Visual Analytics for
Domain-Specific Knowledge Exploration and Exploitation

By

Xiaoyu Zhang
Dissertation

Submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Office of Graduate Studies

of the

University of California

Davis

Approved:

Kwan-Liu Ma, Chair

Barbara S. Linke

Panpan Xu

Committee in Charge

2023

Ignorance is the curse of God, knowledge the wing wherewith we fly to heaven.

— William Shakespeare

Contents

Abstract

Visual Analytics for
Domain-Specific Knowledge Exploration and Exploitation

In the past decade, the proliferation of data and the emergence of large language models have presented both opportunities and challenges in academia. The expanding volume of data, which records knowledge from various human activities, enables data-driven approaches to optimizing numerous aspects of industrial manufacturing and people's daily life. These improvements largely stem from machine learning models trained with this data. However, the industry still faces limitations in both extracting knowledge from large, unstructured, or heterogeneous datasets and transforming the extracted knowledge into actionable insights. This challenge is exacerbated in highly specialized domains where only a few analysts possess the expertise to interpret the data. Despite the recent advancements of large language models providing more intelligent assistance for many data analysis tasks, it remains essential to ensure that these machine learning models and the knowledge they encompass are safe to use and employed for social good with human verification.

In my dissertation work, I develop visual analytics (VA) and human-computer interaction (HCI) methodologies for representing and interacting with various forms of knowledge and data, particularly text data. I propose a visual knowledge discovery framework that integrates human expertise with computational approaches throughout the knowledge discovery process, while also addressing the limited availability of domain experts and the increasing scale of data. Moreover, I investigate how visual analytics can efficiently and safely harness extensive knowledge from large machine learning models, enabling users to effectively steer the exploration process and make well-informed decisions.

This dissertation presents six published research works organized around my visual knowledge discovery framework and its three key tasks: knowledge exploration, knowledge presentation, and knowledge exploitation. Firstly, I demonstrate how vi-

sual analytics can support knowledge exploration with large, high-dimensional, and heterogeneous data in the domain of manufacturing and machine maintenance. Subsequently, I introduce two knowledge presentation solutions for two distinct types of data—numerical data facts and unstructured text data. Lastly, I showcase three visually-assisted knowledge exploitation applications in various domains and scenarios, encompassing document summarization, technical text annotation, and data-driven machine learning model validation.

My work demonstrates how mixed-initiative methods through visual analytics applications can resolve real-world challenges in highly-specialized domains. I leverage state-of-the-art machine learning techniques, particularly natural language processing models, while always involving domain practitioners in the loop. My approach facilitates communication among parties with mismatched knowledge levels, including domain experts, data analysts, computer scientists, and artificial intelligence. Meanwhile, I prioritize the critical role of human knowledge and integrate it into intelligent visualization interfaces that undergo qualitative evaluations. I believe that domain experts' insights, supervision, and verification are invaluable, regardless of how advanced machine learning techniques become. Through the projects outlined in this dissertation, I hope to encourage philosophical and social discussions surrounding the rapidly expanding field of artificial intelligence. Ultimately, my objective is to contribute to a future where intelligent visual analytics systems can augment and enhance human capabilities, enabling individuals to navigate through the potential challenges brought by advanced AI techniques.

Acknowledgments

I can still vividly recall the first morning after I arrived in the U.S. to pursue my Ph.D. studies. When I woke up on a bare mattress laid on the apartment floor, I basked in the spectacular California sunshine, my heart brimming with a mixture of excitement and anxiety towards the uncertain future. Five years has passed, during which I experienced successes and loss, joy and sorrow, and persevered through the unprecedented dark times of the pandemic. As I sit here writing the final few lines of my dissertation, my heart is filled with gratitude and cheerfulness for all that I have encountered and will experience in my life.

I would like to thank my Ph.D. advisor, Professor Kwan-Liu Ma, for his precious support and encouragement to pursue my goals. Thanks to his guidance, I am able to remain in academia and continue to dedicate myself to education and research. I am grateful to my qualifying examination and dissertation committee members, Professor Barbara S. Linke, Dr. Panpan Xu, Professor Hao-Chuan Wang, and Professor Zhou Yu for their help to improve my dissertation. Special thanks to Professor Senthil Chandrasegaran, Professor Bin Zheng, and Dr. Liu Ren for their constant encouragement and guidance on my life, study and job applications.

I extend my heartfelt gratitude to my dear friends, Yalin, Huiwen, Fei, Xiaoyang, Noriko and Art, for their unconditional support and love, which helped me overcome countless challenges. The joyful moments we shared together will forever shine in my heart. I am grateful to all of my collaborators, especially Kelvin, Jorge, Liang, Wenbin, Huan, Takanori, Alden, Thurston, and Mike. I will always cherish the memories of working together under tight deadlines and the research skills I learned from each of you. I also want to thank myself for never giving up or compromising on my dream.

Last but not least, I express my gratitude to all of my family members for their unwavering love. I also want to dedicate this dissertation to my beloved grandfather, who is no longer with us – Grandpa, I made it! I hope you can see it from heaven.

Chapter 1
Introduction

1.1 Motivation

Over the past decade, there has been a significant increase in interest and applications of artificial intelligence across various research fields. Large language models such as BERT [85], CLIP [264], and GPT-4 [266] have emerged and demonstrated their ability to solve complex, cross-modal tasks and generate exceptional results. More recently, the booming of online artificial intelligence systems like ChatGPT [267] and DALL-E [265] has made these large machine-learning models more accessible to a broader range of users and has attracted significant societal attention. However, it is crucial to be aware that the outstanding capability of these large machine learning models is built upon the proliferation of training data and the knowledge embedded within it, which is a double-edged sword for all users. On the one hand, such models could serve as more comprehensive knowledge sources and powerful intelligent assistants to support various tasks, such as question answering, recommendation, and generation. On the other hand, the quality and reliability of the data can significantly influence the performance and output of the machine learning models, making it unwise to treat them as complete black boxes and rely solely on them without any human verification. Therefore, it is essential to ensure that these machine learning models and the knowledge they contain are safe to use and used for social good, truly benefiting people's daily lives, commercial interactions, and industrial manufacturing.

It is important to note the significant challenge presented by the management and integration of diverse forms of knowledge from humans, data, and machine learning models. Knowledge entwined with data can appear in various formats, with its volume and complexity continually escalating. This intricacy is further intensified in highly

specialized domains where data is interpretable only by experts possessing substantial domain knowledge. Such data interpretation process is often time-consuming and resource-intensive, often requiring experts to review thousands of entries. While the recent emergence of large machine learning models offers a new way of storing and utilizing knowledge, human involvement remains indispensable for directing model behavior and making crucial decisions.

In my dissertation research, I employ visual analytics—"the science of analytical reasoning facilitated by interactive visual interfaces" [76]—to address these challenges. My work assists domain experts and practitioners in comprehending and engaging with data and machine learning models, and ensure that they can steer the knowledge exploration and exploitation process according to their expertise and at their discretion. In particular, I accomplish the following three research objectives in collaboration with my colleagues and published multiple research papers:

1. Develop user-friendly visual analytics tools to aid domain experts in examining knowledge within large and complex datasets;
2. Determine suitable visualizations for presenting existing knowledge originating from diverse sources and various forms;
3. Establish robust workflows for capitalizing on the amalgamated knowledge in a secure, reliable, and personalized manner.

In the remainder of this chapter, a background description is furnished in Section 1.2 to establish the theoretical foundation of this dissertation. Subsequently, a overview of the content is presented in Section 1.3, encompassing my contributions to each task and the corresponding publications.

1.2 Background

To facilitate a comprehensive understanding of the work presented in this dissertation, this section provides a concise review of the relevant background. First, I define domain-specific knowledge, outline its two categories, and discuss the associated challenges. Next, I examine existing knowledge discovery models and propose a visual

knowledge discovery framework based on these models. Lastly, I delineate the role of visual analytics in three core tasks of this framework: knowledge exploration, knowledge exploitation, and knowledge presentation. The content of this dissertation is organized according to these three tasks, ensuring a coherent and logical presentation of the research findings.

1.2.1 Domain-Specific Knowledge

Domain-specific knowledge is a concept with various definitions and transdisciplinary developments in the literature of educational research, psychology, linguistics, and philosophy of science. One of the original and higher-level definitions of domain knowledge comes from the field of educational research, which identifies three key elements of domain knowledge: *declarative knowledge (knowing that)*, *procedural knowledge (knowing how)*, and *conditional knowledge (knowing when and where)* [7]. This dissertation is informed by this definition, but contextualized to the perspective of library and information science (LIS), where the main focus is "*highly selective and relevant knowledge that is to provide users with as complete a view as possible of theories, topics, and approaches to a given subject and make it possible for them to be informed and to select according to their needs*" [156], rather than "*automated commonplace*" knowledge. Such knowledge typically has to be learned from domain experts or specialists and can vary in forms.

As Nonaka et al. describe in the "Knowledge-Creating Company" [257], knowledge can be classified into two categories: *tacit knowledge* and *articulated knowledge*, which lie at opposite poles of the epistemological dimension. In the context of domain-specific knowledge, tacit knowledge encompasses *unformatted* insights about the domain dataset, experience regarding the priority of multiple domain tasks, and actionable decisions from domain experts. At the other end of the pole, tacit knowledge can be *formatted* into articulated knowledge, which includes unstructured documents, structured tables, knowledge graphs (such as ontology), and more. In recent years, the advent of artificial intelligence has also enabled the integration of domain-specific knowledge into large machine learning models, particularly large language models like BERT [85] and GPT-4 [266], which can be leveraged for specific prediction tasks.

The exploration and exploitation of domain-specific knowledge can be a difficult task. First, capturing and formalizing tacit knowledge is challenging due to communication barriers between domain experts and knowledge engineers, so they must establish a common language and develop a shared vocabulary to communicate effectively. Even if tacit knowledge is formatted into articulated knowledge, it may still be incomplete, inconsistent, and difficult to share. For example, articulated knowledge formatted as unstructured text can be terse and jargon-laden and may vary in consistency across data entry personnel. In such cases, developing a lexicon, or domain-specific vocabulary, is often necessary to interpret text data in a semantically consistent way. The General Inquirer [339] is one of the earliest attempts to build a lexicon for the content analysis of the text. More recently, researchers have developed word embeddings—representations of words and their semantic relationships in a vector space—to "characterize words by the company they keep" [109]. However, it is critical to verify the reliability of knowledge in such word embeddings or large machine learning models before applying them to real-world applications. For instance, the ChatGPT chatbot [267], powered by GPT models [266], has been known to provide plausible answers with inaccurate "facts". In this dissertation, I introduce multiple visual analytics techniques to provide human-in-the-loop solutions to these issues, particularly human-centered AI [323] and explainable AI [15, 353], to leverage knowledge from large machine learning models in a trustworthy manner.

1.2.2 Domain-Specific Knowledge Discovery Models

Knowledge discovery, as a nontrivial information extraction process, aims to identify implicit, previously unknown, and potentially useful patterns from the dataset [113]. Knowledge discovery in database (KDD) is a broad and well-established research area that has been studied by researchers from academia and industry over the past thirty years. It is also referred to as data mining by researchers in this area, for their shared interest in machine discovery, machine learning, data visualization, knowledge acquisition, knowledge-based systems [110], etc. The scope of this dissertation overlaps with KDD in the field of domain-specific knowledge exploration and exploitation with visual

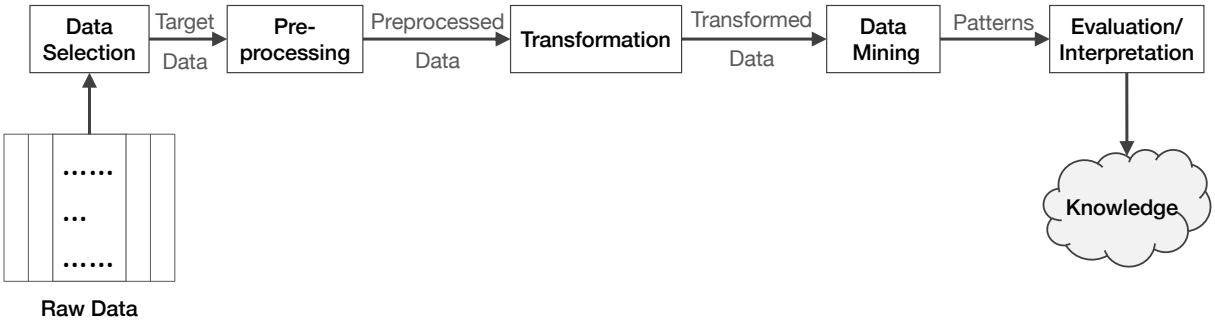


Figure 1.1: The linear pipeline of knowledge discovery in database proposed by Mitra et al. [244]

analytics technology. Respectively, knowledge exploration is related to the searching, sensemaking, and innovation of new possibilities, while knowledge exploitation is related to the selection, refinement, and presentation of old certainties [229]. They are also two typical roles knowledge could play in a linear KDD pipeline—as input or output. However, I consider it more reasonable to model the entire KDD process as a loop where both knowledge exploration and exploitation happen. I will review dominant knowledge discovery models and introduce my loop model in the following text.

To streamline the knowledge discovery process, it is critical to understand the key analysis steps involved and identify areas where visual analytics (VA) tools can provide assistance. Knowledge discovery models or workflows may differ in their structure, key steps, and application domain, depending on the specific requirements of the analysis tasks. To emphasize the advantages of loop models over linear models, I categorize existing models into three groups based on the presence and location of iterations within the workflow:

Naive Linear Model. Linear models without any iterations are quite rare and mostly appeared in the early stage of this research field. Date back to 1993, Matheus et al. proposed their "idealized knowledge-discovery system" as a linear pipeline with user input and knowledge base paralleled with the main workflow [231]. Mobasher et al. also provided a linear workflow with several parallel paths, but the input was slightly different because the model was specially designed for pattern discovery in world wide web transactions [245]. In Mitra et al.'s later survey about data mining in soft computing tools [244], they separate the KDD process into seven stages and

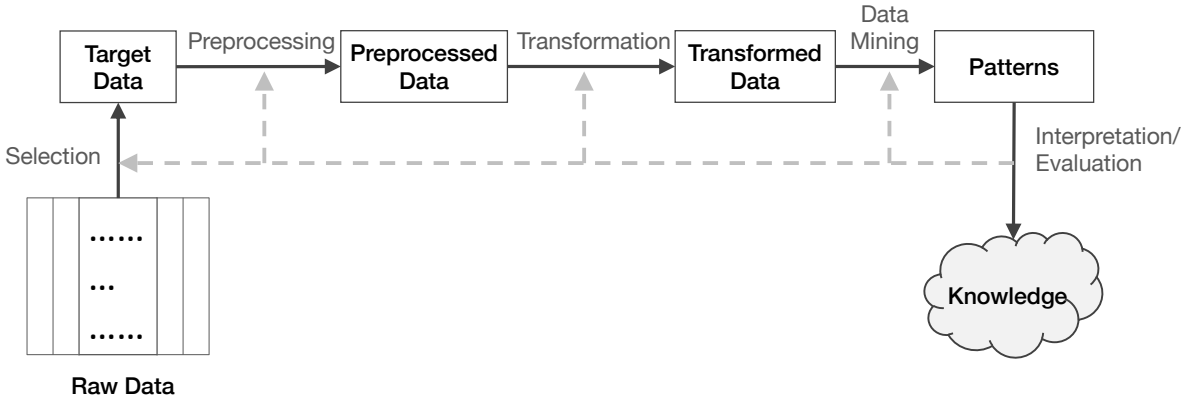


Figure 1.2: The famous KDD process proposed by Fayyad et al. [102–104, 108]

organize them linearly (Figure 1.1). The advantage of such models exists in their simplicity and clearness. However, the lack of feedback from the later steps limits the amount and quality of knowledge they could discover.

Linear Model with Inner Feedback. This type of model is featured by small inner loops among several steps inside the pipeline, but the output knowledge is not directly fed back to the input one. One representative model of this type is the well known KDD process proposed by Fayyad et al. (Figure 1.2) [102–104, 108]. In this model, there is an interaction between any two steps in the pipeline, but the overall process is still linear instead of a complete loop. Similar models can be seen in [83, 114] with minor adjustment of step names and where the inner loops take place. In particular, [114] emphasizes the role of visualization in the process of knowledge discovery, and two transformation operations are added before and after the visualization stage to accommodate the data. Pragmatic as these models are, knowledge is considered as static patterns inherit from the input data instead of the sustainable resource that can be iteratively used to inspire new observations.

Circular Model. In 1992, Frawley et al. framed the process of knowledge discovery in databases as an iterative loop [113], where the domain-knowledge acts both as an output and as an input of the discovery system (Figure 1.3). This framework provides the foundation for my visual knowledge discovery model. Besides, circular knowledge discovery models for the general purpose also appear in [21, 158, 280] as well as the

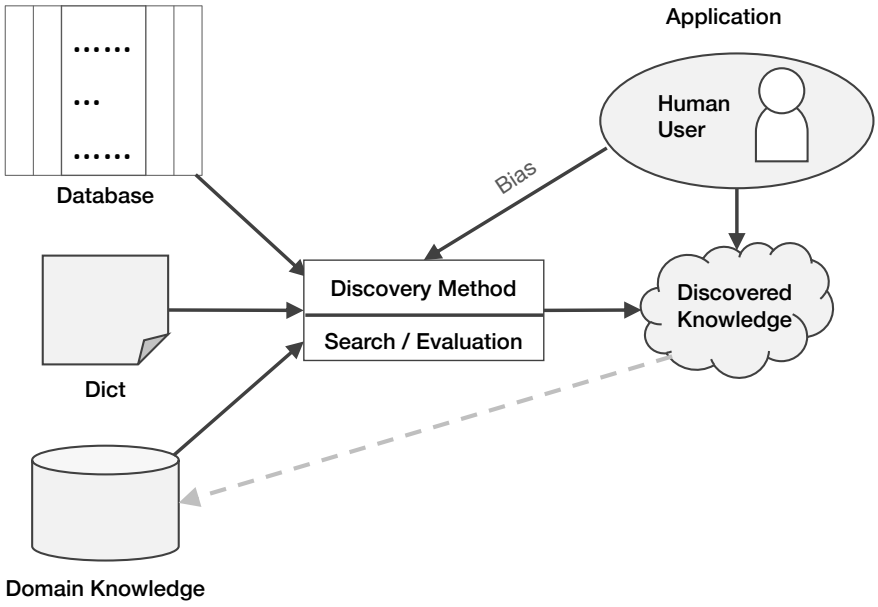


Figure 1.3: The loop of knowledge discovery in database proposed by Frawley et al [113].

series work by Silberschatz and Tuzhilin [324–326]. In particular, [21] provides a unique framework about knowledge creation derived from [257], where articulated knowledge and tacit knowledge are contrasted as the two poles of the epistemological dimension. Circular models are also widely utilized for domain-specific knowledge discovery. For instance, the model introduced by Shaw et al. [316] is for knowledge management and data mining for marketing. Wagner et al. described a knowledge generation loop for clinical gait analysis [366], in which the role of clinician and patients is specified. In the field of maintenance and manufacturing, the application domain ofchapter 2, there are also quite a few circular models being proposed [61, 65].

