity levels for summarizing information graphics [16]. Lundgard and Satyanarayan investigated natural language descriptions and classified them into four levels considering their usefulness in terms of accessibility [40].

If applied appropriately, text descriptions are useful for readers to better understand data. For that reason, researchers have been trying to generate them automatically by various means [20]. For instance, DataShot extracts information from tabular data and automatically creates fact sheets [65]. Bryan et al. proposed an approach for annotation generation in Temporal Summary Images [8]. Essentially, generating any text from data needs an understanding of the data itself, so this problem shares many common aspects with many deep learning problems, such as information retrieval [59, 60], image caption generation [5, 11, 70], etc. Therefore, some deep learning techniques can be applied to text generation from data as well. For example, AutoCaption extracts visual features from the visualization and generates captions based on pre-defined templates [39]. Chen et al. employed a deep learning model composed of ResNet,

LSTM, and Relation Maps to generate captions [10]. Obeid and Hoque used a Transformer Model to produce text from chart [45]. Lai et al. implemented Mask R-CNN to produce annotation, which is displayed next to the visualized data [34].

Connecting text and data together can benefit both authors and readers. Based on this concept, several works have focused on creating a better experience for content creation and consumption. Kong et al. proposed a pipeline for extracting references between texts and charts, and built a chart-highlighting system to test their results [31]. Kim et al. facilitated document reading by linking texts and tables with automatic referencing, making the reading more interactive [29]. Badam et al. also explored coupling text and tables for enhancing data-rich document reading in their work on Elastic Documents [4]. Goffin et al. investigated interaction techniques with word-scale visualizations that were constructed from text-data connections [22]. Sultanum et al. constructed VizFlow, which leverages text-chart links to support data-driven article authoring [57]. Dragicevic et al. presented explorable multiverse analysis reports that let readers explore alternative analysis options by interacting with the research papers directly [17]. Latif et al. presented a framework for authoring data documents that support text-data interactions [36]. The idea was subsequently extended in VIS Author Profiles [35] and Kori [37]. All these works demonstrated great potential in designing tools that leverage text-data connections.

2.3 Natural Language Interfaces for Data Operations

The capabilities of AI and NLP models have improved significantly in the last decade. Natural Language Interfaces (NLIs) [14], as a result, have also been boosted in performance and have been applied to solve many problems in databases [13, 25, 26, 48, 50, 63], visualizations [21, 44, 49, 55, 73, 74], etc. NLIs help lower barriers for casual users to work on data [1]. There

have been some works that investigated how NLIs can help with data operations. The most common operation is search query, which has been studied extensively in the field of database systems [47]. For instance, researchers from IBM introduced ATHENA, a system that translates natural language query input into SQL by using an ontology-based two-stage pipeline [52]. They later improved the system and proposed ATHENA++ [54], an end-to-end model that can tackle complex business intelligence SQL queries by adding Nested Query Detector and Nested Query Builder based on Stanford CoreNLP [41] and linguistic patterns.

Apart from a database search query, CrossData, perhaps the closest related work to this present research, leverages keywords within the focused sentence to infer data operations to facilitate data document authoring [12]. Iris identifies data science commands from conversational texts via statistical model and operates on the linked data [18]. Latif et al. used input text with pre-defined markup to apply data filtering by index and value to the associated tabular data [36]. In VizFlow, users can manually select and connect text and data for data-driven article authoring [57]. Beyond tabular data, NLIs can also facilitate data in other representations [18, 27, 30, 32, 69, 72]. Despite all these instances, how NLIs relate to data operations is underexplored [6, 52, 68]. These works clearly illustrate the promising future of NLIs for data operations.

Chapter 3

Materials and Methods

Towards understanding of the linkage between language and data, the goal is to dive deep into the data phrase inference problem. Therefore, we need to examine a good amount of data documents along with their corresponding data operations to construct a structure of the problem to decouple this complex problem into several easier ones. Collecting data operations can be challenging, because there are endless possible ways we can operate on data, and different data operations can possibly reach the same end result. We chose Google Sheets to be our main tool for data operation collection as it provides a wide range of functions for spreadsheet data management, which aligns with our goal of collecting comprehensive data operations. Another reason is that it can cascade functions into a single formula, so it becomes useful for exploring any potential formulas that can produce the data phrase we want. Even if we started with Google Sheets functions, our end results should be technology-independent, so we were not constrained to any product in the market. There are some terms that need to be introduced:

- Data sentence: A sentence that describes a portion of the associated data.
- Data phrase: A phrase within a data sentence that requires both language and data components to be formed. There may be multiple data phrases in one data sentence.

- Data document: A data-rich document that includes considerable amount of data sentences.

Essentially, there are three steps to achieve data phrase understanding via data operations. The steps involve data collection, formula and keyword generation, and data analysis.

3.1 Data Collection

When collecting data, there are two major concerns we need to address. First, the collected sentences need to reflect how people actually write in diverse domains in the real world. However, some operations will be used much more frequently than others, and, therefore, the second concern is the completeness of all possible data operations. Taking these concerns into account, we gathered sentences in two different ways:

1. Collect professionally written data documents from various public sources
2. Recruit participants to write sentences based on the given Google Sheets functions

3.1.1 Public Data Documents Collection

Data documents are documents that convey information with rich media such as tables, graphs, and other visualizations [4]. They can come from different sources, including but not limited to private government, research institutes, private companies, etc. More information about the collected documents can be found from Table 3.1. We looked for tables in each document and their associated data sentences. It is possible that there are no associated data sentences at all, and we would skip those tables. One data sentence can have multiple data phrases that we can predict.

	2004	2006	2008	2010	2012	2014
Highway Miles	971,036	984,093	994,358	1,007,777	1,005,378	1,020,461
Lane Miles	2,319,417	2,364,514	2,388,809	2,451,140	2,433,012	2,445,667
VMT (trillions)	2.532	2.574	2.534	2.525	2.527	2.572
Bridges	307,840	312,062	316,012	319,108	321,724	325,467

The number of bridges on Federal-aid highways increased from 307,840 in 2004 to 325,467 in 2014.

Figure 3.1. All possible data phrases in the sentence "*The number of bridges on Federal-aid highways increased from 307,840 in 2004 to 325,467 in 2014*."

For instance, based on Figure 3.1., the sentence "*The number of bridges on Federal-aid highways increased from 307,840 in 2004 to 325,467 in 2014*" has five data phrases: "*bridges*", "*307,840*", "*2004*", "*325,467*", and "*2014*". All of them can be inferred from other phrases in the sentence combined with information from the associated table. Please notice that "*increased*" is not considered a data phrase, because the increment property can be inferred linguistically solely from "*307,840 in 2004 to 325,467 in 2014*", disregarding the associated table. We recorded informative tables in Google Sheets. Each spreadsheet was linked to the associated sentences in another spreadsheet. With this structure, we could proceed with formula generation, whether through manual generation or an annotation system, in the following section.

3.1.2 Sentence Creation from Google Sheets Functions

To ensure completeness of the viable data operations, we analyzed all 494 Google Sheets functions and hand-picked a portion of them that are generalizable to diverse fields for sentence generation [24]. As Google Sheets shares a lot of common functions with MS Excel [75], we also looked into MS Excel functions for reference in the selection process. In the end, we selected 54 functions in total that are field-independent, the least redundant, and needed diverse inputs from people who are comfortable with English. We filtered out functions for several

Table 3.1. Collected public data documents with their sources.

Article Name	Source
Alcohol and Drug Use and Treatment Reported by Prisoners: Survey of Prison Inmates, 2016	U.S. Department of Justice
Policy and Governmental Affairs, Chapter 1: System Assets	U.S. Department of Transportation
Olympic swimming records: An American splash and a superman called Michael Phelps!	International Olympic Committee
Apple Inc. Annual Report 2021 on Form 10-K	Apple Inc.
Hate Crime Recorded by Law Enforcement, 2010–2019	U.S. Department of Justice
State and Local Law Enforcement Training Academies, 2013	U.S. Department of Justice
National Hospital Care Survey Demonstration Projects: Severe Maternal Morbidity in Inpatient and Emergency Departments	Centers for Disease Control and Prevention
COVID-19 Pandemic Pinches Finances of America's Lower- and Middle-Income Families	Pew Research Center
Levels & Trends in Child Mortality, 2015	UN Inter-agency Group for Child Mortality Estimation
Non-U.S. Citizens in the Federal Criminal Justice System, 1998–2018	U.S. Department of Justice
The world's energy problem	Our World in Data
PETA Financial Statements and Supplementary Information July 31, 2016	People for the Ethical Treatment of Animals, Inc.
Tropical Storm Arlene, 2017	National Hurricane Center
ACLU Consolidated Financial Report, 2012	American Civil Liberties Union Foundation, Inc.
U.S. Treasury Bulletin, December 2021	Department of the Treasury
Arts Credits Earned in High School and Postsecondary Enrollment: Differences by Background Characteristics	Institute of Education Sciences
COVID-19 Weekly Epidemiological Update, 16 May 2021	World Health Organization
Results of the Statewide 2017-18 California Student Tobacco Survey	Center for Research and Intervention in Tobacco Control (CRITC), UC San Diego

reasons. For example, the function is too specialized to some domains such as *BIN2DEC* (binary to decimal), *CHISQ.DIST* (left-tailed chi-squared distribution), *HARMEAN* (harmonic mean), *ERROR.TYPE* (get error type) etc. Another reason could be that the function was too close to other existing function such as *AVERAGEA* and *AVERAGE* (not ignore vs ignore text), *IF* and *IFNA* (general case vs N/A case handling).

Twelve participants were recruited to attend two-hour virtual writing sessions. Before participation, they were given a questionnaire about their background and English proficiency. Everyone was an undergraduate student at UC San Diego from various fields such as Psychology, Computer Science, Cognitive Science, etc. None of them had a learning disability. All of them were comfortable with English as they were native or otherwise needed to pass the English Proficiency Requirement[1] to be able to attend the university. Each of them was given a unique Google Sheets link together with a list of Google Sheets functions (10 – 12 functions depending on complexity) and data-rich PDF links. References for the given functions were also provided. In the session, the instructor would explain the goal of the research and demonstrate how to complete the task before letting them work on the task. After the demonstration, the instructor would let participants work on their given task. For each function, a participant needed to generate up to five that utilize the given function for predicting a phrase in the sentence. They were encouraged to combine the functions if that could produce sentences that sounded more natural. They were allowed to use any tables from the PDF links, but they had to provide table numbers and data entries used in the table. Moreover, they needed to specify the target answer that they wanted to predict, part of speech of the target, keywords that infer the function usage, and Google Sheets formula for sanity check. Participants were allowed to ask the instructor for

[1] https://admissions.ucsd.edu/international/index.html

clarification during the session. While they were working on their task, the instructor could view their answers live to verify the correctness of the answers. After successfully completing the session, participants were granted two credits for their research participation requirement[2].

3.2 Formula and Keyword Generation

Bridging language and data is the core contribution of this research. As mentioned in the Introduction section, we view this connection through the lens of data operations. We would like to know what operations are needed to correctly infer each data phrase and how we could identify data operations from language. Hence, the Google Sheets formula and keywords from each sentence can be a good representation of the idea. There may be criticism about whether keywords can be an effective portrayal of data operation identifiers as people can write implicit sentences that have no specific keywords but still have inferable data phrases. An example sentence is "*Athletes with fastest Butterfly Olympic swimming records are Caeleb Dressel, Kristof Milak, Sarah Sjostroem, and Zhang Yufei*", which we sort the names in alphabetical order but there is no explicit keyword in the sentence indicating sorting. We would like to confirm that the concern is valid, and you can see the discussion details about this issue in the Results and Discussion section. Despite such concern, we believe that the keywords still provide plentiful information for data operation identification and the results should be useful, nevertheless. In this step, we generated formulas and keywords from three approaches.

1. Manual generation
2. Collection from virtual writing sessions
3. Data collection system

[2] https://psychology.ucsd.edu/undergraduate-program/undergraduate-resources/sona/index.html

Although we could possibly handle the formula generation entirely by ourselves, language is inherently ambiguous and, therefore, we would like to see other perspectives on forming a formula because there could be multiple ways to reach the same result.

3.2.1 Manual Generation

Manual generation was the first approach we used in this research. By looking through the collected public data documents, generating formulas by ourselves helped create a deep understanding of the problem. This understanding helped us formulate our thoughts for the analysis in the latter section. For each target data phrase, we constructed a formula that could produce the target if doable. We took notes if we found new keywords or new functions. However, sometimes the existing Google Sheets functions could not possibly produce the exact same answer we were looking for. For example, for spelling out numbers, we would type out an explanation of the required operations instead as well as take notes of the program's limitations. If the format is repeated, we would leave a comment that the format had already been seen.

3.2.2 Collection from Virtual Writing Sessions

This collection is a continuation of the Sentence Creation section. In the virtual writing sessions we conducted, all twelve participants needed to provide a formula and keywords for each sentence they produced. They were encouraged to construct complex formulas that cascade multiple functions together. They were allowed to copy a portion of table values in the PDF to the work spreadsheet to experiment formula construction with data references. Compared to formulas from public data documents, we obtained more diverse formulas from this approach as we instructed participants to create formulas with the given functions that might not be used frequently in typical data-driven authoring such as *STDEV*, *PERCENTILE*, etc. Please see Figure 3.2. for a summary of the collection process from Virtual Writing Sessions.

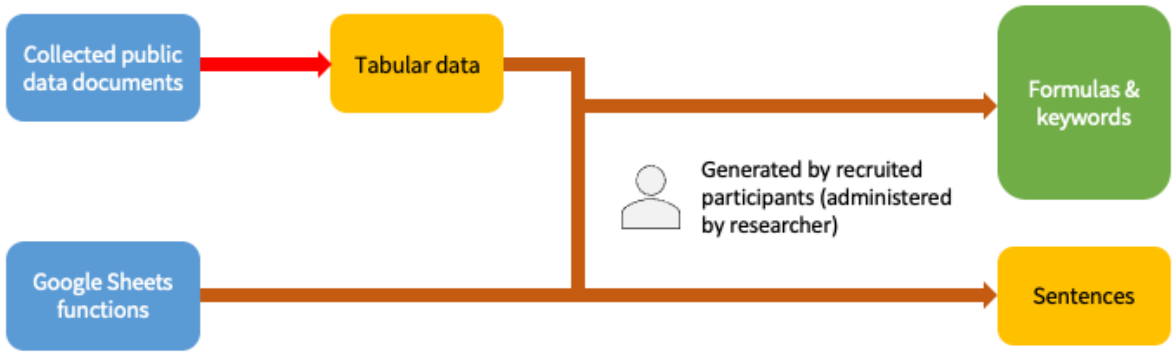

Figure 3.2. Data collection process of sentence generation based on existing Google Sheets functions.

3.2.3 Workforce Recruitment for Public Data Documents

Although we have partially generated formulas and keywords by manually looking through the collected data documents ourselves, we needed more perspectives of how people interpret language to form a formula. Besides, there were many data documents, so getting assistance on formula generation would be nice. Thus, we developed a formula-keyword recording platform using ReactJS [43] and NodeJS [46]. We parsed the resulting spreadsheets from the Public Data Documents Collection section to JSONs and uploaded them to Firebase [23]. The recording system was connected to Firebase to store formulas and keywords results. The interface of the system was separated into two parts, data and language. The data part was basically a table on Google Sheets in an iframe, while the language part was the sentence prompt with keyword annotation capability. These two parts were connected via Sheets API[3], which was useful for sharing information between them.

Five undergraduate students were recruited. A questionnaire similar to previous section about their background and English proficiency was given to them before starting the task. Then, participants were given instructions about how to use the system along with resources

[3] https://developers.google.com/sheets/api

about Google Sheets functions. We held several sessions for participants to join if they have any questions regarding the task. Each participant had to finish twenty tasks, which should take approximately two hours, in order to grant two research credits. The typical workflow was:

1. Select a target phrase to work on.

2. Experiment with formulas on Google Sheets based on the given sentence and data.

3. If the formula produces the correct output, apply the formula in the first row of the spreadsheet and the input form in the language part.

4. Click Apply formula, then click words in the sentence to map formula components to keywords and roles (Operation, Parameter, and Output).