1.2.3 Visual Knowledge Discovery Framework

As mentioned above, I base my visual analytics framework for knowledge exploration and exploitation (see Figure 1.4) on the knowledge discovery loop proposed by Frawley et al. [113]. Unlike Frawley's framework, which centers around a discovery method comprising search and evaluation algorithms, my framework places an interactive visual interface at its core to highlight the crucial role of interactive discovery. For the same purpose, I consolidate the "selection", "preprocessing", and "transformation"

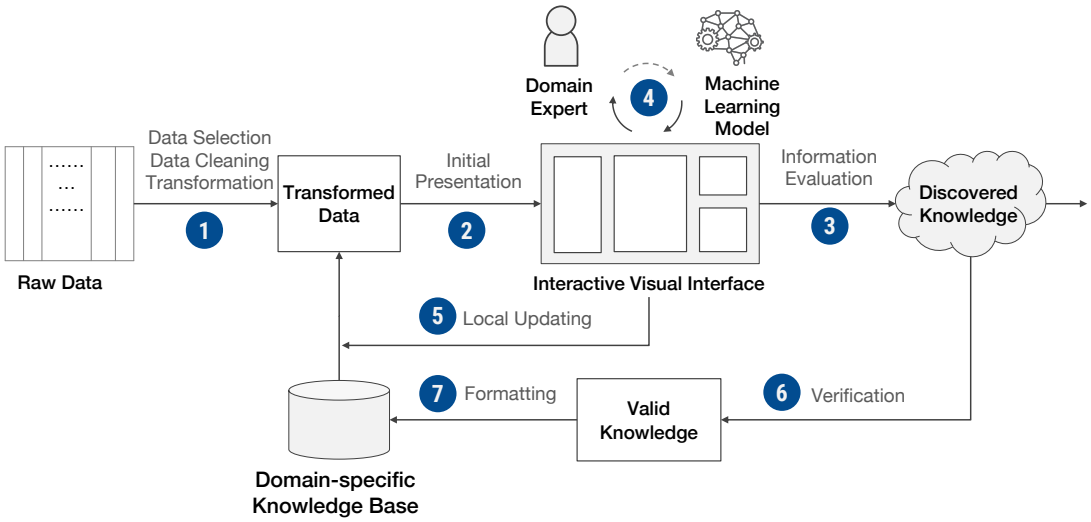


Figure 1.4: The visual knowledge discovery framework proposed in this dissertation.

steps in the classic Fayyad model [103] into step 1 in my framework. After incorporating domain-specific knowledge, the "transformed data" would contain sufficient information for the interactive visual interface (step 2). Step 4 represents the continuous interaction among the technician/analyst, the visualization dashboard, and the underlying machine learning model to harness existing articulated knowledge and extract new tacit knowledge. I refer to this process as "knowledge exploration" and discuss further details in section 1.2.3.1. Both steps 2 and 4 necessitate careful design for "knowledge presentation", which I address in section 1.2.3.2. Finally, the extracted knowledge that passes evaluation (step 3) constitutes the initial result of the first iteration. To incorporate this feedback into subsequent iterations, I either directly update the transformed data (step 5) or update the domain-specific knowledge database or machine learning model (steps 6 and 7). I name this process "knowledge exploitation" and will discuss it in section 1.2.3.3.

1.2.3.1 Knowledge Exploration

To define knowledge exploration, I would first refer to John Tukey's famous description of exploratory data analysis (EDA)—"*looking at data to see what it seems to say*"—from his 1977 book [356]. Knowledge exploration is a special case of EDA where the goal is to discover "*implicit, previously unknown and potentially useful patterns*" [113].

The preference of users towards visualizations and interactions in the exploration process can vary greatly depending on specific tasks and challenges. Kandel et al. summarize five high-level data analysis tasks, i.e. discover, wrangle, profile, model, and report, in their interview study with 35 enterprise analysts [172]. Alspaugh et al. identify multiple challenges in the exploratory analysis process corresponding to these tasks based on their interview study with 30 professional data analysts [9]. Moreover, with the constant increase of data size and complexity, the more heterogeneous and multiple-dimensional dataset has imposed new challenges to the exploration process [11, 38, 385]. In response to these tasks and challenges, Blascheck et al. observe seven exploration strategies including eyes only, reading text, opportunistic interactions, entry points, structural interactions, permutation interactions, and leveraging the familiar after tracking and analyzing the eye movement and interaction data of 24 participants in a controlled environment [27].

To accommodate the various exploration tasks and strategies, I utilize a similar principle as that used in [172] to categorize existing research and tools. In the original paper, the authors identified three archetypes: the hacker, the scripter, and the application user, based on their programming proficiency and preferred tools. Within the context of this dissertation, I categorize these archetypes as three exploration patterns, each featuring a different archetype based on the primary type of knowledge possessed by the user, namely programming, modeling, and domain knowledge, respectively.

Users with more programming knowledge (*hackers*) are capable of manipulating the source data and tend to customize the analysis workflow by themselves. Visual analytics tools serving this group of users need to allow more interaction and operation flexibility, or even exposes the plug-in interface to them. For instance, the event sequence exploration interface developed by Law et al. supports a recursive manner for users to interweave self-defined queries and interactive pattern mining during the exploration process [194]. The knowledge retrieval system developed by Stitz et al. also allows users to define an analysis state by explicitly formulating a search query [338]. In recent years, there has been a trend of presenting visual analytics techniques as interac-

tive notebooks [119,203,403] to accommodate this group of users' inclination to design their workflow using lower-level programming languages [172]. In Chapter 6, I present my solution for technical text annotation verification and customization as a computational notebook. Additionally, hackers are more accepting of complex interactions and corresponding algorithms, such as edge bundling for biclustering algorithms [341] or minimum description length principle [51], which allows for the integration of more robust tools irrespective of their steep learning curve. It is worth noting that this group of users tends to develop less sophisticated statistical models than scripters due to their focus on prior-modeling steps and limited knowledge of modeling [172].

Indeed, the knowledge of mathematical modeling grants the second group of users (*scripters*) the ability to build and evaluate more advanced models. The corresponding visualization tools need to provide more complete and intuitive ways for users to explore the models as well as more reliable measures to test them. Plenty of research work has made such effort to facilitate easier correlation detection [401], model comparison [75, 89], parameter tuning [50, 300] and result comparison [92]. To win the trust of users in this group towards the analysis result is also challenging, due to their rigorous skepticism along with excessive knowledge about the pros and cons of those models. For instance, Xu et al. visualize the complete ensemble analysis process and provide an intuitive comparison to support the evaluation of the anomaly detection algorithms [388]. Blumenschein et al. provide automatically calculated statistical measures to increase the users' trust in the patterns revealed by the system [28].

Finally, there are *domain experts* who, though might not understand too much about the underlying statistical or implementation mechanism of the tool, possess abundant knowledge of the specific application domain and the data (*application user*). The knowledge required to interpret the data is sometimes out of the scope of the tool developers, so collaboration between developers and the domain experts is critical. One typical example is the multiscale visual drill-down tool developed by Furmanova et al. to assist proteomic experts in exploring multi-body protein complexes [120]. To fulfill the need of this user group, the information provided by the tool need to

be concise but accurate. One neat solution is to build the visualization upon the primary structure or relationship embedded in the dataset, which allows users to quickly align the views they see with their knowledge about the dataset. For instance, a tree-like graph can be used to resemble hierarchical genealogy [256] while node-link graph and density map can simulate social connections in social network datasett [213, 384]. There are even automatic tools to completely free domain experts from the tedious learning curve of new tools, such as Gramazio's automated classifiers for cancer genomics visualization [132]. There is also a trustiness issue for this group of users, but it is more related to their level of understanding of the underlying mechanism of the tool. To make sure the domain experts trust the tool and the result provided by it, the cybersecurity analysis tool developed by Goodall et al. not only facilitates them identifying anomalous IPs, but also provides additional context information explaining why they are anomalous [130].

1.2.3.2 Knowledge Presentation

My dissertation work centers on knowledge representations that leverage visual interfaces to present knowledge, which I refer to as knowledge presentation. This aspect is a essential component in almost all visual analytics systems intended for knowledge discovery. The choice of knowledge presentation techniques varies based on the application scenario and target audience. For instance, according to Chandrasegaran et al.'s review on knowledge representation in product design systems [55], which examines the classification and implementation of knowledge representation in product design, popularization could be used for educational purposes, communication for collaborative purposes, and sensemaking for exploration purposes.

One of the primary tasks of knowledge presentation is to support knowledge exploration, sensemaking, and analysis. This requires carefully selecting and presenting specific information at the right moment and in the right place to meet the different requirements of the exploration stages. Following the famous mantra for visual information seeking, "Overview first, zoom and filter, then details-on-demand" [322], the first critical information to be presented is an overview of the dataset. Ideally, intuitive

and concise representations such as hierarchical layout [126,319], geographical distribution [368], or clustering information [206,386] can reveal the primary structure of the data during this stage. As the user interacts with the system via various operations, more details need to be presented without overwhelming the users. Numerous previous works have adopted Lindstrom's [211] level of detail (LOD) principle and show different information granularities based on the user's visual interaction. However, some researchers believe that "you can't always sketch what you want" [197] and prefer to provide more accurate query approaches for users [338]. As a guideline, Sarikaya et al. provide a systematic review and research directions for the usage of dashboards in knowledge exploration and presentation [306]. Besides, understanding the differences among multiple data of interest is crucial for knowledge exploration, where parallel views are widely adopted for conceptual comparison [81,127], and timelines for chronicle comparison [63,217]. Finally, it would be beneficial to provide incremental updating of the visualization to reflect users' thoughts or discoveries during the exploration process, as described in section 1.2.3.3. Sarvghad et al. utilize scented widgets to reveal users' analysis history and help them form new questions by reviewing the different perspectives of their past work [308].

In VA systems where at least two users are collaborating, it is crucial to facilitate their communication and bridge the knowledge gap with appropriate presentation technology. This process is challenging in both synchronous and asynchronous collaborations when the "curse of knowledge" hinders visual data communication [387]. When the collaborators possess significantly unequal domain knowledge, cognitive biases can hinder smooth communication. To address this, Xu et al. designed a chart constellation graph accessible to all collaborators to reflect their findings during the analysis in synchronous scenarios [390]. Similarly, Zhao et al. introduced the knowledge-transfer graph (KTGraph) to automatically capture tacit knowledge and hand off partial findings in asynchronous scenarios [402]. For a comprehensive reference, Chandrasegaran et al. reviewed a vast number of state-of-the-art solutions in their survey [55], among which ontologies are a preferred form to eliminate bias. In line with this direction, I introduce

a technique to use an ontology as the knowledge base and visualize domain-specific knowledge in documents in Chapter 4.

Finally, when it comes to the case that knowledge needs to be presented to multiple audiences, the major challenge is to maintain the trade-off between conveying the excessive obscure information and accessing the general public efficiently, such as exhibiting large cultural heritage collections in digital galleries [380], disseminate complex astronomical phenomena in popular media [29] or communicating personalized health risk with patients [142]. Cui and Wang et al. approach this problem by completing a series of work to convert the data facts into infographics, slideshow, and fact sheets [60,79,370,373]. Roberts et al. provide a pipeline called explanatory visualization framework (EVF) to guide learners in their explanatory visualization designing process for complex computational algorithms [298]. In response to the popularity of deep neural networks, Wang et al. [371] and Kahng [171] et al. both make their attempt to assist learners to understand the DNN architectures with multiple visualizations.

Apart from conventional solutions, new interaction modals and forms have been introduced for knowledge presentation and interaction in recent years. To extend the limited interaction provided by mouse and keyboard, Ma et al. involve the museum visitors in their plankton populations visualization with a touch table and control the average data interpretation time under one minute [226]. With a similar device, Agarwal et al. manage to accommodate both single or co-located users with different levels of visualization expertise [5]. Researchers make even further attempts to present the knowledge with wheeled micro-robots [195]. Another trendy branch is immersive analytics with devices such as immersive headsets or CAVE. As a reference, Marai et al. summarize their experience and perspective about immersive analytics based on their 25-year-long practice with multiple immersive devices and analytical tasks [228]. Beyond the visual channel, researchers have also been exploring approaches to leverage the touch, gesture, hearing [159], and smelly [277] channels for novel and tangible knowledge presentation. While many of the works are still in the fragile beta testing phase, they demonstrate promising interaction possibilities that have the potential to

significantly enhance the capability and accessibility of knowledge presentation. For further reference,, Lee et al. offers a detailed forecast of the future direction of Post-WIMP interaction techniques [196].

1.2.3.3 Knowledge Exploitation

According to March's definition [229], knowledge exploitation emphasizes the utilization and extension of existing possibilities compared to knowledge exploration. In other words, it retrieves knowledge that has already been created or identified [225] and incrementally accumulates it with "*moderate but certain and immediate returns*" [218]. Inspired by this definition, I distinguish the knowledge exploitation process into two types: immediate local updating and global knowledge base updating. These are also two different sources of exploited knowledge. Immediate local updating utilizes knowledge generated during the VA system's operation, while global knowledge base updating is based on existing knowledge collected and formalized before the operation [105]. This separation is also reflected in my knowledge discovery workflow, as described in Section 1.2.2 and depicted in Figure 1.4.

In Figure 1.4, the path from visualization dashboard via step 5 to transferred data is the instant way of knowledge exploitation, where the locally optimized data, models, or parameters drew from the user interaction will be directly fed into the transformed data and thenceforth the visualization. It proves to be very helpful to steer the underlying data model of a visual analytic system in this way. For instance, Kwon et al. have built multiple visual analytic dashboards driven by models derived from domain knowledge of expert users. Such knowledge is extracted from multiple types of user inputs such as sketching lines [190] or temporal interactive information [189]. Ming et al. leverage the information from sentiment analysis and predictive diagnostics to steer their deep sequence model and update the visualization in real time [241]. Another common practice is to draw semantic and syntactic knowledge from the user-input text by utilizing several natural language processing techniques such as topic modeling and text summarization. Sperrle et al. build a system to learn linguistic knowledge from user interactions and use it to provide automatic annotation suggestions of argumentative

text [333]. Lu et al. provide an automatically generated outline as well as multiple writing suggestions to support the prewriting process [222]. The immediate local updating is also widely adopted for the analysis of sports data, given the empirical and uncertain nature of sports activities. For example, Chung et al. create a knowledge-assisted ranking framework to extract the tacit knowledge about sports event ranking from users and produce visualizations upon the analytical model of this knowledge. [66]. Wang et al. take use of the tactics used by table tennis players to simulate competitions and facilitate strategy exploration [369]. Sometimes even the reasoning process itself is the treasure. To preserve the reasoning steps in the sensemaking or collaboration process, Camisetty et al. provide a JavaScript library to enhance existing web-based applications and support the knowledge capturing and replaying [46].

In another path, tacit knowledge can be carefully evaluated, verified, and formatted before being incorporated into the next round of iteration (step 3→step 6→step 7 in Figure 1.4). The byproduct of this route is an updated knowledge base that can be stored and shared for future usage. As a typical example of such structured knowledge representation models [123], ontology is widely used in the field of medicine/biology [125–127], engineering [299, 381], sociology [154], computer science [271, 340], etc. According to various application fields and utilizing purpose, there are multiple methods to exploit the knowledge stored in an ontology and turn it into visualization. The review of Katifori and Akrivi [173] systematically categorized these methods according to the dimensions of the visualization. Dudáš et al. [93] further extended this work by adding more recently emerged visualizations. Apart from ontology, Stitz et al. provide a method to create formatted knowledge by retrieving analysis states from user interactions and structuring them as provenance graphs [338]. Similarly, the clinical gait analytics solution provided by Wagner et al. allows patients to externalize and store their implicit knowledge and share it for later inspection [366].

It is also worth noting that visualization holds a distinct role in exploiting tacit knowledge from domain experts in the recent trend of human-centered AI [323]. Such knowledge remains critical for many machine learning model optimization tasks, par-

ticularly when domain knowledge is necessary for interpreting the training data. Researchers have utilized visual interfaces to present machine learning model performance intuitively and efficiently gather tacit expert knowledge for model optimization [82,241,242]. In my collaborative research project with Bosch Research (see Chapter 7), I employ this methodology for slice-based machine learning model optimization and intend to apply it in more of my ongoing and future research.

1.3 Content Overview

This dissertation is organized into different chapters according to the visual knowledge discovery framework (Figure 1.4) and its three key tasks described in Section 1.2.3.

Chapter 2 takes the domain of manufacturing (machine maintenance) as an example to demonstrate a visual analytics system for task 1, knowledge exploration. This chapter introduces a visual analytics approach that uses multiple dimensionality reduction and clustering algorithms to visualize and group different components of the data. My approach assists analysts to make sense of machines' maintenance logs and their errors, and their gained insights help them carry out preventive maintenance. I illustrate and evaluate my approach through use cases and expert studies respectively, and discuss generalization of the approach to other heterogeneous data. A version [348] of the research in this chapter was presented at IEEE Pacific Visualization Symposium 2021.

The following two chapters addresses task 2, knowledge presentation, for two types of data. Chapter 3 explores an approach to present *numerical data facts* by automatically generating infographics from natural language statements. I first describe a preliminary study conducted to explore the design space of infographics. Based on the preliminary study, I show how to build a proof-of-concept system that automatically converts statements about simple proportion-related statistics to a set of infographics with pre-designed styles. Finally, I demonstrate the usability and usefulness of the system through sample results, exhibits, and expert reviews. A version [79] of the research in this chapter was published in IEEE Transactions on Visualization and Computer Graphics and was presented at IEEE Visualization Conference 2019.

Chapter 4 presents ConceptScope, a technique that represents the *conceptual relationships in document data* by referring to a domain ontology and leveraging a Bubble Treemap visualization. ConceptScope facilitates exploration and comparison of single and multiple documents respectively. I demonstrate ConceptScope by visualizing research articles and transcripts of technical presentations in computer science. A version [347] of the research in this chapter was presented at ACM Conference on Human Factors in Computing Systems (CHI) 2021.

The next three chapters showcase visual-assisted knowledge exploitation (task 3) for three applications in different domains and scenarios. Respectively, Chapter 5 presents ConceptEVA, a technique that exploit knowledge from domain ontology to support document summary customization for academic reading. I describe a mixed-initiative approach to generate, evaluate, and customize summaries for long and multi-topic documents by leveraging an existing knowledge base, DBpedia, and human input. A version [349] of the research in this chapter was presented at ACM Conference on Human Factors in Computing Systems (CHI) 2023.

Chapter 6 presents LabelVizer, a technique that exploit knowledge from domain experts to support technical text validation and relabeling for smart manufacturing. LabelVizers is a human-in-the-loop workflow that incorporates domain knowledge and user-specific requirements to reveal actionable insights into annotation flaws, then produce better-quality labels for large-scale multi-label datasets. I present my workflow as an interactive notebook and report the evaluation result with with two use cases and four expert reviews. A version [351] of the research in this chapter was presented at IEEE Pacific Visualization Symposium 2023.

Chapter 7 presents SliceTeller, a technique that exploit knowledge from domain experts to support machine learning model validation for autonomous driving, smart home, and AI fairness. Besides, I present an efficient algorithm, SliceBoosting, to estimate trade-offs when prioritizing the optimization over certain slices. I report the evaluation results with three use cases, including two real-world use cases of product development, to demonstrate the power of SliceTeller in the debugging and

improvement of product-quality ML models. A version [350] of the research in this chapter was published in IEEE Transactions on Visualization and Computer Graphics and was presented at IEEE Visualization Conference 2022. This work also won the Best Paper Honorable Mention Award of IEEE Visualization Conference 2022.

Finally, Chapter 8 provides a succinct summary of the contents and insights gleaned from my dissertation work, as well as my vision for future research on visualization-assisted knowledge discovery and utilization.

Chapter 2
Knowledge Exploration with Multidimensional and Heterogeneous Data

In machine repair and maintenance, error diagnosis is a crucial task to identify abnormal patterns, formalize root-cause analysis, and plan preventive maintenance. To make such diagnoses, analysts often refer to maintenance logs. However, analyzing maintenance logs is not as straightforward because they tend to be large, multidimensional, and heterogeneous (i.e., consisting of numerical, categorical, and text components). This challenge is further compounded by inconsistent and/or missing entries. To conduct an effective diagnosis, it is important to *explore knowledge* from the data with support from analytic algorithms while involving the human in the loop. In this chapter, I introduce a visual analytics approach that uses multiple dimensionality reduction and clustering algorithms to visualize and group different components of the data. To help analysts label the clusters, each clustering view—one for each data type—is supplemented with visualizations that contrast a cluster of interest with the rest of the dataset. My approach assists analysts to make sense of machines' maintenance logs and their errors, and their gained insights help them carry out preventive maintenance. I illustrate and evaluate our approach through use cases and expert studies respectively, and discuss generalization of the approach to other heterogeneous data.

2.1 Introduction

Making sense of large-scale, heterogeneous data is one of the main challenges faced by data science and visualization communities in real-world application scenarios. For

instance, in large-scale manufacturing setups, human- and machine-created logs of operation and maintenance need to be analyzed to identify problem areas and prevent major failures before they occur [44]. The size of collected logs can easily number over hundreds of thousands of records and often include multiple types of data: numerical data (e.g., operating temperatures), categorical data (e.g. machine types), ordinal data (e.g. error severity), and text data (e.g. machine status description) [157]. In addition, manually-entered components—especially natural-language descriptions—can feature different forms of inconsistencies. For instance, the same problem may be described using different wording by different operators, or even the same operator at different times [314]. This type of information is captured by most maintenance departments, and the issues are thus ubiquitous across the maintenance industry. These factors make it difficult for managers and technicians—even with the help of data analysts—to analyze logs to identify patterns (e.g., common phenomena seen in some type of errors), and perform preventive maintenance.

Machine learning (ML) assisted visual analytics have been developed to address the challenge in reviewing large, high-dimensional data [301,343]. For instance, researchers have used dimensionality reduction (DR) to provide an overview of high-dimensional data in lower dimensions [169, 358] and clustering to summarize the information of large data into a small number of groups [18,189]. Contrastive learning, which extracts salient patterns in one dataset relative to the other, is then used to help interpret the results of DR and clustering [116, 118]. Maintenance log analysis can benefit from methods to extract and explain important patterns that are common across or specific to certain kinds of issues. At the same time, the problem of data inconsistency can be mitigated by keeping the human in the loop. However, these ML methods are designed to apply to a single data type, such as numerical or categorical. Thus, when analyzing heterogeneous data, we need to consolidate different methods. In addition, existing contrastive learning methods are applicable only to either numerical or binary data. New methods for other datatypes (e.g., categorical and text) are required.

In this work, we present an approach to separate different variable types—numerical,

categorical, and text—in a heterogeneous dataset and provide lower-dimensional, clustered visualizations for each type. We then use ccPCA [116]—contrasting clusters in Principal Component Analysis—as the contrastive learning method for *numerical* variables in the data. In order to provide a similar functionality for *categorical* variables, we introduce a method called contrasting clusters in Multiple Correspondence Analysis (ccMCA). ccMCA helps characterize a selected cluster (of categorical data) by comparing its attributes with those of the remaining data. For text variables, we first convert natural-language descriptions into high-dimensional vectors using word embeddings [332], and then perform DR and clustering. In place of contrastive learning, we plot text frequencies compare each cluster with the rest of the data.

Finally, we link the visualizations across all the views to help the analyst characterize clusters in the context of the other data dimensions. We illustrate our approach with use-case scenarios and expert reviews using a real-world dataset of maintenance and repair logs for heating, ventilation, and air-conditioning (HVAC) systems.

Our main contributions include: (1) an approach to analyze multidimensional, heterogeneous maintenance log data by separating numerical, categorical, and text dimensions and creating coordinated views of lower-dimensional projections and clusters for each data type, (2) a new contrastive learning method called ccMCA to help the user characterize data clustered on the basis of categorical dimensions, and (3) the application of existing methods to characterize the data clustered on the basis of numerical and text dimensions.

2.2 Related Work

While the proposed work falls under the application area of machine maintenance data analysis, our approach draws from and contributes to existing approaches in heterogeneous and high-dimensional data. We highlight representative research on these topics here.

2.2.1 Machine Maintenance Log Analysis

With an increasing emphasis on smart manufacturing and a push toward reducing machine down time, process monitoring, diagnostics, and prognostics have gained prevalence. This trend, coupled with cheaper and easier-to-obtain sensors and data storage solutions, has led to increases of maintenance data [44]. Despite the potential benefits of uses of high volume maintenance data for better machine management, companies frequently struggle to adopt advanced manufacturing technologies and strategies due to cost and lack of technical expertise in data analysis [166]. Simple yet powerful solutions for data analysis are necessary to aid manufacturers in improving their practices. Annotation methods for short-text maintenance work orders [224, 313] has been the subject of recent research. For instance, Sexton et al. [315] developed Nestor, an open-source tool that uses internal "importance" heuristics and domain-expert annotation of seed data to annotate large volumes of short texts, as maintenance logs tend to be, with domain-relevant tags[1].

Visual analytics is another technique that has gained popularity in this domain in recent years. Recent work includes ViDX [389], a visual analytic system for historical analysis and real-time monitoring of factory assembly lines. La VALSE [137] and MELA [318] are scalable visualization tools with multiple visualization interfaces incorporating different logs for interactive event analysis. ViBR [51] provides a visual summary of large bipartite relationships by using a minimum description length principle and is used for vehicle fault diagnostics. These approaches are created for specific data or a data type. On the other hand, we treat the data as high-dimensional, heterogeneous datasets that include unstructured text, thus allowing our approach usable across different organizations and domains.

2.2.2 Visualizing Heterogeneous Data

The challenges of visualizing heterogeneous data, i.e., data with mixed data types or variables, such as numerical, categorical, and text, were recognized early in visualization. Almost 25 years ago, Zhou and Feiner [407] provided a systematic approach

[1] https://nist.gov/services-resources/software/nestor

to design visualizations for heterogeneous data based on data characteristics and the tasks involved. The *size* of heterogeneous datasets poses additional challenges for visualization, such as requiring large screens and appropriate visual mappings. Different approaches were developed to address these challenges, such as developing automated specification algorithms to map data attributes to visual attributes [47], and high-resolution immersive visualization environments [290].

Visualizing heterogeneous data also provides a way for the user to establish *context*. For instance, coordinated timeline visualizations of audio, video, and text data of human-human or human-machine interactions can provide context to observations about movement, speech, and activity data [52, 112]. More recently, immersive visualizations of system activity overlaid on a spatial layout corresponding to the physical locations of said systems were used to provide contextual information in real-time network security analysis [227].

Unstructured text also forms an important datatype. Descriptive text about problems and repairs are often entered by operators and maintenance personnel who assume familiarity with the machines and related processes. The text thus tends to be terse and laden with jargon, and is often inconsistent across people. Developing a lexicon—a domain-specific vocabulary—is often necessary to interpret such text data in a semantically consistent way. The General Inquirer [339] is one of the earliest attempts to build a lexicon for content analysis of text. Categories such as Linguistic Inquiry and Word Count (LIWC) [281] focus on psychological relevance (such as moods) and general-purpose applications. Such models are trained on general text corpora such as news articles, online forums, and fiction. For application to large-scale technical text data, automated tagging needs to be balanced with manual sifting of the text.

Visual analytics has been used to achieve better balanced tags, using a combination of high-dimensional data visualizations and user-steered analyses. For instance, ConceptVector [272] visualizes word-to-concept similarities to guide users to categorize text data given a specific domain, such as politics or finance. Similar vector space representations are used by Heimerl and Gleicher [152] to design visualizations that help users

understand word vector embeddings. In addition, several tools such as the Exploratory Labeling Assistant [106] and AILA [64] use machine-learning based recommendations to help users characterize or label documents.

Drawing from this combination of statistical and manual approaches, we use word embeddings to translate short texts to high-dimensional vectors, and apply DR and clustering to find groups of semantically related short texts in a 2D space. We use similar DR and clustering representations for numerical and text data dimensions, which gives us consistent representations across data types.

2.2.3 Visualizing High-Dimensional Data

Most machine maintenance log data tend to be high-dimensional, with information about each breakdown or maintenance event consisting of multiple fields relating to different personnel and/or departments [314]. While high dimensionality has its advantages, such as the ability to contextualize and correlate features of the data, it also makes the data less usable for any form of sampling or statistical analysis [90]. Dimensionality reduction provides a lower-dimensional representation while still preserving the essential information of the original data [360]. Nonlinear DR techniques such as t-SNE [358], LargeVis [345], and UMAP [233] are especially relevant for large-scale, high-dimensional data as they preserve local neighbor relationships, which can help identify subgroups in the data.

DR can be further exploited to cluster the data with higher speed and performance [320] or to produce an overview of the data [216, 302]. During this process, visual analytics of the clustered data is often needed to help users determine *which* attributes contribute to the distinctness of each cluster [39]. Statistical charts (e.g. boxplots) [189] or density plots [335] of selected clusters from the DR result have been used for this purpose. However, showing one statistical chart for each attribute becomes visually overloaded as the number of attributes increases. A better approach would be to identify and visualize salient attributes that contribute to a selected cluster. For instance, Broeksema et al. [41] visualized the results of multiple correspondence analysis (MCA)—a variant of principal component analysis (PCA) for categorical data—together

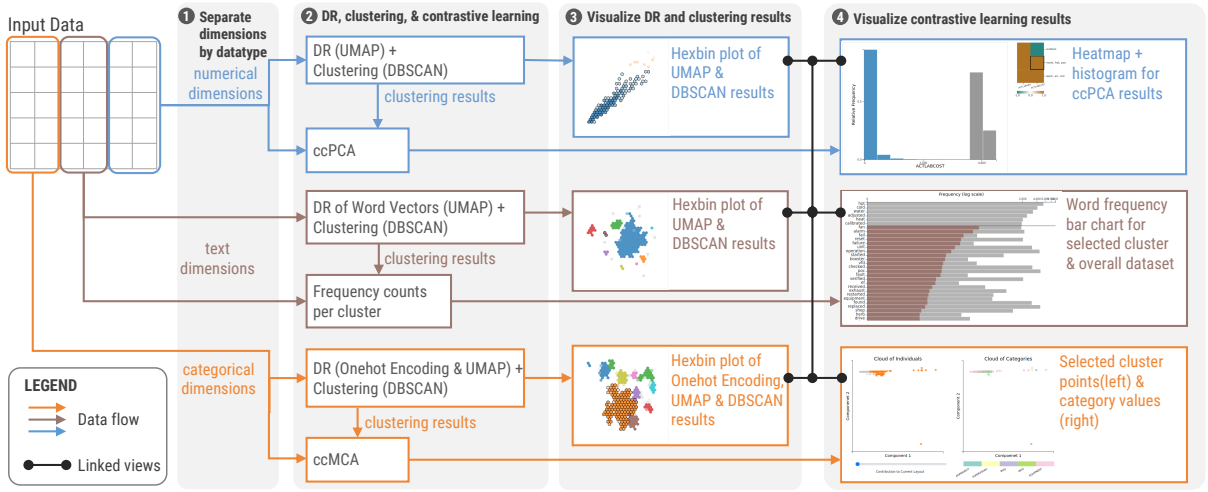


Figure 2.1: Data processing pipeline for individual views, based on the category of data dimension (categorical, text, and numerical). The figure also shows which views are linked via selection and filtering interactions.

with a colored Voronoi cell that represents a highly-related attribute to each data point. Similarly, Joia et al. [169] drew a convex hull around each cluster and filled the resulting polygon with a word cloud consisting of names of the attributes related to the cluster. Faust et al. [101] took a different approach, using local perturbations in the input data to represent how the higher dimensions are represented in the projected views. More recently, Fujiwara et al. [116] used contrastive learning to find attributes that contrast a selected cluster from the rest of the data. We incorporate this contrastive learning-based approach to analyze the numerical attributes of the maintenance log data, while introducing an analogous approach for the categorical attributes.

2.3 Requirements

Typically, visual analysis of heterogeneous, multidimensional data is performed with the goal of identifying patterns within the data and extracting meaning from them [13, 385]. With our application area in mind, we draw our requirements from existing work on maintaining and tagging machine performance, error, and maintenance log data.

Most of our requirements are based on prior work by Brundage et al. [42, 44] who generate a set of commonly-occurring data elements from their study of various maintenance work order datasets including temporal (e.g., time between failures, machine down time etc.), machine (machine type, location etc.) human (operator/tech name,

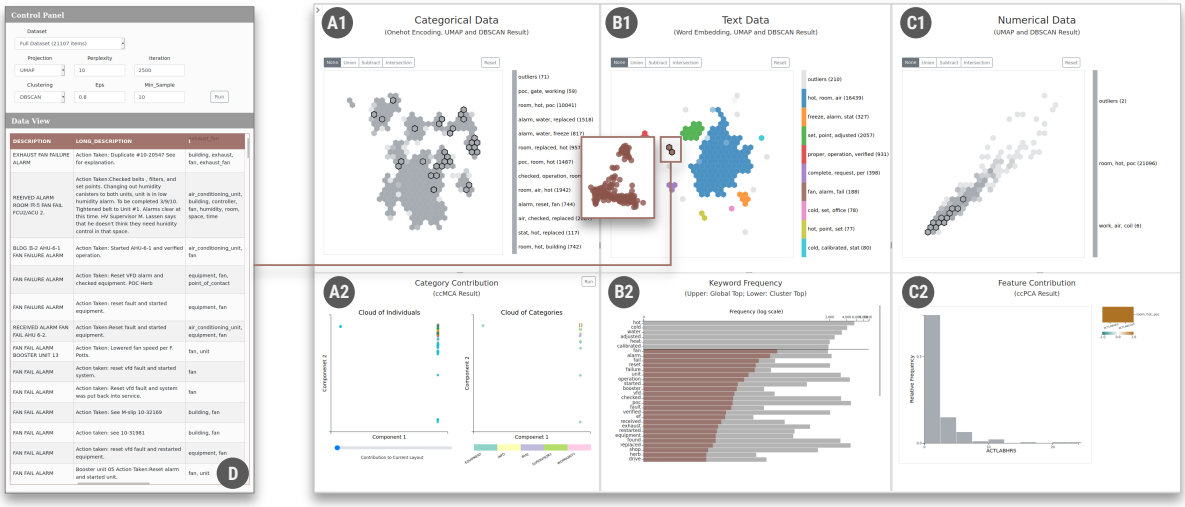


Figure 2.2: Dashboard interface showing projected views of categorical components of the dataset using MCA (A1), text components of the dataset using UMAP (B1), and numerical components of the dataset, also using UMAP (C1). Each projected view is clustered using a chosen clustering algorithm (DBSCAN in the example above). Each projected view is supported by an additional view that is used to characterize a chosen cluster in that view. For the categorical data view, ccMCA (A2) is used to show the selected cluster's separation and the attribute values that contribute to it. A text frequency chart (B2) contrasts the text that occurs most frequently in the selected cluster against the overall text frequency in the dataset. Finally, ccPCA is used to display a heatmap of cluster vs. data dimensions and a histogram showing the value distribution of a selected numerical dimension against the rest of that data (C2). Raw data for any chosen cluster can be viewed using a slide-out tabular view (D). Linking across views A1, B1, and C1 shows the distribution of data clustered in the active view (in color) across the other two views (grayscale).

skill level etc.), raw text (problem descriptions, solution etc.), and tagged elements (items, actions, etc.). Broadly speaking, these elements can be classified based on their data type as numerical, categorical, and text. They also propose a maintenance management workflow with six steps: (1) analyzing the work order, (2) selecting & prioritizing work orders, (3) planning equipment, resources, and labor, (4) scheduling the tasks involved, (5) executing the tasks, and (6) completing and documenting the tasks performed. Our goal is to aid the user—assumed to be a planning engineer or an analyst—in the execution of Steps 1–3. Depending on the scenario, this may require accurate identification of the maintenance task involved, using maintenance logs to anticipate component failure, or correcting work orders with misdiagnosed problems or misidentified tasks.

We thus infer that a system that uses maintenance log data to aid maintenance

planning and management needs to be robust to different types of data dimensions, supports visual analysis of data at scale, and helps the user characterize and label parts of the data based on their domain knowledge. The system requirements are:

- **R1 Robustness to Data Type:** The system should accommodate all three types of data commonly required for the analysis of maintenance logs, i.e., numerical, categorical, and text data. Given the inherent difference between the data types, an appropriate analysis approach is needed for each.
- **R2 Scalability:** Maintenance log data in an organization can vary from a few thousand records to hundreds of thousands of records, depending on the organization size. With each record consisting of several dimensions of mixed data types, the system needs to be robust to different data scales.
- **R3 Data Subset Identification:** When visualizing large-scale data with heterogeneous dimensions, it is not optimal or practical to start by examining individual data points. It is more important and efficient to be able to identify subsets comprising data points that are closely related to each other. This may mean that all data points in an identified subset have common attributes, or that they may be related to each other based on their values along multiple dimensions. With different dimensions composed of different data types, the system should allow subset identification approaches suitable across data types.
- **R4 Data Subset Characterization:** Analyzing maintenance log data requires not only the identification of patterns/subsets within the data, but also their *characterization*, or what separates them. For instance, a problem common to a group of machines could be characterized by all machines being similar (e.g., lathes) or all machines requiring replacement of the same component or components supplied by the same vendor. Identification of such common characteristics become more difficult as the relationship shared by a subset of maintenance logs becomes more complex. Thus, the system should provide an effective analysis support to characterize the subsets from many dimensions.
- **R5 Extensibility:** Different organizations may choose to log information about their

maintenance activity in different forms and granularities. The only aspects that may be common across these datasets is that they are multidimensional and heterogeneous. The system should be extensible to different datasets with minimal effort, not overly dependent on any one specific dataset's attributes or format.

2.4 Data Processing & Visualization

Based on the requirements identified in section 2.3, it is clear that the three types of data common to machine maintenance logs—numerical, categorical, and text—need to be processed appropriately and visualized using approaches that are robust to changes in the data scale. In this section, we describe the data processing approaches and visualization designs that address the identified requirements.

2.4.1 Workflow

In section 2.2, we see that visualizing heterogeneous data is advantageous as it allows the user to draw inferences based on context from different data dimensions. We also see that the issues of scale and dimensionality make it challenging for such observations and inferences to be drawn. Both issues are addressed by using clustering techniques to form subsets within the data (**R3**). These can then be visually and interactively explored to understand the relationship between the data points that make up the subset.

To aggregate the techniques mentioned above, we model our data processing and visualization workflow as a pipeline with six steps: **Step 1**: grouping the data dimensions together based on their data type (Figure 2.1 stage 1); **Step 2**: performing DR for numerical, categorical, and text data separately and obtaining a 2D projection for each (Figure 2.1 stage 2); **Step 3**: clustering the 2D data to form subsets (Figure 2.1 stage 2); **Step 4**: visualizing the 2D projection and clustering results to provide scalable overviews of the dataset (Figure 2.1 stage 3 and Figure 2.2 A1, B1, C1); **Step 5**: characterizing the clusters separately for each data type using contrastive learning or statistical methods (Figure 2.1 stage 2); **Step 6**: cluster characterization for each data type with an appropriate visualization (Figure 2.1 stage 4 and Figure 2.2 A2, B2, C2). Each step is detailed in the rest of this section.

2.4.2 Identifying Subsets in Heterogeneous Data

DR (step 2) and clustering (step 3) are two essential data processing steps to identify subsets in the data. Based on our review of DR and clustering techniques in section 2.2.3, we use Uniform Manifold Approximation and Projection (UMAP) to project the data to a lower-dimensional space. Note that high-dimensional representations are obtained for the categorical data with one-hot encoding, and for text with word embeddings (see section 2.4.4.1) before the DR step. The 2D projection of the data can then be clustered using any approach.

We choose DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [99] as it uses a density-based approach that is more suitable for data that may have outliers (e.g., uncommon machine breakdown or repair). Combined with our visualization approach, this technique is more suitable for our case as the analyst can probe into individual records in the case of outliers, and can also examine larger clusters using the linked views. By separating the data dimensions based on data type, we ensure that our approach is robust to datasets with different dimensions with mixed data types (requirement **R1**). This approach of dimension grouping by data type, DR & clustering to find subsets, and characterizing based on contrastive learning and text frequency comparison makes our approach extensible to most heterogeneous datasets (**R5**).

2.4.3 Scalable Overview

To show an overview of the DR and clustering results (step 4), we use a hexbin plot [48] for each data type in the dashboard visualization shown in Figure 2.2, i.e., categorical (Figure 2.2A1), text (B1), and numerical (C1) dimensions. The hexbin plot is robust to different data scales (requirement **R2**) in that its rendering speed is not significantly impacted by data size or screen resolution. Instead of using a linear color scale typical to hexbin plots, we use a different hue for each cluster and map data density to color intensity within every cluster.