5. Click Submit.

The system would perform input validation. For instance, at least one component has to be Parameter from table data entries. Another example would be if the user did not apply or applied formula incorrectly in Google Sheets, the system would not let the user submit the

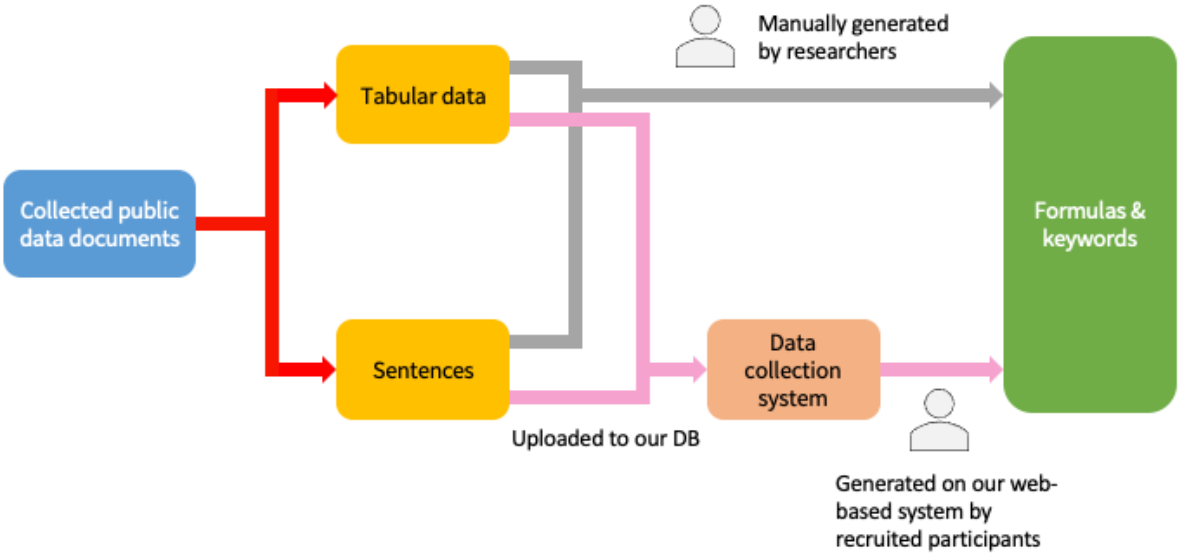

Figure 3.3. Data collection process of formula generation based on the collected data documents. The figure combines two approaches, Manual Generation and Workforce Recruitment.

answers. If the user passed all the test cases, the formula and keywords would be stored in Firebase, otherwise an alert would show error message. We would validate the inputs to ensure the input quality before granting them research credits. As this approach utilizes same materials as the Manual Generation approach, a summary of both 3.2.1 and 3.2.3 approaches can be found from Figure 3.3.

3.2.4 Summary

In total, we collected 240 data sentences and 580 data phrases. Table 3.2. demonstrates the numbers of collected data sentences and phrases from all approaches. Most data phrases were from Manual Generation contributing 328 phrases, while Virtual Writing Sessions gathered most of the data sentences of 130 sentences. Typically, one data sentence has several data phrases, so the number of the collected data phrases is much higher than the number of data sentences. For Virtual Writing Sessions, however, the method collected one data phrase per sentence because participants focused on the given function, and recorded just the objective phrases.

Table 3.2. Summary of data collection of all approaches.

Method	**#Data sentences**	**#Data phrases**
Manual Generation	85	328
Workforce Generation w/ Web-based System	25	122
Virtual Writing Sessions	130	130
Total	**240**	**580**

3.3 Data Analysis

Data collection process takes time, so we concurrently analyzed the data we had at hand while collecting more data. Since Google Sheets is a product that has been used by many professionals for a long time, we used Google Sheets functions as our initial set of data operations. Constructing formulas on Google Sheets should be versatile enough to cover most of potential operations inferred from language. We iteratively refined the functions by merging similar functions, adding higher-level functions or language-related functions, etc. as we analyzed more data depending on factors such as the core intention of the writer, grammatical correctness, etc. Even though we started with Google Sheets functions, the end goal is to construct Language-Inferred Data Operations (LIDOs) that is technology-independent. We furthered classified LIDOs into categories and recorded keywords used to identify LIDOs.

When we examined more data, we began to get overwhelmed by possible operations. That is because there are endless ways to describe data with language. Thus, we dissected formulas into smaller steps and formed a structure to better explain the results. This becomes one of the main findings in this research, and we call it Data-Language Inference Framework (DLIF). The steps included Operations Identification, Data Entries Location, Operation Inputs Retrieval, Execution, Text Parsing. More details are discussed in the next section.

3.3.1 Constructing LIDOs

Based on 54 Google Sheets functions we initially were working with, as we got some formulas and keywords, we merged even more functions with similar outcome together. For instance, we merged *STDEV*, *STDEV.P*, and *STDEV.S* (standard deviation of population vs samples) together as a single *STDEV* operation, because according to the language-level intent, the writer of that sentence wants to find standard deviation, mostly ignoring details of the

calculation. Another example would be combining *PERCENTILE*, *PERCENTILE.INC*, *PERCENTILE.EXC*, *QUARTILE*, *QUARTILE.INC*, and *QUARTILE.EXC* together as a single *PERCENTILE* operation since they had similar mechanism of getting value from position of the data point.

We also had to create high-level operations if necessary. These operations are abstract, but better capture intent inferred from language in the sentence. For example, from the captured screen system prompt Figure 3.4. "*Federal-aid highways constitute just 24.3 percent of the Nation's roadway mileage, but carry 84.6 percent of the Nation's VMT*", the Google Sheets formula for predicting "*24.3*" that we collected was *=ROUND((*D6 */* E6*)*100, 1)*, where *D6* and *E6* were data entry positions in the table. The formula gave the right answer, but the operations were too low-level. According to the sentence, the intent was to find percentage of Federal-aid highway mileage (D6) out of the whole Nation's road mileage (E6). It would be more reasonable to define *PERCENTAGE* as a high-level operation to capture intent of finding the percentage. Inside *PERCENTAGE*, there would be an implementation of finding percentage including division and multiplying by 100.

Another challenging part of LIDOs construction is the keyword analysis. That is because we could not use the keywords we collected directly due to ambiguity in language. Keywords for operations were also mixed with parameters. For instance, people may be confused about the sentence "*From April 16th to 17th at 6AM, the latitude of the Tropical Storm Arlene changed by 2.6*" predicting "*2.6*" as they think "*April*", "*16th*", "*17th*", "*latitude*", "*changed*", and "*by*" are keywords for identifying *DIFFERENCE*. However, "*changed*" is the only keyword that maps to *DIFFERENCE*, and other words are parameters and preposition. The reason is that "*changed*"

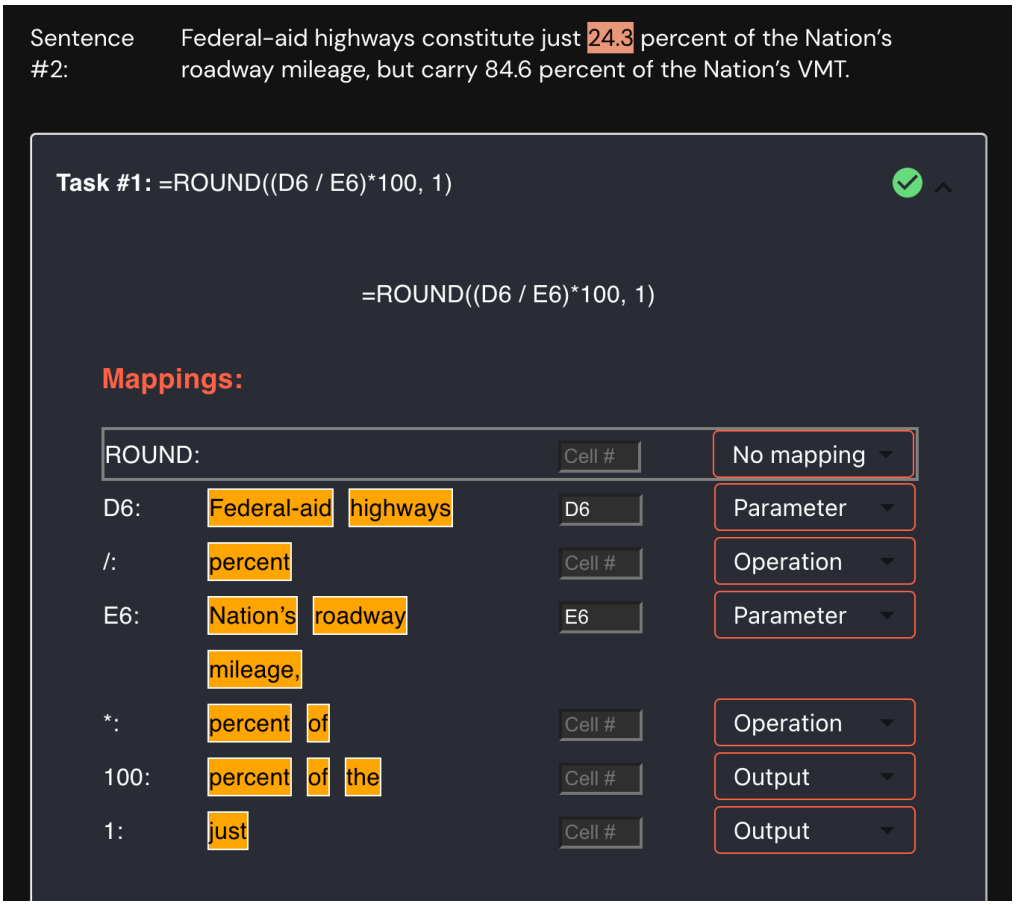

Figure 3.4. Example of combining low-level functions to high-level *PERCENTAGE* operation.

implies we are talking about differentiating new and old values, so semantically "*changed*" maps to *DIFFERENCE*.