We also preserve the conventional DR representation, i.e., a scatterplot with each data object shown by a dot. We adopt Lindstrom's [211] Level of Detail (LOD) rendering and allow users to switch between these two plots or change the granularity of

hexagonal bins by simply zooming in or out of the area they are interested in. Thus, only a small part of the scatterplot needs to be rendered when the users zoom in close. Finally, users can choose to examine the data objects in detail by perusing the slide-out tabular view (Figure 2.2D) or by hovering over the dots.

Note that at any point, only one of the three clustered views (A1, B1, or C1 in Figure 2.2) can be active. The active view is indicated by its clusters highlighted with a categorical color palette. The remaining views are monochromatic/greyscale to prevent the user from mistakenly assuming that a cluster of one color (e.g., blue) in one view corresponds to a cluster of the same color in another view.

2.4.4 Characterizing Clusters

Characterizing a cluster or subset in the data (requirement **R4**) requires determination of how the cluster is different from the rest of the data. Different data types necessitate different contrastive analysis techniques. We discuss the techniques we use to characterize clusters for text, numerical, and categorical data in this subsection.

2.4.4.1 Text Dimensions

Detailed text descriptions of problems, symptoms, and solutions, form perhaps the richest component of maintenance log data. They are also rife with inconsistencies, typographical errors, or the use of non-standard shorthand that is endemic to that particular organisation. Text descriptions are also often supplemented by "tags"—standardized phrases that label the descriptions to identify the problems, items, and solutions. These tags are typically assigned partly based on the knowledge of the user who tags the descriptive text, and partly using machine learning approaches [313,315].

In order to group the data based on text dimensions, the *meaning* of the text needs to be considered instead of specific keywords that may vary across technical personnel. To convert this text into a more consistent semantic representation, we use word embeddings, which are vector representations of words that take into account their semantic relationships [357]. Words such as "warm" and "hot" can thus be translated to vectors that are close to each other, but distant from a vector representing a word different in meaning, such as "telephone". We create high-dimensional vector representations

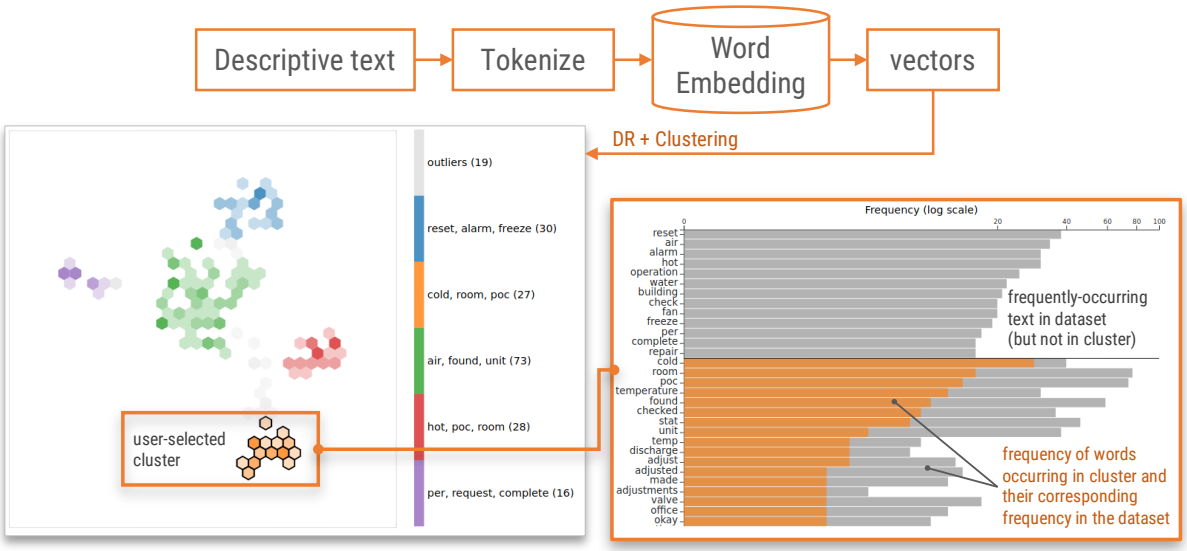


Figure 2.3: Text processing steps and its display in the text panel. The supporting view shows frequencies of text occurring in the selected cluster and contrasts it with both, the frequencies of corresponding text in the overall dataset as well as the text that is most frequently-occurring in the dataset but not in the selected cluster.

for the descriptive text by summing and normalizing words in the text. We then use a suitable DR technique (UMAP) to obtain 2D projections of the vectors, and cluster them using DBSCAN (Figure 2.3).

Each cluster represents a collection of descriptions. To characterize a given cluster, we overlay a frequency plot of the most common terms occurring in the cluster on a frequency plot of terms occurring in the overall dataset (Figure 2.3 right). Contrasting the most frequent terms of both plots helps the user identify defining characteristics of the cluster. For instance, examining the frequency plots of Cluster 1 in Figure 2.3, we can surmise that the cluster represents maintenance logs of ventilation systems related to lower room temperatures, commonly remedied by adjusting certain valves. The user can examine the raw data related to any cluster using the slide-out tabular view (Figure 2.2D) to further gain insight into the cluster and characterize it (requirement **R4**). Each of the cluster labels is editable, so the user can change the cluster name from "Cluster 1" to a more descriptive "Lower temperature adjustment".

Table 2.1: Summary of notations.

$\mathbf{X}_T$, $\mathbf{X}_B$	target, background matrices
$\mathbf{X}_E$, $\mathbf{X}_K$, $\mathbf{X}_R$	matrices of the entire, target cluster, rest data points
$\mathbf{C}_T$, $\mathbf{C}_B$, $\mathbf{C}_E$, $\mathbf{C}_R$	covariance matrices of $\mathbf{X}_T$, $\mathbf{X}_B$, $\mathbf{X}_E$, $\mathbf{X}_R$
$\mathbf{G}_T$, $\mathbf{G}_B$, $\mathbf{G}_T$, $\mathbf{G}_R$	disjunctive matrices of $\mathbf{X}_T$, $\mathbf{X}_B$, $\mathbf{X}_E$, $\mathbf{X}_R$
$\mathbf{Z}_T$, $\mathbf{Z}_B$, $\mathbf{Z}_E$, $\mathbf{Z}_R$	probability matrices of $\mathbf{G}_T$, $\mathbf{G}_B$, $\mathbf{G}_T$, $\mathbf{G}_R$
$\mathbf{B}_T$, $\mathbf{B}_B$, $\mathbf{B}_E$, $\mathbf{B}_R$	Burt matrices of $\mathbf{X}_T$, $\mathbf{X}_B$, $\mathbf{X}_E$, $\mathbf{X}_R$
α	contrast parameter

2.4.4.2 Numerical Dimensions

As Brundage et al. [42] illustrate with different maintenance key performance indicators (KPIs), measures such as problem/breakdown counts and time between failures are ways to quantify the role of other performance indicators such as machine type, problem severity, and technician skill. Other parameters such as cost can be derived from these factors. To understand how those parameters contribute the separation of clusters for numerical data, we adopt a method called contrasting clusters in PCA (ccPCA) [116]. We briefly describe ccPCA and its application to our system. Notations used in the following sections are summarized in Table 2.1.

Introduction to cPCA. cPCA aims to reveal enriched patterns in a target matrix $\mathbf{X}_T$ relative to a background matrix $\mathbf{X}_B$. To do so, cPCA finds directions (called contrastive principal components, cPCs) that maximally preserve the variation in $\mathbf{X}_T$ while simultaneously minimizing the variation in $\mathbf{X}_B$. This can be achieved by performing EVD on $(\mathbf{C}_T - \alpha\mathbf{C}_B)$ where $\mathbf{C}_T$ and $\mathbf{C}_B$ are covariance matrices of $\mathbf{X}_T$ and $\mathbf{X}_B$, respectively. α $(0 \leq \alpha \leq \infty)$ is a hyperparameter, called a contrast parameter, which controls the trade-off between having high target variance and low background variance. When $\alpha = 0$, the resultant cPCs only maximize the variance of $\mathbf{X}_T$ (i.e., the same with using ordinary PCA). As α increases, cPCs place greater emphasis on directions that reduce the variance of $\mathbf{X}_B$.

Introduction to ccPCA. In order to characterize clusters, ccPCA utilizes cPCA as its base. Let $\mathbf{X}_E$, $\mathbf{X}_K$, and $\mathbf{X}_R$ be matrices of the entire dataset, a target cluster selected from the entire dataset, and the rest of data points, respectively. ccPCA enhances the

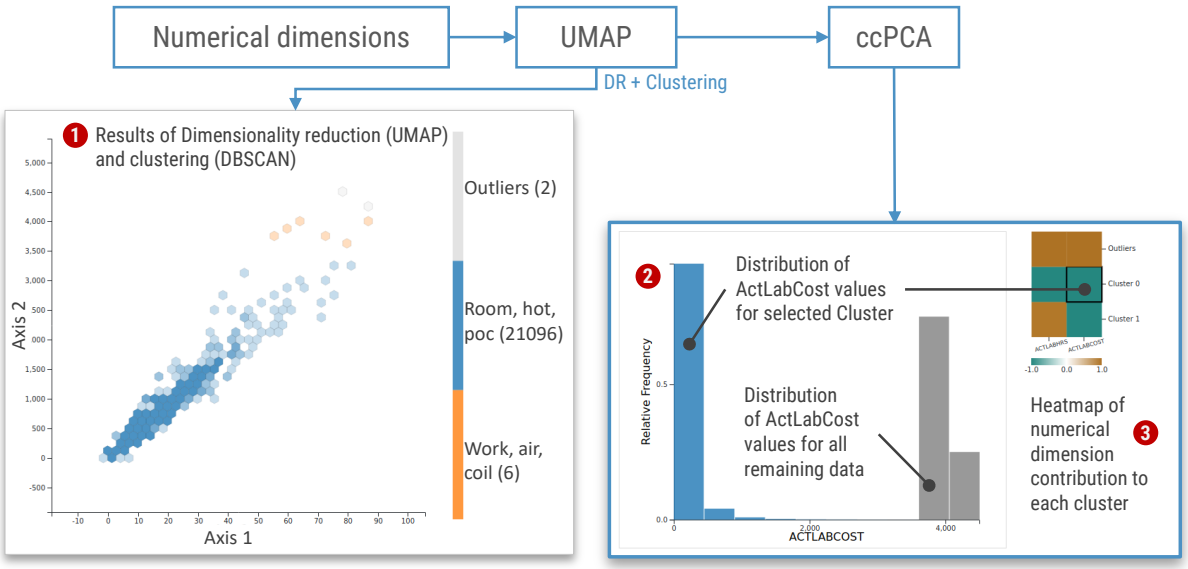


Figure 2.4: Projection and visualization of numerical data along with the ccPCA view shows that the selected cluster has lower labor cost than the rest of the data.

original cPCA by using $\mathbf{X}_E$ as a target matrix and $\mathbf{X}_R$ as a background matrix, instead of using $\mathbf{X}_K$ and $\mathbf{X}_R$ as target and background matrices, respectively. ccPCA finds the directions that preserve both the variety and separation between a target cluster and others. By referring to feature contributions (called contrastive principal component loadings or cPC loadings) to the directions, we can obtain the information of which numerical features contribute to the uniqueness of a target cluster relative to others.

Visualization. ccPCA provides how strongly each dimension contributes (positively or negatively) to each cluster's contrast with the rest of the data. This contribution is shown as a heatmap (Figure 2.4(3)) that indicates the magnitude and direction of contribution of the numerical dimensions to each cluster with a blue-green-to-brown diverging colormap [148]. By selecting a cell in the heatmap, Figure 2.4(2) shows histograms of the corresponding dimension's value distributions of the selected cluster and the rest of the data with the cluster color and gray color, respectively. Based on Figure 2.4(2), we can infer that the numerical dimension "actual labor cost" (`ActLabCost`) contributes strongly to Cluster 0's contrast against the rest of the data, and the histograms show that the `ActLabCost` values for the selected cluster are much lower than the rest of the data. The user can further investigate this cluster by selecting it in the

Table 2.2: Comparison of representation learning methods. ccMCA is a new method we introduce in this work.

data type	method	purpose	solution
numerical, binary	PCA	preserving the variance of $\mathbf{X}_T$	EVD on $\mathbf{C}_T$
	cPCA	identifying enriched patterns in $\mathbf{X}_T$	EVD on $(\mathbf{C}_T - \alpha\mathbf{C}_B)$
	ccPCA	characterizing a cluster $\mathbf{X}_K$	EVD on $(\mathbf{C}_E - \alpha\mathbf{C}_R)$
categorical, binary	MCA	preserving the variance of $\mathbf{X}_T$	EVD on $\mathbf{B}_T$
	cMCA	identifying enriched patterns in $\mathbf{X}_T$	EVD on $(\mathbf{B}_T - \alpha\mathbf{B}_B)$
	ccMCA	characterizing a cluster $\mathbf{X}_K$	EVD on $(\mathbf{B}_E - \alpha\mathbf{B}_R)$

DR view (Figure 2.4(1)) to examine the corresponding data distribution in the text and categorical dimension views as described in subsection 2.4.3, or examine the cluster in detail using the tabular view (Figure 2.2(D)). Note that even though only two numerical dimensions are shown in Figure 2.4 (C1 & C2), where making the DR step redundant, the clustering and ccPCA computations are still useful in identifying and characterizing subsets within the data.

2.4.4.3 Categorical Dimensions

For the characterization of categorical data (**R4**), we cannot use ccPCA as it requires the data to be numerical or binary. Thus, we introduce a new contrastive learning method, called contrasting clusters in MCA (ccMCA)[2] by extending multiple correspondence analysis (MCA). Table 2.2 compares the related methods.

Multiple Correspondence Analysis (MCA) Here, we provide a brief introduction to MCA. MCA can be considered as PCA for categorical data analysis. That is, MCA learns a lower-dimensional representation from high-dimensional categorical data as it maximally preserves the variance of the data. The issue of PCA when applying to categorical data is that PCA handles each category in the data as a numerical value and, as a result, it unnecessarily ranks the categories (e.g., red: 0, green: 1, blue: 2).

To avoid this, MCA first converts an input matrix $\mathbf{X}_T$ of categorical data into a disjunctive matrix $\mathbf{G}_T$ (or disjunctive table) by applying one-hot encoding to each categorical dimension. For example, when $\mathbf{X}_T$ consists of two columns (or often called questions) of "color" and "shape" and each has categories (i.e., categorical answers)

[2]The source code of ccMCA will be released upon publication.

of {"red", "green", "blue"} and {"circle", "rectangle"}, $\mathbf{G}_T$ will have five columns of "red", "green", "blue", "circle", and "rectangle" and each of the matrix element will be either 0 or 1. Afterward, by dividing each cell in $\mathbf{G}_T$ with a total of $\mathbf{G}_T$, we obtain a probability matrix (or correspondence matrix) $\mathbf{Z}_T$. This probability matrix corresponds to an input feature matrix for PCA. Similar to PCA, we apply normalization to $\mathbf{Z}_T$. With the normalized $\mathbf{Z}_T$, we can obtain a Burt matrix $\mathbf{B}_T$ with $\mathbf{B}_T = \mathbf{Z}_T^\top \mathbf{Z}_T$. $\mathbf{B}_T$ corresponds to a covariance matrix used in PCA (note: in PCA, a covariance matrix of $\mathbf{X}_T$ can be obtained with $\mathbf{C}_T = \mathbf{X}_T^\top \mathbf{X}_T$). Thus, as PCA obtains principal components by performing eigenvalue decomposition (EVD) on $\mathbf{C}_T$, MCA obtains the principal directions by performing EVD on $\mathbf{B}_T$ to preserve the variance of $\mathbf{G}_T$.

Contrastive MCA (cMCA) Now, we introduce contrastive version of MCA (cMCA) [117] and enhance cMCA to ccMCA in the next subsection. As described above, MCA and PCA fundamentally share the same idea of finding the best directions to preserve the variance by using EVD on a covariance matrix. Therefore, we can extend MCA to cMCA by employing the same idea with cPCA.

Extension from MCA to cMCA. As described in section2.4.4.2, the only difference between PCA and cPCA is that while PCA directly performs EVD on a target covariance matrix $\mathbf{C}_T$, cPCA takes a subtraction of target and background covariance matrices with a contrast parameter (i.e., $\mathbf{C}_T - \alpha\mathbf{C}_B$) and then performs EVD on it. To reveal enriched patterns in a target matrix of categorical values, we can use the same idea that we use with cPCA and apply it to MCA. As stated in section 2.4.4.3, in MCA, a Burt matrix $\mathbf{B}_T$ contains similar information with a convariance matrix $\mathbf{C}_T$ in PCA. Therefore, we can obtain contrastive directions by computing $\mathbf{B}_T - \alpha\mathbf{B}_B$, where $\mathbf{B}_T$ and $\mathbf{B}_B$ are target and background Burt matrices, and then performing EVD on $(\mathbf{B}_T - \alpha\mathbf{B}_B)$. Here, α $(0 \leq \alpha \leq \infty)$ is also a contrast parameter and has the same role with cPCA.

Contrasting Clusters in MCA (ccMCA) For the cluster characterization, we enhance cMCA to ccMCA. Here, we apply the similar idea of the extension from cPCA to ccPCA.

Extension from cMCA to ccMCA. cMCA can be enhanced to ccMCA by using $\mathbf{X}_E$ and $\mathbf{X}_R$ as input target and background matrices. Since the directions identified by

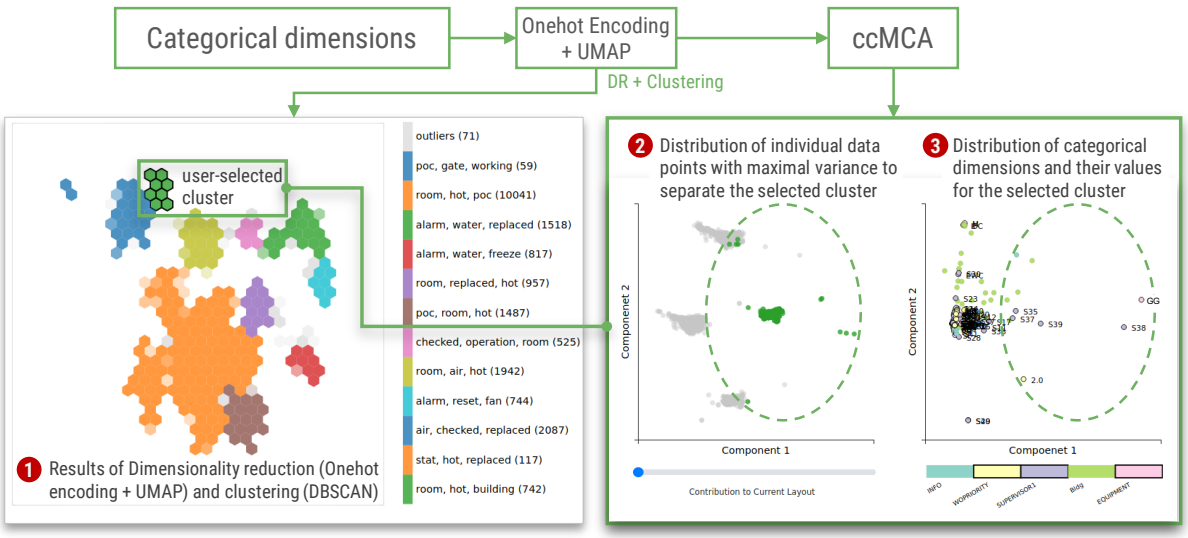


Figure 2.5: Projection of categorical data (1), with the ccMCA view showing the separation of the selected cluster (2), and its corresponding category distribution (3). The categories of "EQUIPMENT" with value "GG", "WOPRIORITY" (work priority) and "SUPERVISOR1" ids appear to be the what separate this cluster from the rest.

ccMCA differs based on the contrast parameter α, we also provide the automatic selection method of α by employing the same method introduced by Fujiwara et al. [116], which utilizes the histogram intersection for its optimization. Figure 2.5(2) shows the ccMCA result when selecting the green cluster from Figure 2.5(1) as a target cluster. The green points are clearly separated from others while keeping a high variance.

One ccMCA's major and different challenges from ccPCA is that how we inform the feature contributions. ccMCA also provides contributions (or loadings) of each dimension (i.e., category) of $\mathbf{G}_T$ with $\mathbf{w}_i = \sqrt{\lambda_i}\mathbf{v}_i$ where $\mathbf{w}_i$ is feature contributions to the i-th principal direction, λ_i is the i-th top eigenvalue generated via EVD, and $\mathbf{v}_i$ is the corresponding eigenvector. Because EVD is performed on Burt matrices of $\mathbf{G}_T$ and $\mathbf{G}_B$, which are obtained by applying one-hot encoding to $\mathbf{X}_T$ and $\mathbf{X}_B$, $\mathbf{w}_i$ shows a contribution for each category (e.g., "red", "green", and "blue") but not for each question (e.g., "color"). Therefore, the number of dimensions of $\mathbf{w}_i$ can be easily overwhelmed. For example, when there are 6 questions and 5 categories for each question, the number of dimensions in $\mathbf{w}_i$ becomes 30. Also, as each data point's position in a ccMCA projection (e.g., Figure 2.5(2)) reflects a compound of contributions, looking at each contribution may not be sufficient to understand the association between

the projection and contributions. For instance, even when one category may have a strong contribution to positive direction of the first axis (x-axis in Figure 2.5(2)), this does not ensure that data points with large positive x-coordinates have answered the corresponding category because, at the same time, many other categories may have a weak contribution to the positive direction.

To address this issue, similar to MCA, we provide the *principal cloud of categories* (or *column principal coordinates*), as shown in Figure 2.5(3). In MCA, the principal cloud of categories (PCC) is used to grasp which categories each data point likely have answered by comparing the positions of data points in a MCA projection (or the *principal cloud of individuals*, PCI) and categories in PCC. When a data point in PCI is placed at a close position with certain categories in PCC, this data point tends to have these categories as its answers. We can also perform the same analysis above for ccMCA.

In MCA, PCC $\mathbf{Y}_T^{\text{col}}$ is usually obtained by taking a product of a diagonal matrix $\mathbf{D}_T$ of the sum for each column of $\mathbf{G}_T$ and the top-k eigenvectors $\mathbf{W}_T$ obtained by EVD (i.e., $\mathbf{Y}_T^{\text{col}} = \mathbf{D}_T\mathbf{W}_T^\top$). However, because ccMCA performs EVD on $(\mathbf{B}_T - \alpha\mathbf{B}_B)$ and the result is influenced by $\mathbf{X}_B$ as well, we cannot compute PCC with the above manner. Thus, instead, we use MCA's *translation formula* from PCI to PCC. The translation from PCI to

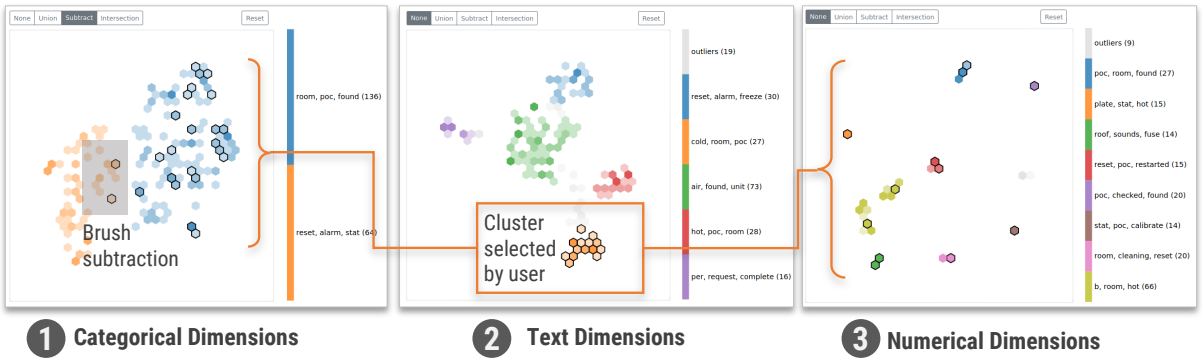


Figure 2.6: Linking between the projected views of categorical, text, and numerical data allows the user to explore the data clusters from the perspective of data types. For instance, selecting Cluster "cold, room, poc" from the projected and clustered view of the text dimensions (2) highlights the distribution of the same points in the other two views (1) & (3). We can see some correlation between the selected cluster and the "room, poc, found" cluster in the categorical dimension view. The brush and Boolean subtraction tools can be used to refine the selection and reveal the correlation between the two clusters.

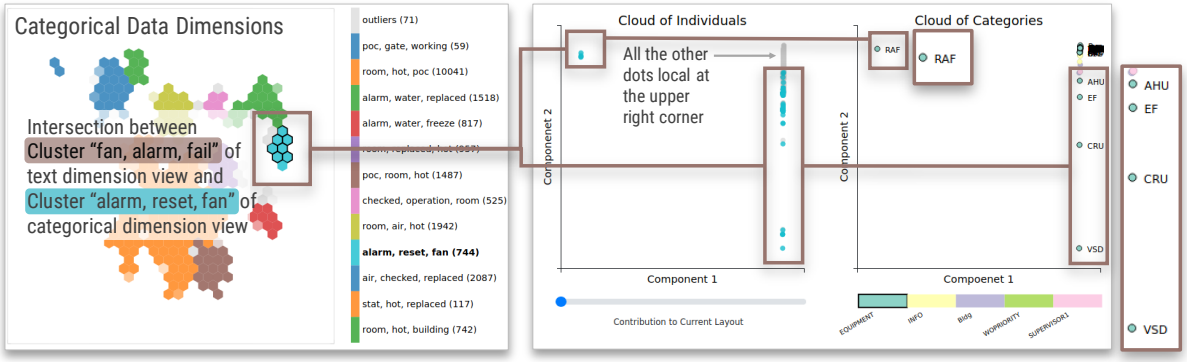


Figure 2.7: Characterising clusters in the use case scenario (section 2.6).

PCC can be performed with:

$$\mathbf{Y}_{\text{col}} = \mathbf{D}_T^{-1}\mathbf{Z}_T^{\top}\mathbf{Y}_T^{\text{row}}\text{diag}(\boldsymbol{\lambda})^{-1/2} \tag{2.1}$$

where $\mathbf{Y}_T^{\text{row}}$ is PCI of a target matrix and $\boldsymbol{\lambda}$ is a vector of the top-k eigenvalues. An example of the resultant PCC is shown in Figure 2.5(3). By referring to Figure 2.5(2) and (3), the analyst can characterize a selected cluster by understanding which categories highly associated with the uniqueness of the cluster.

2.4.5 Linking and Interactions

The visualizations across all six panels of the dashboard along with the tabular view are fully linked, and support brushing and direct selection (of a bin/cluster). For instance, users can select a cluster of interest in one of the three projected 2D views (A1, B1, or C1 in Figure 2.2), and observe the distribution of the cluster in the remaining two views. Each projected view is supported by a cluster characterization view (A2, B2, and C2 in Figure 2.2). When a cluster is selected from any one of the projected views, all three characterization views update to show the results of that cluster's characterization analysis based on categorical (A2), text (B2) and numerical (C2) data dimensions. The tabular view also updates to show the attributes of the data in the selected cluster.

The linked views update in a similar manner even if—instead of selecting a cluster—the user selects, say, a single hexbin, or brushes across multiple hexbins. Boolean operations such as union, intersection, and difference are also supported for more sophisticated selection of data across the three projected views. For instance, the user

can intersect multiple clusters across different views to find points common across clusters, or combine the clusters by union.

Figure 2.6 shows an example of interactive linking. The user selects a cluster labeled "cold, room, poc" in panel 2 (projected view of text). This highlights hexbins in the other two views that correspond to this cluster. In the example shown, most of the data points overlap with Cluster "room, poc, found" in panel 1 (categorical dimensions), indicating a correlation between these two clusters. To better observe the overlapping points, the user subtracts the two outliers in panel 1 by brushing them out, and checks the supplementary views. Panel 3 shows the points distributed across clusters, indicating no correlation between the selected clusters along their numerical dimensions.

2.5 Implementation

The dashboard visualization is implemented as a web framework, with a Flask server at the backend. The separation of numerical, categorical, and text dimensions is currently performed manually. We conduct dimensionality reduction and clustering at the backend for each of these three dimensions and visualize the results by creating an interactive web-based dashboard application. We use HTML/JavaScript for the frontend using Bootstrap and React libraries, and D3 [34] to create the interactive visualizations.

We use the Scikit-Learn [278] machine learning library for most of the DR and clustering algorithms, except for UMAP and ccPCA, for which we use implementations by McInnes et al. [233] and Fujiwara et al. [116] respectively. We use our own implementations of MCA and ccMCA for DR and contrastive learning for categorical dimensions. For the text dimensions, we use the Natural Language Toolkit (NLTK) [221] for the text processing, ConceptNet Numberbatch [332] as the word embedding to vectorize the text, and Gensim [291] to perform the word-vector lookup.

Using NLTK, we tokenize the descriptive text and tags, and remove stop words. Of the remaining text, individual words are looked up in the word embedding to return their vectors. These vectors are then added and normalized to obtain a single vector representing the unstructured text component of each data point. While ConceptNet

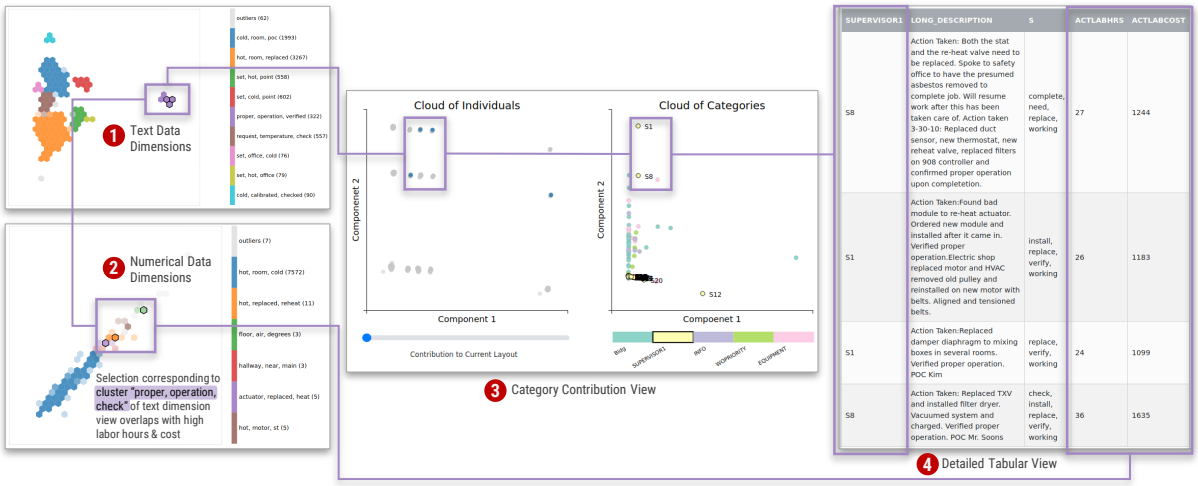


Figure 2.8: Examining a subset of the original data characterized by temperature-related complaints. The selected purple cluster is—using the category contribution view (3)—shown to be related to high costs associated with two supervisors (see section 2.6).

Numberbatch contains a fairly large vocabulary of over 500,000 words, there may be domain- or organization-specific terms used in maintenance logs that are not present in the word embedding. In our current implementation, we discard these terms on the assumption that enough of the meaning is captured in the rest of the text for clustering data. However, in future iterations, we plan to update word embeddings using vocabulary from technical manuals and organizational documentation.

2.6 Use Case Scenario

We illustrate the use of our system using maintenance log data of heating, ventilation, and air conditioning (HVAC) systems used in multiple office buildings of an organization. The maintenance logs consist of over 21,000 records collected over ten years, and contain multiple dimensions of categorical, text and numerical data. For the purpose of this use-case scenario, we select dimensions of the data that have the least number of missing fields. The dataset is grouped by the following sets of dimensions.

The first group involves the categorical dimensions of (1) `building number` in which the HVAC system is installed or where a complaint was recorded, (2) `equipment type` of the HVAC subsystem or machine, (3) `priority` of the work order, (4) `location` of the system or complaint (building number + floor + room), (5) the index of the `supervisor`

in charge of the systems at the time of logging the problem/solution.

Most of the numerical dimensions in the data involve dates and times of logging, and are not accurate or consistent enough to compute meaningful timespans. The second group thus involves the two remaining numerical dimensions of (1) `actual labor hours` incurred, and (2) `actual labor cost` incurred.

The third group consists of the text dimensions of (1) `long description` or a description of the problem or complaint that needed addressing, (2) `description` or a small set of keywords highlighting the important aspects of the problem, and (3) a set of multiple `tags` assigned to each maintenance record. The text fields were cleaned to remove extraneous characters (e.g., HTML tags, symbols, URLs etc.), remove punctuation, normalize whitespace sequences, and correct typographical and unicode errors.

Our scenario involves Alice, a maintenance supervisor responsible for the smooth running of HVAC systems across the organization. Alice uses our prototype to examine the dataset and identify patterns in the logs to identify potential issues and plan preventive maintenance. One of her initiatives has been to try and allocate manpower for recurring or preventable maintenance problems.

Overview. Alice loads all three data groups discussed above into the prototype to get an overview of the data after DR and clustering. She looks over the default tags assigned to each cluster and notices such commonly-occurring terms as *"room"*, *"air"*, *"hot"*, and *"cold"*. The largest cluster in panel B1 (Figure 2.2) showing text dimensions appears to contain complaints related to room temperature, with the tags *"too hot"* and *"too cold"* being the most common. From experience, she figures these represent the most typical complaints about HVAC systems in offices. The top keywords in the frequency plot (B2 in Figure 2.2) confirm her hunch.

Looking over at the numerical data projection (C1 in Figure 2.2), Alice notices that it appears to be linearly correlated. Examining the heatmap in the feature contribution panel (C2 in Figure 2.2) confirms this observation as she finds that the *"actual labor hours"* do indeed correlate with *"actual labor cost"*. She makes a mental note to refer to the correlation to filter the data by time or cost in her analysis.

Cluster characterization. Apart from the clusters related to the temperature problems, Alice notices a unique brown cluster in the text dimension views tagged as "*fan, alarm, fail*" (B1 in Figure 2.2) and decides to take a closer look at it. From panel B2, she finds that the top keywords in this cluster—*fan, alarm, fail, reset, repair*—are significantly different from those in the rest of the dataset. She also finds that the cluster overlaps with all clusters in the categorical view (A1 in Figure 2.2) that have the above three terms as one of their main tags. The highest overlap in the categorical data view is with a cyan cluster tagged with *"alarm"*, *"reset"*, and *"fan"*.

She uses the Boolean operator to separate the intersection between these two clusters. From the category contribution panel (Figure 2.7), she notices that several types of equipment including *"RAF"* (Return Air Fan), *"EF"* (Exhaust Fan), *"VSD"* (Variable Speed Drive), *"CRU"* (Customer Replaceable Unit), and *"AHU"* (Air Handling Unit) contribute the most to this cluster. From her experience, she knows that the above equipment has always had relatively unreliable fans. Cross-checking with the numerical data panel, she realizes that the labor cost is relatively low for these problems, so she makes a note to have regular preventive maintenance done on the equipment.

Projection Interpretation. Now Alice decides to have the "too hot/cold" issue looked into further, and calls in an engineer to filter the dataset by these two tags and examine this filtered dataset separately. After loading the subset into the system, she notices a symmetry in the layout pattern of the text dimension view, about a horizontal axis. The clusters in the upper half of the projection all contain the keyword *"cold"* while those on the lower half contain *"hot"*. She infers that the vertical direction in the projected space relates to temperature, and becomes interested in the clusters located in the middle, especially the solitary purple cluster with tags *"proper, operation, verified"* (Figure 2.8-1). She notices a significant variation of labor cost and hours in this cluster (Figure 2.8-2). Selecting all points with a higher labor cost and hours, she learns from the updated keyword frequency plot that they correspond to the action *"replaced"*. From the category contribution panel, she finds that this part of the data is highly related to two supervisors *"S1"* and *"S8"* (Figure 2.8-3). She confirms this observation

by checking the tabular view (Figure 2.8-4). She believes that the high cost may either a clerical mistake, or an issue with the vendor supplying the parts. She decides to talk with these two supervisors to get to the bottom of the issue.

2.7 Expert Review

Our prototype was reviewed by three experts in machine maintenance analysis to determine its usefulness and throw light on the kinds of patterns or insights it might reveal to domain practitioners. The first expert (E1) was a data scientist from industry who had developed approaches for extracting actionable information from maintenance data for over six years. The second expert (E2) was an industrial engineer specializing in model-based systems engineering methodologies. Finally, the third expert (E3) was computer scientist from academia who worked on algorithm development and natural language processing for 25 years.

Informed by two pilot studies with our coauthors, we designed a semi-structured, open-ended expert review. We followed Elmqvist and Yi's "pair analytics" paradigm [98] with one of the authors as the experimenter and the domain expert as the participant. The study used a video conference setup where the experimenter controlled the tool while the expert remotely observed the visualizations and suggested filters, queries, and interactions via screen sharing. The experts perused a document explaining the views and functions of the prototype prior to the study, and were shown a 20-minute tutorial demonstrating the prototype at the start of the study. They were then asked to explore two datasets (30 mins each), one the HVAC dataset described in section 2.6, and the other a subset of 17,000 records from the HVAC dataset involving temperature-related complaints.

We categorize our observations on the domain experts' remarks during the exploratory tasks and their feedback on the prototype into *functionality* of the prototype, *visual encoding*, and *interaction*.

Functionality. The tutorial and demonstration at the start of the study involved use cases and obserations such as the one presented in section 2.6. At the end of the

demonstration, all three experts found our workflow to be *"highly reasonable"* (E2) and found the cases compelling. Yet, during the exploratory part of the study, they found it difficult to pin down the questions they could ask and answer of the data. For instance, E2 asked, *"What key question am I to answer?"* Based on the experts' questions about the visualizations, filters, and interactions during the exploratory study, we infer that the difficulty encountered by the experts was partly due to the relatively short time they spent with the interface, and their unfamiliarity with the data.

Visual Encoding. Domain experts found the linked views to be intuitive and useful. E3 remarked, *"I like the ways that the panels are automatically updated with respect to the selections that (are) made. And being able to see the three types of data all together is good. Definitely a good idea to have them combined"*. On the other hand, they found the notion of separating the data dimensions into categorical, numerical, and text to be too abstract. As E1 explained, *"Looking at categorical, text and numerical data makes sense from a data perspective, but it's not necessarily the functional break down that makes sense."* Instead, they reported that they would have preferred a way of representing the data that allowed them to see the problems in a functional way, e.g., where in the building, or where in the machine a problem occurred, or what temporal patterns were observable in the data. E1 and E3 also found it a little confusing that the default cluster labels in the numerical and categorical panels still used keywords from the text component of the data. On the other hand, while they were able to characterize at least one of the clusters, none of them re-labeled the cluster(s). All three experts also found the characterization view for the categorical data difficult to understand. E1 said that they had *"a really hard time understanding this visualization"*. E3 noted that they had *"never seen the information displayed in this way with two side by side panels of the cloud of individuals and categories... it's a little non-specific as far as whether the dots that show up in the (cloud of) categories are close enough to the dots in the (cloud of) individuals and how relavant it is."*

Interactions. All three experts found the brushing and linking to be highly useful, though the hexbin plots were a little confusing for E1 and E2, who took a highlighted hexbin in one view to indicate that all the points in that bin were linked to the cluster

selected in another view. E1 suggested providing *"a measurement of how much the correlation or lack of correlation is."* E3 initially found the Boolean operations to be less intuitive, but after asking for and seeing examples of how they were used, deemed the operations to be highly useful. Finally, E3 suggested the addition of numerical filters using which they could identify maintenance costs higher than a certain threshold, while E2 suggested filtering out data associated with commonly-occurring tags to help examine less-common problems.