3.3.2 Constructing DLIF

Gathered Google Sheets formulas represent series of LIDOs. Analyzing LIDOs is challenging for many reasons. First, there are many LIDOs applied in one prediction, which makes the calculation complex. Second, operations such as *FILTER*, *VLOOKUP*, etc. tend to be skipped because it would be too difficult to write. Participants usually replace these functions with hardcoded strings or data positions instead. Moreover, different series of LIDOs can reach the same answer.

Therefore, we extracted patterns in each formula and tried to find structure that makes sense calculation-wise and language-wise. We synthesized all collected formulas and spotted essential steps to complete the calculation process.

Chapter 4

Results and Discussion

For profound comprehension of the problem, we need to know the scope of the work first. Constructing taxonomy of Language-Inferred Data Operations (LIDOs) based on our collected documents is important for setting up our problem. Also, it would be easier to track the completeness of the work by relating our resulting taxonomy with related literatures [3, 66, 71]. We also present some examples of LIDOs in our collected sentences with keywords for identifying LIDOs. The problem itself involves many subtasks, so we deconstructed the problem into five different steps. We may establish Data-Language Inference Framework (DLIF), a conceptual framework which tackles the problem with multiple smaller steps. We would like to define syntax for discussing the data phrase inference problem. In a data sentence, the target phrase would be underlined. For example, "*The largest amount that was included to find the total balance of assets is $22,077,814*" has the target answer of "*The largest*". In one data sentence, there could be multiple targets, so only the focused target will be underlined. If we are discussing about LIDO Identifiers of that sentence, the keywords will be in bold. For instance, "*From 2011 to 2012, the total investments and cash equivalents* ***summed up to*** *550,417,020*" has "*summed*", "*up*", "*to*" as LIDO Identifiers for *SUM* operation.

4.1 Language-Inferred Data Operation Taxonomy

Even though there are several proposed taxonomies [3, 66, 71, 76], none of them are referred from natural language. Therefore, it is important to construct task taxonomy that fits this problem. We analyzed the formulas obtained by the data collection process, and subsequently established a taxonomy for LIDOs. LIDOs can be both low-level and high-level tasks as long as they can be inferred from the intention of the associated data sentence.

4.1.1 Resulting Task Taxonomy

Table 4.1. Resulting task taxonomy of LIDOs with example functions.

Category	**Definition**	**Example Functions**
Arithmetic Operations	Operations that are used with numeric values to perform common mathematical calculation.	*DIFFERENCE*, *PERCENT*, *DIVIDE*, *ADD*, etc.
Logical Operations	Operations that combine one or multiple conditional statements producing *TRUE* or *FALSE* value.	*AND*, *OR*, *NOT*, etc.
Comparison Operations	Operations that compare two values producing *TRUE* or *FALSE*.	*EQ*, *GT*, *GTE*, *LT*, *LTE*, etc.
Set Operations	Operations that apply operations from set theory to data table producing a new set of values.	*UNION*, *INTERSECTION*, *COMPLEMENT*, *DIFFERENCE_SET*, etc.
Statistical Operations	Operations that pertain on a collection of data.	*RANK*, *COUNT*, *STANDARDIZE*, *SUM*, *AVERAGE*, *STDEV*, *CLUSTER_NUM*, etc.

Table 4.1. Resulting task taxonomy of LIDOs with example functions (Cont.).

Pinpointing Value	Operations that give a representative value from a collection of data.	*KTH_LARGEST*, *MAX, MIN*, *PERCENTILE*, *COL_NAME*, *ROW_NAME*, etc.
Text Generation	Operations that generate text, usually in use with Comparison Operations.	*TREND_VERB*, *TREND_NOUN*, *CORRELATION_ADJ*, *SUPERLATIVE_ADJ*, etc.
Filter	An operation that locates focused entries in data.	*FILTER*
Reordering Values	Operations that reorganize sequence of values	*SORT*, *ARRANGE*, etc.
Determining Range	Operations that find a span of values within data.	*TIME_RANGE*, *VAL_RANGE*, etc.
Formatting Operations	Operations the transform a value from one form to another.	*PROPER*, *SPELL_OUT_NUM*, *PARSE_TEXT*, *PARSE_NUM*, etc.
Specialized Operations	Operations that are used in specific fields, sometimes with external knowledge.	*DEC2BIN*, *COS*, *GRADE_CALC*, *COMPUTE_TAX*, etc.

4.1.2 LIDOs with Keywords Example Discussion

There are many LIDOs, so, in this paper, we will discuss some interesting ones. Regarding *DIFFERENCE*, the operation finds the absolute difference between two values. Example data sentences with targets and LIDO Identifiers are "*From April 16th to 17th at 6AM,*

the latitude of the Tropical Storm Arlene ***changed*** *by 2.6*", "*The whole experiment* ***took*** *1* ***week***", "*The first date of the study was 5* ***years ago***", etc. Please notice that there can be various LIDO Identifiers that infer absolute difference. Some data sentence like "*The whole experiment* ***took*** *1* ***week***" has to include not only the verb but also the unit "*week*" because the word implies that we need to find numeric value in front of "*week*". This investigation can be useful for identifying *PERCENTAGE* as well with an example sentence like "*Total bridge deck area grew at an average annual* ***rate*** *of 1.2* ***percent****, while bridge crossings (measured as annual daily traffic) increased at an average annual rate of 0.9 percent.*"

Almost all the time, Logical Operations are part of *FILTER*. However, there are some occasions where we use one of the Logical Operations as the main operation of the sentence. For instance, "*It is true that the number of reported hate crimes based on race* ***and*** *religion increases from 2010 to 2018*" (*AND* operation). This finding applies to Comparison Operations where most of them are used by *FILTER*, e.g., "*The percentage of federal prisoners reporting heroin use in the 30 days prior to arrest remained the same between 2004 and 2016,* ***at*** *about 4%*" (*EQ* operation).

Set Operations are useful for obtaining data entries. They can be used to merge multiple results from *FILTER*. One interesting use case of the set property is to find unique values from the selected data entries in the table. For instance, "*The* ***unique categories*** *of expenses are salaries, employee benefits, rent*" will use *UNION* to combine all entries in the expense category column, resulting unique values of expense categories.

Text Generation operations are essential for constructing words that cannot be extracted from data. Generally, nouns can be found in the corresponding data from column names or row names. However, verbs, adjectives, adverbs, etc. cannot be discovered directly from tables, so if

we wish to predict these kinds of words, we need to generate text based on certain conditions. That is the reason why we define this category of LIDOs. Mostly these high-level operations were derived from Google Sheets function's *IF* as the function can output text depending on the condition. Examples are such as "*The number of bridges on rural local roadways decreased by the largest amount, **from** 208,641 bridges in 2004 **to** 203,995 in 2014*" (*TREND_VERB*), "*Access of electricity is related to GDP per capita*" (*CORRELATION_ADJ*), etc.

FILTER is a foundational operation that is applied in every formula. If the data sentence is microscopic, obviously we need to filter out some unrelated data. On the other hand, for macroscopic text description, we can view that *FILTER* does not filter out any data entries, meaning all entries in the table are qualified. Therefore, *FILTER* is a base operation in every formula. More details are addressed in DLIF in the next section.

Another interesting category are the Formatting Operations. Once the system derives the prediction, it needs to output data phrase that complies with the language etiquette as well. For example, in some formal writing, the author needs to spell out numeric values if the value is less than one hundred. In some scientific article, we need to preserve significant digits in the report. That is why we need operations to handle these situations. For instance, "*Ninety-six percent of undocumented non-U.S. citizens, 57% of documented citizens, and 51% of U.S. citizens were not released pretrial (table 5)*" (*SPELL_OUT_NUM*). If there is no special need, we can parse value to text directly, e.g., "*Sentences received by non-U.S. citizens were typically shorter than those for U.S. citizens*" (*PARSE_TEXT*).

4.1.3 Application of Operations

LIDOs are normally used not evenly. Some operations are used in all data phrase predictions, but some are quite rarely used. This is because of the nature of writing; we do not

describe clusters of data often, but we may compare values in almost every writing. Therefore, in the next section, we introduce a framework for tackling the data phrase inference problem, and LIDOs involved in each step are different. More information can be seen there.

4.2 Data-Language Inference Framework (DLIF)

Based on the resulting taxonomy in the previous section, we recognized that there are some operations that involve only calculation from data, while some operations involve both language and data to accomplish the tasks. It would be difficult to build a system with this complex basis, so we decided to establish a conceptual framework DLIF to simplify the problem into five smaller steps: Operations Identification, Data Entries Location, Operation Inputs Retrieval, Execution, Text Parsing.

4.2.1 Operations Identification

This process is perhaps the most important step, because it determines all of the steps afterwards. It tries to answer what are the operations the system needs to take to achieve the correct phrase. It is possible to have more than one operation per prediction. From our investigation, in some cases, we can simply use keyword matching to identify operations. For example, the sentence "*From April 16th to 17th at 6AM, the latitude of the Tropical Storm Arlene changed by 2.6*", if we have "*2.6*" as the target, then it is clear that we need to calculate the absolute difference between latitudes on April 16th and 17th at 6AM, which the associated data are visualized as Table 4.2. We can identify the function *DIFFERENCE* by the keyword "*changed*", because "*changed*" implies we are interested in their absolute difference. In the case when "*changed*" does not refer to numerical change, such as "*Ten percent informed us that their diet habits changed by routinely checking the calories*", would not be suitable to *DIFFERENCE*.

Table 4.2. The associated data table of the example "*From April 16th to 17th at 6AM, the latitude of the Tropical Storm Arlene changed by 2.6.*"

Date/Time (UTC)	Latitude (°N)	Longitude (°W)	Pressure (mb)	Wind Speed (kt)	Stage
16 / 0600	35.8	50.3	992	55	extratropical
16 / 1200	35.1	49.5	989	55	"
16 / 1800	34.4	48.7	989	55	"
17 / 0000	33.7	47.8	987	50	"
17 / 0600	33.2	47.0	988	45	"
17 / 1200	32.7	46.1	989	45	"
17 / 1800	32.3	45.3	991	40	"
18 / 0000	32.1	44.7	993	40	"
…	…	…	…	…	…

Another feature that can be extracted to help identify the correct data phrase is to use constituency tree [12]. One example is the sentence "*Men's swimming 4x100m record is 10.80s faster than Mixed.*" If we want to predict "*Men's*", we can look up at the table and see what record is faster than Mixed 4x100m, and the answer is Men's record using 3:26.78s. Another phrase we can predict from this sentence is "*10.80s*" as we can infer from differentiating Men's and Mixed record times. For the first operation predicting "*Men's*", the main operation is to look at the row name *ROW_NAME*, while for the second operation prediction "*10.80s*", the main operation is *DIFFERENCE*. However, both have the same keyword for operation identification that is "*faster than*". Other words are context-dependent, and are not transferable to other sentences, so "*faster than*" is the only keyword for data operation mapping in this case. If we parse the sentence with Stanford CoreNLP [41], we can obtain more information that can help the prediction. We can notice that although both "*Men's*" and "*10.80s*" are nouns, but "*Men's*" is a subject ("*swimming 4x100m record*") complement, while "*10.80s*" is a verb ("*is*") complement.

It is more likely that we need to look up either column or row name for predicting a subject complement, because for a verb complement tends to supplement actions; therefore, numeric values tend to show up more to complement verbs. Nevertheless, this method does not work 100% of the time. Extracting more features should help make the prediction more accurate. Also, if the associated table had meta data about the data it contains, it would help the decision as well. For instance, the system could look for year columns of data associated to "*The number of bridges on Federal-aid highways increased from 307,840 in 2004 to 325,467 in 2014*", because it could infer "*in*" as a proposition for temporal values, therefore looking up year columns in the data makes sense.

Table 4.3. The associated data table of the example "*Men's swimming 4x100m record is 10.80s faster than Mixed.*"

Event	Time	Name	Olympics
Men's 4x100m	3:26.78s	Ryan Murphy, Michael Andrew, Caeleb Dressel, Zach Apple (USA)	Tokyo 2020
Women's 4x100m	3:51.60s	Kaylee McKeown, Chelsea Hodges, Emma McKeon, Cate Campbell (Australia)	Tokyo 2020
Mixed 4x100m	3:37.58s	Kathleen Dawson, Adam Peaty, James Guy, Anna Hopkin (Great Britain)	Tokyo 2020

4.2.2 Data Entries Location

After identifying the needed operations, we can now know what information we have to look for in the data. In a single sentence, there is at least one Data Locator that explicitly or implicitly refer to either rows, columns, or conditions in the associated table. Data Locators can

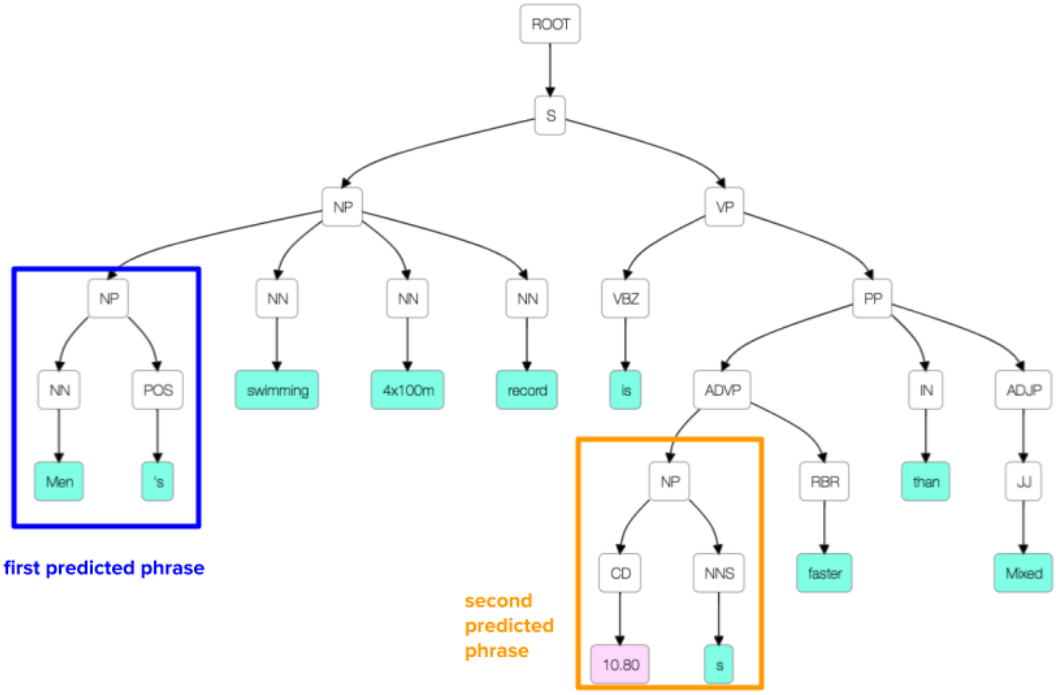

Figure 4.1. Constituency tree of the sentence "*Men's swimming 4x100m record is 10.80s faster than Mixed.*" The figure highlights the two predicted phrases based on the same keywords to illustrate how the tree can be useful to distinguish in this case. Figure was generated using nlpviz[4].

be formed into a set:

$$\{< rows >, < cols >, < conditions >\}$$

where $< rows >$ specifies a list of rows involved, $< cols >$ indicates all of the focused columns, and $< conditions >$ is a list of all conditions for filtering the data. By default, $< rows >$ and $< cols >$ would be valued *ALL_ROWS* and *ALL_COLS* respectively, which means that all rows and all columns are involved in the calculation. For instance, the sentence "*Mac sales in Americas increased during 2021 compared to 2020*" has $< rows > = Americas$, $< cols > = ALL\_COLS$, and $< conditions > = \text{“} > \{Americas, 2020, \text{“”}\}$". Please see the illustration for better understanding.