Overall, the experts found the datatype-based separation less intuitive, but considered the coordinated views and Boolean operations across the views to be of value. They recommended more tangible ways of grounding the data in the domain familiar to them by using locations of machines in buildings, locations of components in machines, and filtering by cost, dates, and keywords.

2.8 Discussion

The use case scenario and the expert review illustrate the importance of interactive visual analysis in the maintenance workflow [44]. The use case scenario illustrates how the overview visualizations can provide useful groupings for analysts to explore and interpret using their domain knowledge. The expert review illustrates the usefulness of interaction and filtering in helping interpret unfamiliar visual abstractions, and highlights a need to ground the representations in a way that is familiar to the domain experts.

In the use case scenario, we saw how the overview visualizations can help identify common patterns across the dataset (e.g., the *"hot/cold"* cluster) and help small but closely related clusters stand out (e.g., maintenance of equipment involving fans). The ccMCA views (Figure 2.7) allowed the user to not only verify common traits—such as the presence of unreliable fans, or replacements ordered by a small subsets of supervisors—across a problem group, but also identify which equipment (in the case of fans) and supervisors (in the case of replacements) had the common traits.

The expert review highlighted both the advantages and disadvantages of our ap-

proach. When the domain experts were demonstrated scenarios such as that described in section 2.6, they were convinced and impressed by the capability of the prototype. Their validation of the workflow used to create the projected views and characterization views (Figure 2.1) also verified that our approach was well-motivated. On the other hand, the experts found the data separation and visualization too abstract to pick up in a single session. They preferred a more tangible means of viewing the data, based on location of the machines, locations of the components in the machines, and based on cost. However, representing high-dimensional data based on only one or two characteristics may not reveal important insights. In addition, one of the main advantages of our approach—its generalizability to other domains—will be lost by grounding it too much in one domain. However, there may be a middle ground wherein the user is able to add an additional "custom" view based on familiar data characteristics. We will explore this in future iterations.

In spite of the difficulty the experts faced with the abstract representations, they found the coordination or linking across views to be a useful feature that helped them understand the data better. As with the representations, they did express a preference for more tangible filters (e.g., based on specific cost ranges). However, at least one expert (E3) had started to appreciate the sophisticated filtering possible through the coordinated views and Boolean operations. The expert feedback suggested that some of what they found difficult about the interface was more due to the short duration of the sessions rather than the data abstractions themselves. A longitudinal study—though unrealistic at the this time with restrictions on data sharing and the current constraint of remote sessions—would help address some of the familiarity issues that the domain experts currently faced.

2.9 Summary

In this work, I present a design effectively coupling machine learning with interactive visualization for analyzing large, heterogeneous, multidimensional maintenance log data. A key approach is to separate the dataset's dimensions based on whether the

data falls under numerical, categorical, or text data, and use lower-dimensional, clustered views that reveals groups in the dataset by each dimension type. I apply existing techniques such as ccPCA and word embeddings with frequency plots to characterize the dataset based on its numerical and text dimensions. Notably, a unique capability is provided with our new contrastive learning method, ccMCA, to characterize a dataset with its categorical dimensions. I present these approaches to clustering and characterization in the form of a dashboard with linked views, and illustrate its utility through a use-case scenario and an expert review. In particular, the scenarios allow us to highlight the power of ccMCA in identifying categorical dimensions and their values that contribute to a cluster, while the expert review highlighted the usefulness of linked views to characterize clusters across different dimension types. I also identify the need for more grounded, domain-specific representations of data to scaffold the experts' understanding of the system. I will continue the discussion on the application scenario of machine maintenance log analysis and demonstrate how experts' knowledge can be leveraged to enable efficient technical text annotation in Chapter 6 .

Chapter 3
Knowledge Presentation with Numerical Data Fact

Infographics, which embeds numerical data in visual embellishments, is an effective visual representation to convey information and impress the audience. Existing tools are capable of but not sufficient enough in the creation of professional infographics. As a result, such tools are not attractive enough to those who are not equipped with the design expertise to create a professional infographic and find it difficult to go through the learning curve of the tools. In this chapter, I introduce a *knowledge presentation* approach that automatically generates infographics from natural language statements. I contributed to the published version [79] of this work as the second author when I interned at Microsoft Research Asia (MSRA). I conducted a preliminary study to explore the design space of infographics as well as participated in the building of a proof-of-concept system that automatically converts statements about simple proportion-related statistics to a set of infographics with pre-designed styles. I will provide more details about my contribution while briefly mention the work accomplished by my colleagues, such as part of the technical details and the evaluation of usability and usefulness of the system through the generated infographics and the expert reviews.

3.1 Introduction

Information graphics (a.k.a. infographics) is a type of data visualization that combines artistic elements (e.g. clip arts and images) with data-driven content (e.g. bar graphs and pie charts) to deliver information in an engaging and memorable manner [147]. Due to these advantages, they are widely used in many areas, such as business, finance, and health-care, for advertisements and communications. However, creating a professional

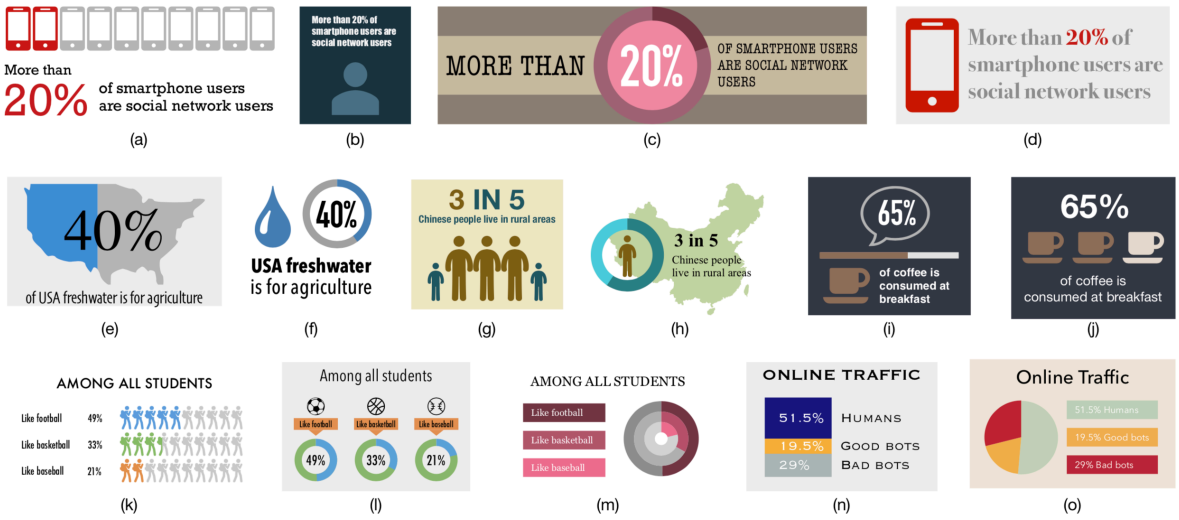


Figure 3.1: Example infographics generated by Text-to-Viz. (a)-(d) are generated from the statement:"More than 20% of smartphone users are social network users." (e) and (f) are generated from the statement: "40 percent of USA freshwater is for agriculture." (g) and (h) are generated from the statement: "3 in 5 Chinese people live in rural areas." (i) and (j) are generated from the statement: "65% of coffee is consumed at breakfast." (k)-(m) are generated from the statement: "Among all students, 49% like football, 32% like basketball, and 21% like baseball." (n) and (o) are generated from the statement: "Humans made 51.5% of online traffic, while good bots made 9.5% and bad bots made 29%."

infographic is not an easy task. It is a time-consuming process and also often requires designer skills to ensure the perceptual effectiveness and aesthetics.

Much research has been devoted to investigating the design aspect of infographics [33, 162, 328] and developing authoring tools [184, 309, 374] to facilitate the creation of data-driven infographics. Based on different considerations, these authoring tools all strive to reach a balance between the ease-of-use and the power of features, and then to speed up the authoring process. However, these tools generally target advanced users. With complicated editing operations and technical concepts, these tools are not friendly to casual users, who, we believe, form another major category of infographic creators, other than professionals, such as graphic designers and data scientists [220].

Consider a hypothetical example in which a program manager, Nina, is preparing a presentation for her manager and wants to emphasize in her slides that "40% of US kids like video games." She decides to add an infographic next to the statement with an authoring tool, e.g., DDG [184] or InfoNice [374]. Since Nina is not a professional graphic designer, she first needs to spend time (re-)familiarizing herself with the tool,

such as the user interface and work flow. However, even if she were familiar with the tool, she still may not know where to begin to create a desired infographic, because all the existing authoring tools assume that the users have a clear or rough idea of what the final infographic may look like. Unfortunately, Nina has no design expertise and has little to no knowledge of how a professional infographic would look like. Therefore, she likely needs to go through existing well-designed infographics (in books or on the Internet) to look for inspiration. Based on the examined samples, she then settles on a design choice that has the best "return of investment" in terms of authoring time and her purpose of emphasizing the message.

From this example, we can summarize some common patterns for this user category. First, creators in this category only occasionally create infographics and thus are not proficient in the authoring tools. Second, they do not aim for the most creative/impressive infographics, which often involve complex designs and a long authoring time and are unnecessary in terms of "return of investment". Instead, something effective but professional is often sufficient for their purposes. Third, they often only have little design expertise, and would likely be unclear on how to design a decent infographic from scratch. On the other hand, if they were provided with good and relevant samples, they could often quickly pick one based on their personal preferences.

To address the needs of users in this category, we explored a new approach to automatically generate infographics based on text statements. Since text is the most common form of communication to exchange information, we believe that this approach can help more people take advantage of infographics. To achieve this goal, there are two major challenges to overcome. The first one is to understand and extract appropriate information from a given statement. The second one is to construct professional infographics based on the extracted information. For the text understanding challenge, we first collected a corpus of real-world samples. Then, these samples were manually labeled to train a CRF-based model to identify and extract information for infographic constructions. For the infographic construction challenge, we analyzed and explored the design space of infographic exemplars that we collected from the Internet. Based on

the design space, we proposed a framework to systematically synthesize infographics.

Considering the numerous types of information that can be represented by infographics [283, 327], and the numerous ways to express the same information textually and visually, it is impossible to cover the entire space in one paper. Instead, we decided to focus on a relatively small and isolated text-infographic space and build a proof-of-concept system for it. To achieve this goal, we first conducted a preliminary survey on the existing infographics to identify a category of information that is commonly represented by infographics and also has clear textual and visual patterns to process systematically. Based on the preliminary survey, we chose a subtype of information related to proportion facts (e.g., "Less than 1% of US men know how to tie a bow tie.") and built an end-to-end system to automatically convert simple statements containing this type of information to a set of infographics with pre-designed styles. Finally, we demonstrate the usability and usefulness of the system.

3.2 Preliminary Study

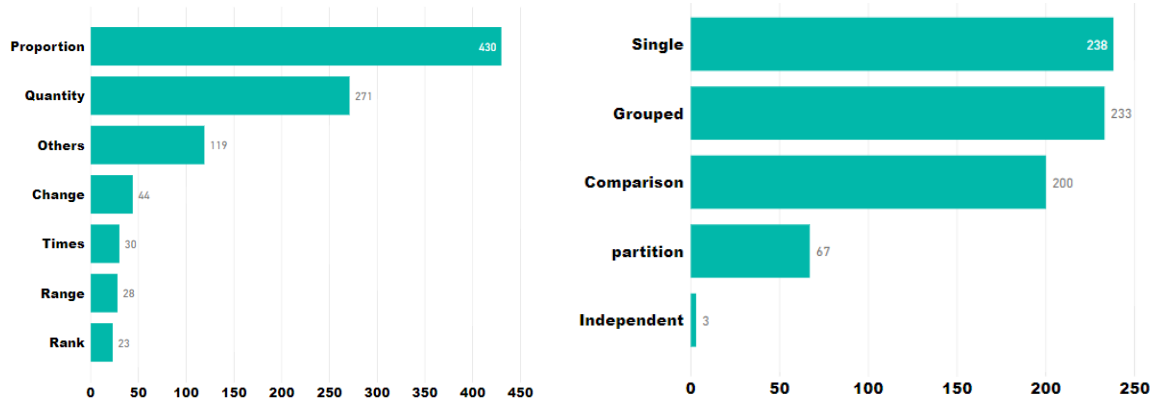
3.2.1 Infographics Collection

The goal of our study is to look into the ways of numerical facts visualization, so a representative corpus of numerical facts as well as the corresponding visualization output are needed. In actual practice, we combine the collection process of these two groups to ensure every numerical fact has proper visualization result. First, we gather infographics with numerical facts and build the visualization item dataset. Then we extract the numerical facts from these figures to build the numerical fact dataset.

As a pre-processing step, we split infographics to their minimum unit. In general, infographics do not appear alone but are tiled in a larger poster to present multiple numerical facts. We collect over 100 posters from a wide range of sources, e.g., *Visual.ly* and *Visualnews.com* and manually separate them into 889 valid infographics. An infographic identified as "valid" fulfills the following four criteria: 1) it contains at least one numerical fact; 2) it contains at least one visual element; 3) it is visually and semantically complete; 4) it cannot be divided into smaller subset fulfilling the first three criteria.

3.2.2 Characterizing Numerical DataFacts

The choice of visualization idiom may vary greatly for different types of numerical facts. So it is necessary to declare the classification of the numerical facts before looking into their visual representation. Based on our survey of the 889 subjects, we divide the numerical facts along two orthogonal dimensions: *function* and *multiplicity*. Figure 3.2 shows the statistical distribution of the them in each dimension. In this section, we define different types of numerical facts within each dimension, along with presenting their visual form and supporting examples from our survey.



(a) Function (b) Multiplicity

Figure 3.2: Categories of Numerical Data Fact

3.2.2.1 Dimension 1: Function

The function of a numerical fact is decided by what information the sentence conveys to the audience [293]. Generally, it will provide basic instruction on which visual encoding form to choose when we visualizing the numerical facts. We have identified eight types of numerical facts according to their function:

Proportion This kind of numerical fact contains statistical information about how much a part occupies the whole. Samples that fall into this category typically contain numerical facts in three forms:

1. percentage: $x\%$

   e.g. "Less than **1%** of US men know how to tie a bow tie [282]"

2. ratio: x_1 in x_2, x_1 out of x_2

   e.g. "In some emerging economies, **3 in 10** youths cannot do basic arithmetic [307]"

3. fraction: x_1/x_2

   e.g. "More than **1/3** of the U.S. adult population is obese [40]"

The numerical fact of this type describes the distribution of the data in a concise but power way, thus is favored by data scientists. According to our survey, it proves to be the most common used type: 411 (68.73%) of the 889 numerical facts in our dataset contain proportion. Because of its prevalence, there are many visual elements specially designed for the representation of *proportion*, like pie chart and donut chart. We will further discuss the visual representation of this type in next section.

3.2.2.2 Quantity

This kind of numerical fact simply contains an absolute value about the amount of the entity, i.e. price, population, duration, etc. It is the second-most populated type in our dataset: 247 (41.3%) out of the 889 samples. For *quantity*, the visualization strategy may vary greatly depending on the feature of target object and the magnitude of the value. However, there is one common point for its visualization – since *quantity* is used to provide a straightforward knowledge about the amount, the value in the sentence is often highlighted with extra visual forms.

Change This kind of numerical fact typically describes the entity's change in value over a period of time. Samples in this category either employ words like "increase", "decrease", "drop" to explicitly show change, or list different value in different time point to implicitly show change. It is worth mention that numerical fact of this type may be confused with *proportion* or *quantity* sometimes, because it also employs expressions like percentage or absolute value. However, *change* has richer semantics than the previous two. And an effective to distinguish them is to check whether the sentence conveys any information about transform. For example, "Crop yield could **drop** 30% by the end of the century [135]" and "**In 2012**, 75 million U.S. adults used a mobile phone as a healthcare tool. **In 2013**, this number **increased** to 95 million [393]" are both numerical facts that should be grouped into *change*. Visual representation for this kind often

involves specific icon like arrow or line/bar chart to show difference over time.

Times This kind of numerical fact indicates the multiple relationship between two entities. It is easy to identify literally with comparative adjectives or expressions like "more than","more likely", "ax times", etc. Typical sentences belong to this category include "Posts with photos generate up to **180% more** engagement **than** those without [1]" and "Grad students borrow **3x more** per year **than** undergrads [187]". As for visual expression, it often takes use of the comparison of size, length or amount of icons to present the multiple relationship.

Range This kind of numerical fact does not provide a definite value but a value set with borders. It has similar semantic to "interval" in mathematics. Therefore, sentence of this type generally contains expression like "x_1 to x_2", "x_1-x_2" in analogy with bounded interval $[x_1, x_2]$, or "$< x_1$", "$> x_2$" in analogy with unbounded interval $(-\infty, x_1]$, $[x_2, +\infty)$. A simple example of this kind is "Users often leave web pages in **10-20** seconds [235]". In our infographics dataset, some numerical facts of *range* is presented with a bounded area in bar chart or pie chart. But a more common type of numerical facts of *range* is time point interval (e.g. 2012-2013). It often acts as a supplementary information of other numerical facts and is not independently visualized.

Rank This kind of numerical fact reflects the relative position of the entity in a group. It is also easy to identify with ordinal numeral (e.g. "Florida has **3rd** largest homeless population [179]) or specific symbols like "#", "No." or "Top n" (e.g. "Lack of money is the **#1** reason why adults in America do not receive proper mental health services [262]). The visualization form of single *rank* and multiple *rank* is different. For multiple *rank* data, the icons of entities are often arranged as an ordered list according to the rank value. However, the single *rank* data is often not specially visualized.

Position This kind of numerical fact is the spatial data providing a location in two-dimensional (2D) or three-dimensional (3D) space, i.e. a latitude-longitude pair specifying the location on Earth or a three dimensional coordinate describing the location within a medical CT [251]. It often appears with map or figures in professional field, but it is quite rare in our infographics dataset. 5

Table 3.1: Category of Multiplicity

Category	Subjects(**S**)	Facets(**F**)
Single	1	1
Grouped	1	n
Comparison	n	1
Composition	$n \rightarrow 1$	1
Independent	n	n

Others There are a large quantity of numerical facts that does not have clear statistical meaning, thus do not belong to the seven categories discussed above. They actually act as a symbol instead of value in the sentence, i.e. date, age, serial number, phone number, etc. In corresponding infographics, they often appear on title, legend or x-axis of a chart to distinguish its corresponding subject with others in the same context.

3.2.2.3 Dimension 2: Multiplicity

Apart from *function*, the choice of visualization idiom is also influenced by how many numerical facts to cover and their relationship. Based on this fact, we define *multiplicity* according to the number of subjects(**S**) and corresponding facets(**F**) described by all the numerical facts in current context. In this section, "subjects(**S**)" refers to which person or object the numerical fact describes, and "facets(**F**)" refers to which type of property it quantifies. As shown in table 3.1, we define five types of *multiplicity*:

Single (1S1F) There is only one numerical fact in the sentence, it describes one facet of the only subject. This is the simplest but the most common (31.94%) *multiplicity* form in our dataset. There are no mutual relationship between numerical facts, so the choice of visualization form of this type are generally decided by its *function* (Dimension 1).

Grouped(1SnF) In this category, the numerical facts depicts more than one facets of the common subject to form a complete picture of it. For example, in the sentence "In the United States alone, there were 10.5 billion searches in July 2009, which is a 106% increase from 5.1 billion in Dec. 2005 [297]", all of the numbers are used to provide information about "searches in the United States". This type of numerical fact is also

very common in our dataset (31.77%). Since there is a common subject, icons about this subject are often used to construct the main body of the infographic. The closer a numerical fact relates to the subject, the higher priority it has to influence the type of the final visualization idiom.

Comparison(nS1F) For this type, multiple numerical facts are provided to compare the same facet of different subjects. Taken "70% of members want to bank online, while 66% of members want to walk in branch [329]" as an example, "66%" and "70%" compare the "online bank" and "walk-in branch" in terms of number of members. The role of each numerical fact in the context is the same. Therefore, they are either combined into a bar chart or placed in parallel with same type of visual elements. When visual elements are placed in parallel, distinguished color, size or icon are usually used to show difference and achieve comparison.