[4] http://nlpviz.bpodgursky.com/

This example produces $153306_{[B2]}$ as a result, which means value of 153306 at cell location B2 (second column, second row). There are times that the sentence includes complement phrases (such as noun phrase, adverbial phrase, etc.). In that case, we may view the sentence as a hierarchy, and we can apply the framework to the complement phrases before applying to the main sentence. This will create a cascaded formula as a result.

Figure 4.2. Illustration of the rows, columns, and conditions inside the example sentence "*Mac sales in Americas increased during 2021 compared to 2020.*"

This step is an NLP problem that also ties to data configuration. That is because based on our collected dataset, there are many occasions that the words in the sentence do not exactly match the column or row names. The authors of the documents may use synonyms to make the writing not boring, or use acronyms that may or may not be defined in the article. Recognizing conditions is also challenging. Conditions can involve not only just logical operations, but also arithmetic operations. For instance, the sentence can be about comparing average values of each row, and so we need to compute average values before making logical comparison. To handle

these challenges, we can solve by addressing two problems, NLP problem and design problem. For the NLP problem, we can use the same technique proposed in the previous step (capture keywords or use constituency tree), or apply more advanced ML techniques for deep language understanding. Regarding the design problem, we can simplify the Data Locator identification by designing a data management system that knows the context of the data it contains. For the example mentioned above, it would be great if the system knows that all the columns are years, which are temporal values. So, thus, when the sentence uses the word "*during*", the system can know that it needs to look at the columns because the columns are years.

4.2.3 Operation Inputs Retrieval

Once we are able to locate the data that we need from the table, we need to retrieve those values. In most cases, we can use the values in the table right away. Nevertheless, there are cases when we need to preprocess the values before using them. For example, if we would like to retrieve dates and times from the table, we probably need to parse the datetime string in the table into a form of datetime object. The system should still keep track of the original locations of the cell as we may need to refer back to its row or column names later. The main challenge of this step is how the system knows what preprocessing needs to be done, which can potentially be solved by contextualizing data by design.

4.2.4 Execution

We already have all the things we need from previous steps. Hence, this step performs the calculation. While in most cases there will be only one output, it is plausible that there are multiple outputs. The outputs are resulting values attached with their original locations in the table. If the obtained DLIF is cascaded, the inner operations should be executed before the outer ones.

4.2.5 Text Parsing

This step parses the outputs to text, which highly depends on the type of document. One example is that if we obtain a number with decimal points from the execution, we need to round the value to the appropriate decimal points. Some scientific article might prefer us to include all significant digits, but in typical articles we can perhaps just round to two decimal places. To our knowledge, there is no such tool that does what we described. It would be great if researchers can design a tool that takes configuration, text templates, etc. as inputs for text parsing.

Chapter 5

Future Work and Conclusions

This work deconstructs the complex problem of linking language and data into smaller steps that are easier to solve. As the world is getting more connected than ever, there are abundance of data to be collected and reported through language. Although this work focuses on written language, our results can be applied to spoken language as well. There may be differences, like in terms of vocabulary, structure complexity, formality, etc., between written and spoken language, but the framework should remain the same. Feasible usage scenarios including but not limited to language-initiated dynamic data visualization, speech-enabled interactive charts, expanding word-scale visualization techniques, auto-generated data query results from SQL, data operation animation via language, etc. Based on this research, there are many ways we can realize and improve the current results. We propose four possible future directions.

5.1 Exploring Documents in Specialized Fields

We have examined almost all Google Sheets functions in this project, except specialized formulas in certain professions such as finance and engineering. As a result, we did not explore documents that use advanced math operations. Examples of these operations include computing cosine of an angle provided in radians (*COS*), calculating the right-tailed chi-squared distribution (*CHIDIST*), etc. Professionals that use these functions deal with much more complex

calculations than basic functions presented in this paper. It can be interesting to see how this concept can be helpful for them. Also, designing tools to facilitate their language-related workflow would be a good trajectory as well.

5.2 Machine Learning Techniques for Identifying Operations

According to the proposed framework, identifying data operations is purely an NLP problem as it does not involve any data management problems. Language can be obscure due to its nature. Solely looking at the keywords and the constituency tree is not sufficient to precisely identify all possible operations. In addition, there can be some false positives in the prediction as well given the system lacks features to take into consideration. Thus, a probable solution is to apply more sophisticated machine learning (ML) techniques to better capture complex and implicit features in the sentence, and better guess the needed operations for data phrase prediction [18]. Works in NLP problems such as language understanding [62], SQL query generation [74], natural language question answering [56], etc. can be modified and applied to this problem since the main problem is to really understand what should be the intent of the sentence, either implicit or explicit, so that we can accurately predict data phrases.

5.3 Adding Context to Data

As we are connecting data and language, one challenge that pops up is the lack of context of the data. We normally treat every data the same with the same available set of functions. That is because connecting language and data is a novel task. Hence, data operations in any data management systems are designed just for data management, being agnostic to the context of the data they contain. If we can add some contexts to the data by any means, it should help connecting language and data tremendously. For example, if the table columns are months in a year, we normally use temporal propositions, such as "*in*", "*during*", or "*from*" in the sentence to

describe insights during that time. By knowing that the columns are temporal values can help the system easier locate information inside the database. We think that the problem can be solved by both ML solution and design solution. In particular, we can perhaps know the data types by using ML models, or we can also obtain data types by designing a system that receive inputs from the user specifying data types. We plan to explore both of these approaches in our future study.

5.4 Beyond Tabular Data

We studied language and data relationship based on tabular data. However, the idea is generalizable to non-tabular data as well, such as graphs, charts, etc. Functions that are tightly coupled with the table format, like *COL_NAME*, will not be applicable to data in other representations. Also, even though we have reviewed a lot of data documents and sentences in this work, we may discover new data operations if we look into other representations of data and their associated texts. For this reason, we would like to survey text-data connection in a broader perspective in our future work. The results should be format-independent, and that should cover a broader audience.

5.5 Conclusions

In conclusion, we investigated data sentence and their connections to data to form a better understanding of their relationship through the lens of data operations. We have found that the problem is complex as it is a mixture of language and data management problems. Thus, we proposed a conceptual framework DLIF to tackle the problem more efficiently by deconstructing this problem into five steps including Operations Identification, Data Entries Location, Operation Inputs Retrieval, Execution, and Text Parsing. Each step has its own challenges. Even though we still find the problem challenging after dissecting it into smaller steps, we can now solve the problem in a more structured way. In addition, researchers with different expertise can solely

work on the parts they are comfortable with, for instance, some parts require machine learning solutions, but some parts require design solutions. This problem is important as the world is getting more data-driven, and this work can be a basis of various applications. We believe that the problem is worth solving, so we can explore novel user interfaces in the future.

This thesis, in full, is currently being prepared for submission for publication of the material as it may appear in a conference, 2023, Panayu Keelawat, Haijun Xia. The thesis author was the primary researcher and author of this material.

Bibliography

[1] Katrin Affolter, Kurt Stockinger and Abraham Bernstein. A comparative survey of recent natural language interfaces for databases. *VLDB Journal*, 28, 5 (2019), 793-819. https://doi.org/10.1007/s00778-019-00567-8.

[2] Syed Mohd Ali, Noopur Gupta, Gopal Krishna Nayak, and Rakesh Kumar Lenka. Big data visualization: Tools and challenges. In *Proc. of IC3I*. IEEE. https://doi.org/10.1109/IC3I.2016.7918044.

[3] Robert Amar, James Eagan, and John Stasko. 2005. Low-level Components of Analytic Activity in Information Visualization. In *Proc. of InfoVis*. IEEE, 111–117. https://doi.org/10.1109/INFVIS.2005.1532136.

[4] Sriram Karthik Badam, Zhicheng Liu, and Niklas Elmqvist. 2019. Elastic Documents: Coupling Text and Tables through Contextual Visualizations for Enhanced Document Reading. *IEEE TVCG* 25, 1 (2019), 661–671. https://doi.org/10.1109/ TVCG.2018.2865119.

[5] Shuang Bai and Shan An. 2018. A survey on automatic image caption generation. *Neurocomputing*, 311 (2018), 291-304. https://doi.org/10.1016/j.neucom.2018.05.080.