Composition(n→1S1F) This type of numerical facts also depicts one single facet for different subjects. But these subjects have inner connection and can be combined to form a larger whole. That is why we name it "n→1S1F" in the title. In another word, one subject has been partitioned to smaller subjects according to one specific feature, which can be reflected by the numerical facts. This is also the main difference between this type and *comparison*. A typical example for *composition* is "60% of the population are visual learners, 30% are auditory, 10% are kinesthetic". If represented in proportion, the numerical facts of this kind generally add up to 100%, which makes it easier to distinguish from *comparison*. And because of this special feature, data scientists prefer to combine all the data into one pie chart, donut chart or stacked bar chart, instead of visualize each numerical fact separately.

Independent(nSnF) For this type, there is no direct relationship between numerical facts (but they are usually about the same topic). So they may depict any facet of any subject. This situation is quite rare and only appears 3 times in our dataset. Because of the content independence, the choice of visual idiom is quite free. Generally the most important information has the highest priority to be visualized.

3.3 Proportion-Related Information

We provide general instruction on how to choose visual idioms for each category of numerical facts in last section. In this section, we conduct detailed analysis on the visualization strategies for numerical facts of proportion. We select this type because of its popularity and abundant samples. According to our survey, *proportion* make up almost 70 percent of the numerical facts in view of dimension 1. Generally, the two main components of infographics are description and image. In the following discussion, we first summarize the layout which defines how to arrange description and image spatially. Then we report our discoveries about these two kinds of components for *proportion* respectively. Finally, we provide some suggestion about color theme choice for designers.

3.3.1 Layout

The components of infographics are organized in different style depending on the character of numerical facts, resulting in a layout. It provides general scheme for spatial arrangement of description and image in current infographic. There are some existing description about layout types in previous work [312] [17], though none of them specially discuss layout for infographics of numerical fact. Based on the taxonomies raised in these papers, we discuss the layout for infographics of single proportional numerical facts first. Then we also provide two strategies to construct infographics for multiple proportional numerical facts based on the result of single infographics.

3.3.1.1 Layout for Single Proportion

Single proportional numerical facts are the simplest while the most common type in *proportion*. We choose to discuss the infographics layout for this type first, because they act as the building blocks for other types of *proportion* according to our survey. In order to provide a more practical instruction on the implementation of the visualization system, we divide the infographics of single proportional numerical facts in our dataset into four groups according to the number of their two main components – description and image. Specifically, they are "1 description - 1 image", "2 descriptions - 1 image",

"1 description - 2 image" and "2 descriptions - 2 images". For each group, we observe that there are some general layout styles which are applicable to all the numerical facts in this group. Besides, if the numerical facts contain specific information and can be represented as maps, statistical charts or container pictures, there are also some special layout styles for them.

The *general* layout arranges description or image in horizontal, vertical or tiled panels. The panels have clear margins between each other, but they do not have to be not equally-sized. There is no specific requirement for the content of numerical facts. So this kind of layout is applicable for most infographics of numerical facts. In fact, it is the most common type of layout in our dataset. However, this kind of layout cannot present any special relationship about numerical facts and less aesthetic to some extent.

If the numerical fact contains some *geographic* information, it can be shown in a map with notation. Generally, the center of this kind of layout is the map, with a notation related to a certain location on it. The content of the notation could either be a paragraph of description or a statistical chart about the numerical fact. If there is additional information like title or supplementary instruction, an additional panel of description can also be placed above or below the map.

If the numerical fact can be represented as a *statistical chart*, the layout can be arranged as the chart with notation. In this case, the notation is usually a piece of description and related to a certain point on the chart. Additional description can act as the title or legend of the chart.

If the numerical fact can be matched to some special icons that are able to act as *container*, the other images or description can be placed inside the blanks of the *container*. On this occasion, the distribution of the blanks actually decides the layout of the whole infographic, including the number, size and position of other panels. Although the final result is quite impressive for this kind of layout, the matching process is quite challenging and may require manual work.

3.3.1.2 Layout for Multiple Proportion

Infographics layout for multiple proportional numerical facts can be built upon the layout of single proportional ones. This is based on our observation that each of the numerical fact in the sentences with multiple proportional numerical facts can be represented as a single infographic with the same layout style. So these numerical facts can be visualized separately first. Then their infographics can be combined as a whole according to some rules. According to our survey, there are two common strategies to achieve the combination process.

The first strategy is to merge the single infographics by sharing some of their common elements. This method keeps the original layout of the single infographics while shows multiple data in virtue of some mathematic or design inspiration. It can be further divided into four types according to which visual element they share:

1. **Share axes**. This type aligns the visual marks of all single infographics so that they can be placed in a common coordinate system. It is applicable for statistical charts with axes, like bar chart or scatterplots. However, if the measurement of numerical facts are incomparable or their range of value varies too much, this type may lose effectiveness because of the failure of sharing axes.
2. **Share center**. This type arranges multiple numerical facts in form of concentric circles or sectors with the same center. It is applicable for circular charts with center, like pie chart or Nightingale rose chart. The main limitation for this method is the scalability. If there are too many numerical facts, the concentric circle will become extremely big or the sectors will become hard to distinguish.
3. **Share illustration**. This type links multiple annotations to different position of the shared illustrative image. It is applicable for annotated diagrams with the same illustration, like map or anatomy. This method requires the original sentence to contain clear location information, which limits the feasibility of it in most cases.
4. **Share container**. This type is applicable for infographics sharing the common background. At the same time, this background acts as the container for all the descriptions of the infographics. In another word, there are some blank space

in the background image to place the descriptions from different infographics. So the main challenge of this method is to find ideal background image with appropriate number and size of banks.

The second strategy is to keep the single infographics separated and arrange them in a larger canvas according to some rules. For this type, the original layout of the single infographics are nested in a larger layout. And this strategy can be further divided according to the feature of this large layout:

1. **Parallel**. This is the most straight-forward layout strategy. If all the numerical fact are equipped with the same visual elements, they can be easily positioned in parallel, either vertically or horizontally. Because of the similar visual assets, more attention of the audience is attracted to the difference among the content of numerical facts. This kind of layout is effective when the expected effect is to show discrepancy between several numerical facts.
2. **Tiled**. The single infographics are arranged in tiled panels. Each panel has clear border and only contains one single infographic. There might be only one row or one column of panels, and they may be not equally-sized.
3. **Circle**. Generally there is a panel in the center of the canvas. And the single infographics are arranged around this panel to form a circle. This kind of layout provides an extra space to show the common information of the multiple numerical facts, which makes the single infographics more closely related.
4. **Hierarchical**. The single infographics are arranged in a tree structure to present the hierarchical relationship between them. The expected reading order for this type is generally to follow the expanding or collapsing of the tree. A common scenario for using this layout is to show hierarchical information about different part of the main body.
5. **Stacked**. The single infographics are not regularly arranged but well organized to minimize the blank space of the canvas. For example, description may surround the image according to the outline of the icon, and the text direction also varies to fit current space. This kind of layout is also suitable for most cases. But finding

the best layout for given text and image is not an easy job and may need extra optimization algorithm.

3.3.2 Description

We have mentioned above that description is one of the two main types of components in the infographics of numerical fact. The direct source of the description is the original numerical facts. In this section, we will discuss several most common types of description and their relationship with the original sentences. Besides, we consider "value" as a special type of description and have additional discussion about it, because it is generally presented with distinguished visual effect.

3.3.2.1 Single Proportion

The original sentences(**S**) with numerical fact of single proportion generally contains some specific clauses or phrases, which are the basic building blocks of the description. Syntactically, a complete sentence should contain subject(**SB**) and verb-object phrase (**VOP**). Semantically, the sentence provides information about value(**V**), part(**P**) and whole(**W**) of the proportion. Based on the position of value, the sentence can also be trimmed to the sentence without value $\{\mathbf{S}-\mathbf{V}\}$ or part before value (**PBV**) and part after value (**PAV**).

With above representation, we list five most common types of description as follows:

1. **S**
2. **V** + $\{\mathbf{S}-\mathbf{V}\}$
3. **V** + **VOP**
4. $\{\mathbf{V}\_of\_\mathbf{W}\}$ + **P**
5. **PBV** + **V** + **PAV**

where "+" splits different components of the description. One complete description may has 1 to 3 separate components. The choice of description should be decided according to the actual layout, like how many components and how much space for the description.

3.3.2.2 Multiple Proportion

For description of multiple proportional numerical facts, there are two possible strategies. The first strategy is to separate the original sentence to a group of single proportional numerical facts. Then the description can be extracted according to the rules for single proportional numerical facts. For example, the sentence "3 out of 4 U.S. farmworkers earn less than $10,000 annually, and 3 out of 5 live below the federal poverty line" can be separated to "3 out of 4 U.S. farmworkers earn less than $10,000 annually" and "3 out of 5 live below the federal poverty line". Each of them contains the information to generate the five types of description for single numerical fact. This strategy requires relatively independent structure and complete information for each sub-sentence. So it is more suitable for the type *group* or *independent*.

The second strategy, in contrast, is to view the original sentence as a whole. It presents some closely related numerical facts that can be packed into certain charts like donut chart or bar chart. Generally there is a common-part for all the proportional facts and multiple counterparts for each value. The common-part, counterparts and their corresponding value are the description for this type. For instance, in the sentence "Only 4 percent of elementary schools, 8 percent of middle schools and 2 percent of high schools provide daily physical education for all students", the common-part is "provide daily physical education for all students", and the counterpart-value pairs are "elementary schools-4 percent", "middle schools-8percent" and "high schools-2 percent". This strategy requires the distinguished numerical facts to depict a common facet of their subject, so it is more suitable for the type *comparison* or *composition*.

3.3.3 Image

Image is another basic component of the infographics for single proportional numerical fact. In this section, we discuss the type of visual encoding idioms and corresponding application scenario in the case of single proportion. We also report our discovery about the icon selection process for some types of infographics.

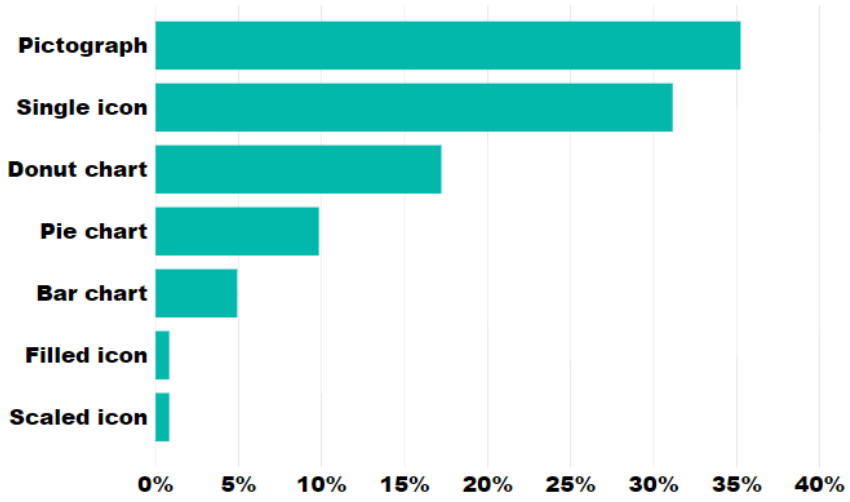


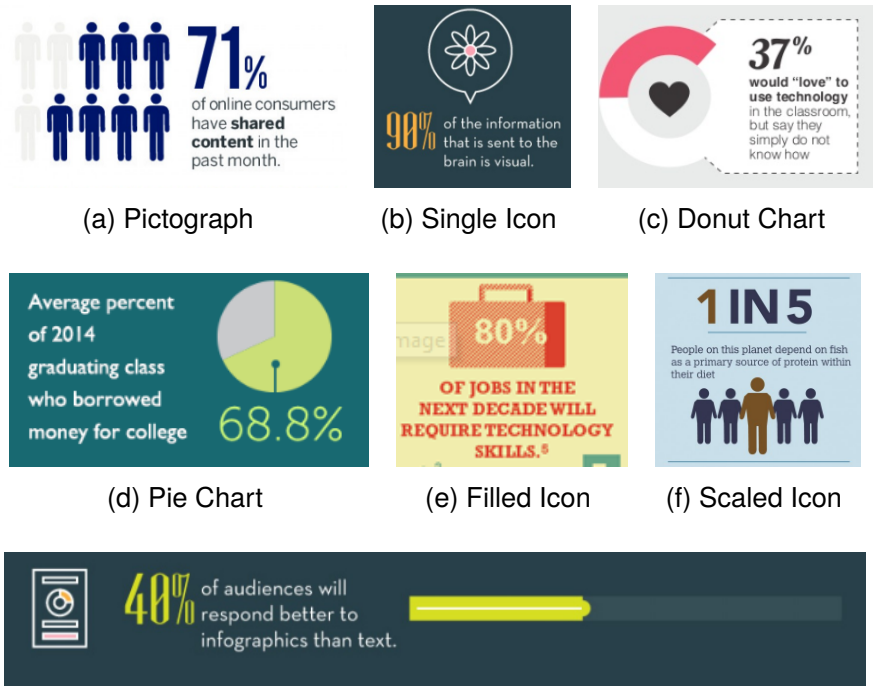
Figure 3.3: Percentage of Image Types

3.3.3.1 Type

Based some existing visualization taxonomies, we go through our dataset and summarize frequently-used image types. Although there are over 30 potential visualization idioms available according to [33], suitable ones for single proportional numerical facts can be narrowed to seven types, which can be covered by a smaller taxonomy in [10]. Figure 3.3 shows the statistical result of their appearance rate.

Pictograph is the most frequently-used types of image (Figure 3.4(a)). It is a group of duplicate content-related icons. The content is usually reflected in two aspects: 1) the vision of icon reflects the keywords of sentence; 2) the number of icons or their color-fill status reflects the value in numerical fact. Pictograph is suitable for numerical facts in which the number of icons can be easily found and is no more than 10. For example, most of proportion that is in the form of ratio, like "x_1 out of x_2", is represented in pictograph. And for the data in the form of proportion as well as divisible by 10 or 5, pictograph is also quite common.

Single icon is a content-related picture (Figure 3.4(b)). Since the single picture can only reflect keywords but not value of the numerical facts, it is generally more colorful and informative than the icons in pictograph, covering more than one keyword sometimes. In order to compensate for the lack of value visualization, the "value" in description usually has extra highlight with strategies mentioned in last section. This



(a) Pictograph (b) Single Icon (c) Donut Chart

(d) Pie Chart (e) Filled Icon (f) Scaled Icon

(g) Bar Chart

Figure 3.4: Examples for Different Image Types

type of image is applicable in most of the cases so long as there is a suitable icon. But because of the limited information it conveys, it is often chosen when the original numerical fact is not very important or the space is limited.

Traditional statistical chart is the third commonly used visualization idiom. For single proportion, donut chart (Figure 3.4(c)), pie chart (Figure 3.4(d)) and bar chart (Figure 3.4(g)) is more suitable. Although donut chart is actually a variant of the pie chart, we treat them separately. That is because there is a circular space in the middle of the donut chart to put extra image or description, which makes donut chart more informative than pie chart. These three types of charts are also suitable for most single proportional numerical facts. But in view of their shape, pie chart or donut chart is preferred when the prospective space for image is square, while bar chart is preferred when the space is linear.

There are several other types of images like filled icon (Figure 3.4(e)) or scaled icon (Figure 3.4(f)). Filled icon reflects the value of numerical fact with the percentage it is filled and scaled icon reflects that with its size. Both of them reflects the keywords with

the vision of icon. They require the icon to have attributes like "fillable" or "scalable", and the filling direction is also dependent on the meaning of the icon. Because of these constraints, they are not that commonly used in the infographics for single proportion.

3.3.3.2 Icon

Icon is the basic graphic primitives in pictograph, single icon, filled icon and scaled icon. The keywords of the original sentence and the space reserved for image are the two main considerations for icon selection. Firstly, icon should be closely related to the content of the original sentence. The more keywords it covers, the better it fits current case. It is possible that icon does not exactly match the keyword, but is a concretization, subset or close neighbor of it. For example, for the keyword "video games", the logo of Xbox is also a good choice. Secondly, the shape of the icon decided whether it is suitable for current layout. Different stretch direction of icon and reserved space can result in big blanks and loose layout, which greatly reduces the aesthetic of final infographics. Besides, additional attributes like "fillable" or "scalable" should also be taken into account for filled icon and scaled icon as mentioned before.

3.3.4 Theme

Color is a complex topic and influenced by many factors. In this section, we concentrate on the palette choice strategies and how to distribute colors in the palette to different components of the infographic.

3.3.4.1 Palette

The choice of palette for infographics is mainly influenced by the semantic topic of numerical facts apart from aesthetic. We notice that there are already many websites (e.g. [4] [77] [62]) providing color schemes that fulfill the aesthetic requirement. Most of them provide palette options with 3-5 types of suggested color and a brief description of application scenarios. Generally, the topic of numerical facts matches the application scenarios of the palette. For example, if the numerical facts is about environment protection, it is highly possible that its palette has description like "natural and earthy" and contains color like green and blue. However, the number of the palettes is limited

and the palette description is often quite ambiguous in order to cover more situation. So it is possible that the numerical facts cannot exactly match a palette, or cannot match any palette at all. Therefore, the content of the numerical facts just influences the choice of color scheme necessarily but not sufficiently.

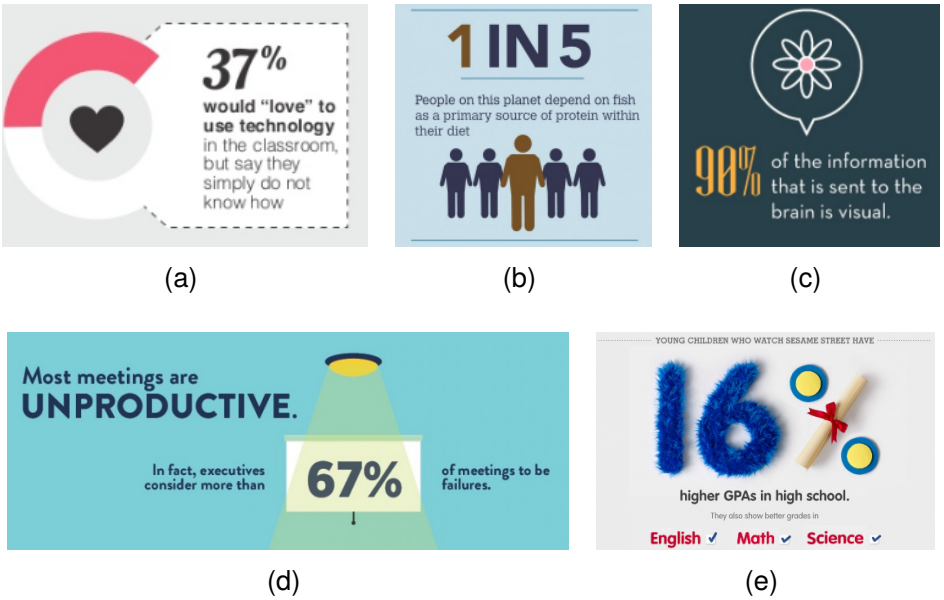
3.3.4.2 Color Distribution

Based on our discussion about components of infographics, we can further divide them into several parts in the view of color distribution, including "title", "text", "value", "background", "empty" and "fill". Among these parts, "title", "text" and "value" are corresponding to the component "description". The "background" defines the color of background. As for "empty" and "fill", they are used to color foreground color and background color of the component "image" if the type belongs to pictograph or chart. However, it is also possible that the original icon already contains color information, especially for single icon. On this occasion, the original color is preferred and the color in the palette is discarded.