[6] Richard Brath and Craig Hagerman. 2021. Automated Insights on Visualizations with Natural Language Generation. In *Proc. of IV*. IEEE, 278-284. https://doi.org/10.1109/IV53921.2021.00052.

[7] Matthew Brehmer and Tamara Munzner. 2013. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE TVCG*, 19, 12 (2013), 2376-2385. https://doi.org/10.1109/TVCG.2013.124

[8] Chris Bryan, Kwan-Liu Ma, and Jonathan Woodring. Temporal Summary Images: An Approach to Narrative Visualization via Interactive Annotation Generation and Placement. *IEEE TVCG*, 23, 1 (2017), 511-520. https://doi.org/10.1109/TVCG.2016.2598876

[9] George Chalhoub and Advait Sarkar. 2022. “It’s Freedom to Put Things Where My Mind Wants”: Understanding and Improving the User Experience of Structuring Data in Spreadsheets. In *Proc. of CHI*. ACM. https://doi.org/10.1145/3491102.3501833.

[10] Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, and Ryan Rossi. 2020. Figure Captioning with Relation Maps for Reasoning. In *Proc. of WACV*. IEEE.

[11] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. 2020. Say As You Wish: Fine-Grained Control of Image Caption Generation With Abstract Scene Graphs. In *Proc. of CVPR*. IEEE.

[12] Zhutian Chen and Haijun Xia. 2022. CrossData: Leveraging Text-Data Connections for Authoring Data Documents. In *Proc. of CHI*. ACM. https://doi.org/10.1145/3491102.3517485.

[13] Zhiyu Chen, Mohamed Trabelsi, Jeff Heflin, Yinan Xu, and Brian D. Davison. 2020. Table Search Using a Deep Contextualized Language Model. In *Proc. of SIGIR*. ACM. https://doi.org/10.1145/3397271.3401044.

[14] Kenneth Cox, Rebecca E. Grinter, Stacie L. Hibino, Lalita Jategaonkar Jagadeesan, and David Mantilla. 2021. A Multi-Modal Natural Language Interface to an Information Visualization Environment. *International Journal of Speech Technology*, 4 (2001), 297-314. https://doi.org/10.1023/A:1011368926479.

[15] Ali Davoudian, Liu Chen, and Mengchi Liu. 2018. A Survey on NoSQL Stores. *ACM Comput. Surv*. 51, 2, Article 40 (2019). https://doi.org/10.1145/3158661.

[16] Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. 2012. Summarizing Information Graphics Textually. *Computational Linguistics*, 38, 3 (2012), 527-574. https://doi.org/10.1162/COLI_a_00091

[17] Pierre Dragicevic, Yvonne Jansen, Abhraneel Sarma, Matthew Kay, and Fanny Chevalier. 2019. Increasing the Transparency of Research Papers with Explorable Multiverse Analyses. In *Proc. of CHI*. ACM. https://doi.org/10.1145/3290605. 3300295.

[18] Ethan Fast, Binbin Chen, Julia Mendelsohn, Jonathan Bassen, and Michael S. Bernstein. 2018. Iris: A Conversational Agent for Complex Tasks. In *Proc. of CHI*. ACM. https://doi.org/10.1145/3173574.3174047.

[19] Maddalena Favaretto, Eva De Clercq, and Bernice Simone Elger. 2019. Big Data and discrimination: perils, promises and solutions. A systematic review. *Journal of Big Data*, 6, 1 (2019). https://doi.org/10.1186/s40537-019-0177-4.

[20] Leo Ferres, Avi Parush, Shelley Roberts, and Gitte Lindgaard. 2006. Helping People with Visual Impairments Gain Access to Graphical Information Through Natural Language: The iGraph System. In *Proc. of ICCHP*. Springer. https://doi.org/10.1007/11788713_163.

[21] Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie Karahalios. 2015. DataTone: Managing Ambiguity in Natural Language Interfaces for Data Visualization. In *Proc. of UIST*. ACM, 489–500. https://doi.org/10.1145/2807442. 2807478.

[22] Pascal Goffin, Tanja Blascheck, Petra Isenberg, and Wesley Willett. 2020. Interaction Techniques for Visual Exploration Using Embedded Word-Scale Visualizations. In *Proc. of CHI*. ACM. https://doi.org/10.1145/3313831.3376842

[23] Google. 2022. Firebase. https://firebase.google.com/.

[24] (10, 20) Google. 2022. Google Sheets function list. https://support.google.com/docs/table/25273?hl=en.

[25] Pengcheng He, Yi Mao, Kaushik Chakrabarti, and Weizhu Chen. 2019. X-SQL: reinforce schema representation with context. *arXiv preprint arXiv:1908.08113* (2019). https://doi.org/10.48550/arXiv.1908.08113

[26] Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349* (2020). https://doi.org/10.48550/arXiv.2004.02349

[27] Amir Jahanlou and Parmit K Chilana. 2022. Katika: An End-to-End System for Authoring Amateur Explainer Motion Graphics Videos. In *Proc. of CHI*. ACM, 1–14. https://doi.org/10.1145/3491102.3517741.

[28] Matthias Jarke and Jurgen Koch. 1984. Query Optimization in Database Systems. *ACM Comput. Surv*. 16, 2 (1984), 111–152. https://doi.org/10.1145/356924.356928.

[29] Dae Hyun Kim, Enamul Hoque, Juho Kim, and Maneesh Agrawala. 2018. Facilitating Document Reading by Linking Text and Tables. In *Proc. of UIST*. ACM, 423–434. https://doi.org/10.1145/3242587.3242617.

[30] Tae Soo Kim, DaEun Choi, Yoonseo Choi, and Juho Kim. 2022. Stylette: Styling the Web with Natural Language. In *Proc. of CHI*. ACM, 1–17. https://doi.org/10.1145/3491102.3501931.

[31] Nicholas Kong, Marti A. Hearst, and Maneesh Agrawala. 2014. Extracting References Between Text and Charts via Crowdsourcing. In *Proc. of CHI*. ACM, 31–40. https://doi.org/10.1145/2556288.2557241.

[32] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision* 123, 1 (2017), 32-73. https://doi.org/10.1007/s11263-016-0981-7

[33] Alexandros Labrinidis and H. V. Jagadish. 2012. Challenges and opportunities with big data. In *Proc. of VLDB Endow*. ACM, 5, 12 (2012), 2032–2033. https://doi.org/10.14778/2367502.2367572.

[34] Chufan Lai, Zhixian Lin, Ruike Jiang, Yun Han, Can Liu, and Xiaoru Yuan. 2020. Automatic Annotation Synchronizing with Textual Description for Visualization. In *Proc. of CHI*. ACM, 1–13. https://doi.org/10.1145/3313831.3376443.

[35] Shahid Latif and Fabian Beck. 2019. VIS Author Profiles: Interactive Descriptions of Publication Records Combining Text and Visualization. *IEEE TVCG* 25, 1 (2019), 152-161. https://doi.org/10.1109/TVCG.2018.2865022

[36] Shahid Latif, Diao Liu, and Fabian Beck. 2018. Exploring Interactive Linking Between Text and Visualization. In *Proc. of EuroVis*. Eurographics Association. https://doi.org/10.2312/eurovisshort.20181084.

[37] Shahid Latif, Zheng Zhou, Yoon Kim, Fabian Beck, and Nam Wook Kim. 2021. Kori: Interactive Synthesis of Text and Charts in Data Documents. *IEEE TVCG* (2021). https://doi.org/10.1109/TVCG.2021.3114802.

[38] Mina Lee, Percy Liang, and Qian Yang. 2022. CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities. In *Proc. of CHI*. ACM, 1–19. https://doi.org/10.1145/3491102.3502030.

[39] Can Liu, Liwenhan Xie, Yun Han, Datong Wei, and Xiaoru Yuan. 2020. AutoCaption: An Approach to Generate Natural Language Description from Visualization Automatically. In *Proc. of PacifcVis*. IEEE, 191–195.

[40] Alan Lundgard and Arvind Satyanarayan. 2022. Accessible Visualization via Natural Language Descriptions: A Four-Level Model of Semantic Content. *IEEE TVCG* 28, 1 (2022), 1073-1083. https://doi.org/10.1109/TVCG.2021.3114770

[41] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proc. of ACL*. Association for Computational Linguistics. http://dx.doi.org/10.3115/v1/P14-5010

[42] Jim Melton and Alan R. Simon. 1993. Understanding the new SQL: a complete guide. *Morgan Kaufmann*, 1993.

[43] Meta Platforms. 2022. React. https://reactjs.org/.

[44] Arpit Narechania, Arjun Srinivasan, and John Stasko. 2021. NL4DV: A Toolkit for Generating Analytic Specifcations for Data Visualization from Natural Language Queries. *IEEE TVCG* 27, 2 (2021), 369–379. https://doi.org/10.1109/TVCG.2020. 3030378 arXiv:2008.10723.

[45] Jason Obeid and Enamul Hoque. 2020. Chart-to-Text: Generating Natural Language Descriptions for Charts by Adapting the Transformer Model. In *Proc. of INLG*. Association for Computational Linguistics, 2020. https://doi.org/10.48550/arXiv.2010.09142

[46] OpenJS Foundation. 2022. Node.js. https://nodejs.org/en/.

[47] Fatma Özcan, Abdul Quamar, Jaydeep Sen, Chuan Lei, and Vasilis Efthymiou. 2020. State of the Art and Open Challengesin Natural Language Interfacesto Data. In *Proc. of SIGMOD*. ACM, 2629–2636. https://doi.org/10.1145/3318464.3383128.

[48] Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proc. of ACL*. Association for Computational Linguistics, 1470-1480. http://dx.doi.org/10.3115/v1/P15-1142

[49] Arkil Patel, Satwik Bhattamishra, Navin Goyal. 2021. Are NLP Models really able to Solve Simple Math Word Problems?. In *Proc. of NAACL*. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/2021.naacl-main.168.

[50] Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proc. of IUI*. ACM, 149–157. https://doi.org/10.1145/604045.604070.

[51] Yuji Roh, Geon Heo, and Steven Euijong Whang. 2021. A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. *IEEE TKDE*, 33, 4 (2021), 1328-1347. https://doi.org/10.1109/TKDE.2019.2946162

[52] Diptikalyan Saha, Avrilia Floratou, Karthik Sankaranarayanan, Umar Farooq Minhas, Ashish R. Mittal, and Fatma Özcan. 2016. ATHENA: An OntologyDriven System for Natural Language Querying over Relational Data Stores Diptikalyan. In *Proc. of VLDB Endowment* 9, 12 (2016), 1209 - 1220. https://doi.org/10. 4324/9780203932148.

[53] Advait Sarkar, Judith W. Borghouts, Anusha Iyer, Sneha Khullar, Christian Canton, Felienne Hermans, Andrew D. Gordon, and Jack Williams. 2020. Spreadsheet Use and Programming Experience: An Exploratory Survey. In *Proc. of CHI EA*. ACM, 1–9. https://doi.org/10.1145/3334480.3382807.

[54] Jaydeep Sen, Chuan Lei, Abdul Quamar, Fatma Özcan, Vasilis Efthymiou, Ayushi Dalmia, Greg Stager, Ashish Mittal, Diptikalyan Saha, and Karthik Sankaranarayanan. 2020. ATHENA++: natural language querying for complex nested SQL queries. In *Proc. VLDB Endow*. 13, 12 (2020), 2747–2759. https://doi.org/10.14778/3407790.3407858.

[55] Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai, and Jianmin Wang. 2022. Towards Natural Language Interfaces for Data Visualization: A Survey. *IEEE TVCG* (2022), 1-1. https://doi.org/10.1109/TVCG.2022.3148007

[56] Marco Antonio Calijorne Soares and Fernando Silva Parreiras. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University - Computer and Information Sciences*, 32, 6 (2020), 635-646. https://doi.org/10.1016/j.jksuci.2018.08.005.

[57] Nicole Sultanum, Fanny Chevalier, Zoya Bylinskii, and Zhicheng Liu. 2021. Leveraging Text-Chart Links to Support Authoring of Data-Driven Articles with VizFlow. In *Proc. of CHI*. ACM, 1–17. https://doi.org/10.1145/3411764.3445354.

[58] Toni Taipalus and Ville Seppänen. 2020. SQL Education: A Systematic Mapping Study and Future Research Agenda. *ACM Trans. Comput. Educ*. 20, 3. https://doi.org/10.1145/3398377.

[59] Mohamed Trabelsi, Zhiyu Chen, Brian D. Davison, and Jeff Heflin. 2020. A Hybrid Deep Model for Learning to Rank Data Tables. In *Proc. of Big Data*. IEEE. https://doi.org/10.1109/BigData50022.2020.9378185

[60] Mohamed Trabelsi, Zhiyu Chen, Brian D. Davison, and Jeff Heflin. 2020. Relational Graph Embeddings for Table Retrieval. In *Proc. of Big Data*. IEEE. https://doi.org/10.1109/BigData50022.2020.9378239.

[61] Immanuel Trummer, Jiancheng Zhu, and Mark Bryan. 2017. Data vocalization: optimizing voice output of relational data. In *Proc. of VLDB Endow.* 10, 1574–1585. https://doi.org/10.14778/3137628.3137663.

[62] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018). https://doi.org/10.48550/arXiv.1804.07461

[63] Fei Wang, Kexuan Sun, Muhao Chen, Jay Pujara, and Pedro Szekely. 2021. Retrieving Complex Tables with Multi-Granular Graph Representation Learning. In *Proc. of SIGIR*. ACM. https://doi.org/10.1145/3404835.3462909

[64] Yuxi Wang, Martin McKee, Aleksandra Torbica, and David Stuckler. 2019. Systematic Literature Review on the Spread of Health-related Misinformation on Social Media. *Social Science & Medicine*, 240 (2019). https://doi.org/10.1016/j.socscimed.2019.112552.

[65] Yun Wang, Zhida Sun, Haidong Zhang, Weiwei Cui, Ke Xu, Xiaojuan Ma, and Dongmei Zhang. 2020. DataShot: Automatic Generation of Fact Sheets from Tabular Data. *IEEE TVCG* 26, 1 (2020), 895-905. https://doi.org/10.1109/TVCG.2019.2934398.

[66] Stephen Wehrend and Clayton Lewis. 1990. A problem-oriented classification of visualization techniques. In *Proc. of Visualization*. IEEE. https://doi.org/10.1109/VISUAL.1990.146375.

[67] Aoyu Wu, Wai Tong, Haotian Li, Dominik Moritz, Yong Wang, and Huamin Qu. 2022. ComputableViz: Mathematical Operators as a Formalism for Visualisation Processing and Analysis. In *Proc. of CHI*. ACM, 1–15. https://doi.org/10.1145/3491102.3517618.

[68] Aoyu Wu, Yun Wang, Xinhuan Shu, Dominik Moritz, Weiwei Cui, Haidong Zhang, Dongmei Zhang, and Huamin Qu. 2021. AI4VIS: Survey on Artificial Intelligence Approaches for Data Visualization. *IEEE TVCG* (2021), 1-1. https://doi.org/10.1109/TVCG.2021.3099002.

[69] Haijun Xia. 2020. Crosspower: Bridging Graphics and Linguistics. In *Proc. of UIST*. ACM, 722–734. https://doi.org/10.1145/3379337.3415845.

[70] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: neural image caption generation with visual attention. In *Proc. of ICML*. PMLR. https://proceedings.mlr.press/v37/xuc15.html.

[71] Yang Chen, Jing Yang, and William Ribarsky. 2009. Toward effective insight management in visual analytics systems. In *Proc. of PacificVis*. IEEE. https://doi.org/10.1109/PACIFICVIS.2009.4906837.

[72] Carmen Yip, Jie Mi Chong, Sin Yee Kwek, Yong Wang, and Kotaro Hara. 2021. Visionary Caption: Improving the Accessibility of Presentation Slides Through Highlighting Visualization. In *Proc. of ASSETS*. ACM, 1–4. https://doi.org/10.1145/3441852.3476539.

[73] Bowen Yu and Cláudio T. Silva. 2020. FlowSense: A Natural Language Interface for Visual Data Exploration within a Dataflow System. *IEEE TVCG* 26, 1 (2020), 1-11. https://doi.org/10.1109/TVCG.2019.2934668.

[74] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2SQL: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103* (2017). https://doi.org/10.48550/arXiv.1709.00103.

[75] Hong Zhou. 2020. Learn Data Mining Through Excel. *Springer*.

[76] Michelle X. Zhou and Steven K. Feiner. 1998. Automated Visual Presentation: From Heterogeneous Information to Coherent Visual Discourse. *Journal of Intelligent Information Systems*, 11, 3 (1998), 205-234. https://doi.org/10.1023/A:1008685907948.