The color in the palette is assigned to different parts according to some common rules. Generally, there is a main color in the palette which is the most closely related to the description of this palette. It is often assigned to "background", because "background" often occupies the biggest area of the whole infographic. The color that contrasts the most to "background" is assigned to parts belonging to "description" in order to help them stand out from the "background". The "text", "title" and "value" do not have to use completely different color, but the color of "title" and "value" is often different from "text" to achieve highlighting and echoing. If "empty" and "fill" for "image" is in use, they should show relatively bigger contrast. Meanwhile, "empty" often belong to the same tone with "background". If the original color of icon is used, their color should also in harmonious with the ones in the palette. Besides, only two kinds of color are not enough for "image" sometimes, like some complex charts for multiple proportion. In this case, the variants of the original "empty" and "fill" with different lightness or saturation are the most ideal compensaton.

3.3.4.3 Value Highlight

Value is the quantitative representation of the numerical fact and often requires extra emphasis. Most of the samples in our dataset highlight value with distinguish size, color and font (Figure 3.5(a)(b)(c)). A second way to highlight value is to embellish it with background picture (Figure 3.5(d)) or turn itself into wordart (Figure 3.5(e)). Besides, value is often placed at the optical center like middle or top part of the picture to draw more attention from audience.



(a) (b) (c)

(d) (e)

Figure 3.5: Value can be highlighted with distinguished size, color, font or additional embellishment.

3.4 Implementation and Evaluation

The prototype system contains two main modules, namely text analyzer and visual generator. First, users provide a textual statement about a proportion fact, such as "More than 40% of students like football." Then, our text analyzer identifies the essential segments in the statement, including modifier, whole, part, number, and others. Then the original statement and the extracted segments are fed into the visual generator for infographic construction. For each dimension (i.e., layout, description, graphic, and color), a set of visual elements are generated or selected. Then, we enumerate all combinations of these elements, to synthesize valid infographic candidates. Finally, all

the synthesized results are evaluated and ranked, and the ones with high scores are recommended to users. Then, users can either directly export any of them as an image to integrate into their reports or presentation slides, or select one and further refine it based on their personal preferences.

To demonstrate the diverse designs that our system can automatically create, we present a variety of infographics created with our system (Figure 3.1). For example, Figure 3.1(a)-(d) and Figure 8(b) are all generated based on the same statement, "More than 20% of smartphone users are social network users." We can see that different templates can produce different infographics. In particular, we can see that Figure 8(b) is based on the layout blueprint illustrated in Figure 8(a). However, since the template does not reserve a place for the modifier component, the generated infographic is less accurate than the others, and hence has a lower informative score. Figure 3.1(e) shows an example of a filled icon, while Figure 3.1(f) shows an example of a tilted layout. Figure 3.1(g) and (h) demonstrate two examples of how our system handles proportion information in the form of "m in n". Our system can either choose the correct number of icons to form a pictograph (Figure 3.1(g)) or convert the information to a percentage number and show it with other visualizations (Figure 3.1(h)). Figure 3.1(i) and (j) show that our color strategy can correctly select colors based on semantic information. Since one of our color palettes has the descriptive keyword coffee, this color palette will be ranked higher when choosing colors for infographics. Figure 3.1(k)-(o) demonstrate the results for showing multiple percentages. Specifically, Figure 3.1(k)-(m) show a comparison case, in which proportion facts cannot be logically accumulated, while Figure 3.1(n) and (o) show an accumulation case. To understand how general users perceive our system, we further conducted a set of casual interviews with a wide variety of audiences in two exhibit events. The last assessment involved expert evaluation with three professional graphic designers.

Given I did not primarily contribute to the technical innovation and interview procedures of the prototype system, I skip further details related to these two parts of work. Readers who want further reading could refer to our publication [79].

3.5 Summary

The use of infographics to present numerical facts enables rapid and clear communication of underlying knowledge, without requiring extensive background knowledge from the audience. In this chapter, I present a preliminary study that systematically analyzes the layout, components (description and image), and theme for conveying numerical facts of proportions. Based on the study, I introduce a framework for automatically generating infographics and demonstrate its feasibility through a proof-of-concept system. The example results and interviews with users/designers highlight the tool's usability and its potential for widespread adoption in everyday life. In the future, I aim to expand this work to support a broader range of statistical information and explore other types of infographics, such as timelines and locations. With advancements in machine learning-based techniques, it is also promising to enhance the system and infographic quality through data-driven approaches.

Chapter 4
Knowledge Presentation with Unstructured Text Data

Current text visualization techniques typically provide overviews of document content and structure using intrinsic properties such as term frequencies, co-occurrences, and sentence structures. Such visualizations lack conceptual overviews incorporating domain-relevant knowledge, needed when examining documents such as research articles or technical reports. To address this shortcoming, I present ConceptScope, a new technique and system that utilizes a domain ontology to represent the conceptual relationships in a document with a Bubble Treemap visualization. Multiple coordinated views of document structure and concept hierarchy with text overviews further aid document analysis. ConceptScopefacilitates exploration and comparison of single and multiple documents respectively. I demonstrate ConceptScopeby visualizing research articles and transcripts of technical presentations in computer science. In a comparative study with DocuBurst, a popular document visualization tool, ConceptScopewas found to be more informative in exploring and comparing domain-specific documents, but less so when it came to documents that spanned multiple disciplines.

4.1 Introduction

Text visualization techniques have evolved as a response to the virtual explosion of text data available online in the last few decades. Specifically, they aim to provide a visual overview—what digital humanities now call "distant reading" [248]—of large documents or large collections of documents, and help the researcher, investigator, or analyst find text patterns within and between documents (e.g. [337]). Most of these visualization techniques are domain-independent, and do not provide a knowledge-based

overview of documents. There have been approaches to provide a visual overview of the semantic content of documents (e.g. [73]). Such approaches have typically looked to lexical hypernymy (is-a relationships) to provide a conceptual overview of the text.

However, when examining domain-specific documents such as research papers, medical reports, or legal documents, it is necessary to examine the documents from the point of view of that specific domain. For instance, when examining a research paper in computer science, a computer science researcher may be interested in whether the paper concerns a general overview of a subject, such as "computer graphics", or concerns more specific concepts such as "infographics" or "TreeMap visualizations". Similarly, the researcher may want to compare papers that appear in the same conference session to see the similarities and differences that may exist between the papers. In such scenarios, the overview visualizations should also represent the computer science domain and how the knowledge is structured in the domain.

While approaches such as topic modeling can provide a bottom-up categorization or thematic separation of a document's text, domain knowledge is often organized formally by experts in the corresponding domains using Ontologies. An ontology, defined as an "explicit specification of a conceptualization" [136, p. 199], is a widely-accepted way in which domain knowledge is formally represented. A knowledge-based overview of a document that uses as a reference the corresponding domain ontology can thus provide a conceptual overview for the domain expert. Such a view can also be used structurally to help the expert compare two or more documents based on the concepts they cover.

In order to provide documents examination from the viewpoint of a specific domain, we present ConceptScope, a text visualization technique that provides a domain-specific overview by referring to a relevant ontology to infer the conceptual structure of the document(s) being examined. ConceptScope uses a Bubble Treemap view [131] to represent concept hierarchies, highlighting concepts from the ontology that exist within the document and their relationships with other concepts in the document, as well as key "parent" concepts in the Ontology. Each concept "bubble" is also populated

with a word cloud that represents text from the document that relates to the concept, providing a contextual overview. Through a set of multiple coordinated views of text, structural overviews, and keyword-in-context (KWIC) views, ConceptScope helps users navigate a document from a specific domain perspective. ConceptScope can also be used to visually and conceptually compare multiple documents using the same domain ontology as a reference. To aid a domain novice, we also provide the user with navigable tooltips that provide concept explanations linking to external references.

We illustrate the utility of ConceptScope by building a prototype application that visualizes computer science-related documents such as research abstracts and articles using the Computer Science Ontology (CSO) as its reference. Through a set of use-case scenarios, we highlight the navigation, exploration, and comparison functions afforded by the technique, and discuss its extension to other domains and scenarios. We also present a brief comparison of ConceptScope with DocuBurst [73] through a qualitative, between-subjects study. Based on our observations, we find that ConceptScope's ontology-based visualization and its grouping of concept-related word clouds in the Bubble Treemap helps participants define and contextualize concepts, and explore new concepts related to a given concept. However, ConceptScope's domain-dependency makes it less suitable for viewing and comparing documents that span domains.

4.2 Related Work

This chapter proposes an interactive knowledge-based overview representation of text content. For our approach, we draw from existing techniques to identify themes or topics in the text, and visual representations of these topics. In this section, we outline existing work in this area and explain our reasoning behind our choice of inspiration from the existing work.

4.2.1 Thematic Visualizations of Document Content

Initial approaches to providing overview visualizations of document content used metrics such as sentence length, Simpson's Index, and *Hapax Legomena* as "literature fingerprints" to characterize documents [178]. This approach was later used to create a

visual analysis tool called VisRA [260] that helped writers review and edit their work for better readability using these representations. Among less abstract representations, Wordle [364] is the most popular. Wordle represents a text corpus as a cluster of words called a *word cloud*, with each word scaled according to its frequency of occurrence in the text. This idea is adapted to other techniques to characterize document content and structures within text, such as the Word Tree [375], which aggregated similar phrases in sentences in a text, Phrase Nets [361] that visualized text as a graph of concepts linked by relationships of the same type found in the text, and Parallel Tag Clouds [74], that show tag clouds on parallel axes to compare multiple documents.

When examining multiple text documents, it is important to identify the various types of connections between them. One of the most well-known tools used to identify inter-document connections is Jigsaw [337], which uses names, locations, and dates to show list, calendar, and thumbnail views of multiple documents. While Jigsaw simply uses text occurrences to form the connections, more sophisticated approaches have since been proposed. Tiara [376]—another system designed for intelligence analysis—uses topic modeling with a temporal component to highlight the change in document themes over time. ThemeDelta [121] allows thematic comparison between multiple documents (or similar documents over time) by combining word clouds with parallel axis visualizations.

More recently, topic modeling-based approaches have been incorporated to provide thematic overviews of text content. For instance, TopicNets [133] uses a graph-based representation where both documents and topics are nodes and links exist between documents and topics, thus serving to form clusters of thematically-related documents. Serendip [6] refines this idea and provides a multi-scale view of text corpora. It uses topic modeling along with document metadata to view patterns at the corpus level, text level, and word level. Oelke et al. [261] use a topic model-based approach to compare document collections, using what they call a "DiTop-View" with topic glyphs arranged on a 2D space to represent the document distribution. ConToVi [95] is a more recent work that uses topic modeling on multi-party conversations to reveal speech patterns of

individual speakers and trends in conversations. While topic model-based approaches are useful for identifying themes within collections of documents, a knowledge-based approach requires the use of human-organized representations of information, which are discussed in the following section.

4.2.2 Knowledge-Based Visualizations

As structured knowledge representation models [126], ontologies are widely used in the field of medicine/biology [126], engineering [299, 381], sociology [154], computer science [340] and so on. Achich et al. [2] review different application domains and generic visualization pipelines of ontology visualization.

According to various application fields and utilizing purpose, there are multiple methods to visualize the knowledge stored in ontology. The review of Katifori and Akrivi [173] systematically categorized these methods according to the dimension of the visualization. Ten years later, Dudáš et al. [93] further extended this work by adding more recently emerged visualizations. Among these visual encodings, we find inspiration in the matrix view of NodeTrix [154], the sunburst view of Phenotype [126] and the context view of NEREx [96].

Our work is inspired by DocuBurst [73], which was the first visualization from the point of view of human-organized structure of knowledge. DocuBurst uses hyponymy, or "is-A" relationship in the English lexicon to identify hierarchical relationships within a given documents, or when comparing two documents. The hierarchy is visualized as a sunburst diagram supported by coordinated views of text content and keyword-in-context views. While DocuBurst uses WordNet—a lexical database of the English language—as its reference, we use open-source domain ontologies, e.g. Computer Science Ontology (CSO)[1], in order to provide a more focused, domain-specific overview of documents.

[1]https://cso.kmi.open.ac.uk/home

4.2.3 Hierarchical Layouts

Visualization of a knowledge-based document overview needs to incorporate the hierarchical information inherent to the knowledge base. While a tree is the common representation of such a hierarchy, it is usually more suitable for showing the structure rather than the content of the information presented. The most famous alternative for representing hierarchical information is the TreeMap [321], a two-dimensional, space-filling layout that represents hierarchy through nesting and a second quantity such as percentage contribution to the whole as the area. Alternatives to TreeMaps such as Icicle plots and Radial TreeMaps [20] and Sunburst diagrams [336] have since been proposed and incorporated into standard visualizations of hierarchies. DocuBurst [73] referenced in the previous section uses the Sunburst diagram as its hierarchical visualization.

While the original TreeMap has afforded enough space in the representation to portray content, it often comes at the cost of some loss of detail in the hierarchy. Alternatives such as circle packing [372] and more recently, Bubble Treemaps [131] have been proposed to address this issue. We incorporate the Bubble Treemap into our design for its relative compactness compared to circle packing, and its use of space that allows for some content representation.

4.3 Design Requirement

In this section, we break down our overall need to provide a knowledge-based overview of document content into specific requirements to inform the design of ConceptScope.

R1 Provide Conceptual Overview: When reading a long document from an unfamiliar domain—such as an academic paper—the reader can benefit from a high-level overview of the information provided. While word clouds can provide a simple overview of the *text* in the document, a lack of understanding of the technical terms might hinder the reader in understanding the overview representation. Instead, an overview that stems from a fundamental categorization of the domain itself—as represented by the hierarchical organization of concepts often available in an ontology—can provide an overview that is accessible to both novices and

experts in the domain.

R2 Reveal Contextual Information: The document text and the ontology do not always overlap. From the point of view of the ontology, the document contains non-relevant information, but information nevertheless important for the reader. For instance, a research paper introducing a new search algorithm can introduce several concepts in the knowledge base of search algorithms. The paper would also make arguments for and against certain algorithms. The reader may benefit considerably from the structure and content of these arguments, which are lost if the overview visualization focuses solely on the ontological components. A way to provide the contextual information surrounding these concepts is thus needed.

R3 Support Exploration of New Knowledge: When exploring a concept that is a subdomain of a domain that is only partially known to the reader, they may be interested in other sub-domains of the domain. For example, if the term "quicksort" appears in an algorithm paper, the reader might want to know of other sorting algorithms such as "bubble sort" and "merge sort". They may also want to learn about related terms such as "divide and conquer" and "time complexity". These new terms may not appear in the document text, but forms an essential component of knowledge that extends from—and aids the understanding of—the core concept (i.e. quicksort). We thus need ways to enable users to access information from the ontology that is related to the concept of interest.

R4 Support Multi-document Comparison: Document comparison is a common requirement that emerges from the creation of visual overviews of documents [73]. In the case of our scenario, the comparison is likely to be conceptual: to get a quick comparison of concepts that are common to multiple documents, and those that are unique to one. The reader may also want to simply compare the differences between the information provided in two documents. While documents such as academic papers may contain abstract which summarizes the main content of the article, it may not be sufficient enough to cover all the concepts that are covered in the papers, not to mention the similarities and differences. Therefore, our tool

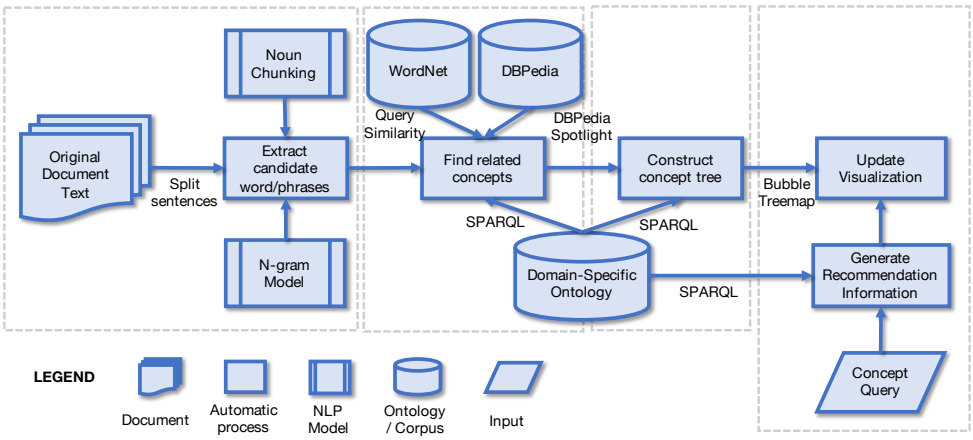


Figure 4.1: Data processing pipeline for ConceptScope.

should be able to provide visual support for users to compare and analyse the conceptual structure and content between two or more documents.

4.4 Implementation

In order to provide the knowledge-based conceptual overviews of a given document, an appropriate mechanism is needed to parse the document and compose queries to the reference ontology. An appropriate representation of the concept needs to be automatically generated in a way that reflects its hierarchy in the domain ontology as well as its occurrence in the document. To achieve this, we need to incorporate techniques from multiple areas including natural language processing, ontology querying, and information visualization. Figure 4.1 shows the framework of assembling them into a pipeline and the section number describing the corresponding technical details.

4.4.1 Generating Query Candidates

Ontology queries are typically performed using SPARQL (SPARQL Protocol And RDF Query Language) [365], which typically use "triples" (subject, predicate, and object) or parts thereof. In our case, trials showed that an exact triple was unlikely to be constructed from the document, nor was it deemed necessary. Instead, it was more important to have the subjects or object be specific terms that are likely to be present in the ontology. We construct these queries from the document with a sentence-level granularity. In order to construct the query terms, we use two approaches: noun

chunking, and n-gram identification.

Noun chunking is the process of extracting subsets of noun phrases such that they do not contain other noun phrases within them [26]. This allows us to identify specific terms that may be relevant to a domain ontology. For instance, when referencing the computer science ontology, terms such as "object-oriented programming" and "local area network" are much more meaningful than the individual words that make up these terms ("local", "object", or "area"). For this reason, we also do not resort to stemming or lemmatization as they change the morphology of the word (e.g., "oriented", if lemmatized to "orient", forms "object-orient programming") which renders the noun chunk invalid as a query candidate. Noun chunks can also include leading or trailing stop words, which are trimmed in order to generate the query candidates.

Noun chunking can produce phrases that contain query candidates, but are not query candidates themselves. For instance, a paper about animation may include multiple variances of animation like "2D computer animation", "stop-motion animation" and "animated transition". Some of these may appear within noun chunks, but not by themselves. To identify such cases, we identify groups of words that commonly occur together in the document as n-grams.

4.4.2 Mapping Queries to Concepts

Once the query candidates are identified, the next step is to map these candidates to the corresponding concepts in the domain ontology of interest. This involves two steps: (1) perform identical matches, i.e. concepts that correspond exactly to those in the ontology, and (2) reduce the number of "failed" matches, i.e. concepts that are related but not present in the ontology. Step 2 is often necessary due to the incompleteness and lack of strict formatting in some of the domain ontologies. For instance, the computer science ontology is not as well-populated as, say, medical or biological ontologies such as the human phenotype ontology.

The two steps—accurate matching and "fuzzy" matching—are illustrated in lines 8 through 15 in Algorithm 1. For any given candidate, we first look for an accurate match in the domain-specific ontology. We then construct a dictionary that includes

all of the concepts in the ontology for an effective search. However, the number of concepts that can be directly detected by accurate matching is small. This is because of the mismatch between specific forms in which a concept is listed in the ontology and its many variations in the document. For instance, "object-oriented programming" may be the exact match in the ontology, but it might appear in the text as "object-oriented approach" which is clearly related buy cannot be identified with an accurate match. In order to solve this problem, we introduce a fuzzy match.

Algorithm 1 Detect CSO Concepts in Document

Input: document text $stringDoc$

Output: concept dictionary $dictConcept$

```
1:  listSent ← Split(stringDoc)
2:  modelNGram ← TrainNGram(listSent)
3:  dictConcept ← ∅
4:  for stringSent in listSent
5:      listChunk ← NounChunking(stringSent)
6:      listNGram ← modelNGram(stringSent)
7:      listCand ← listChunk ∪ listNGram
8:      for stringCand in listCand
9:          if QueryCSO(stringCand) ≠ ∅
10:             dictConcept ← dictConcept ∪ QueryCSO(stringCand)
11:         else
12:             listCand ← DBpediaSpotlight(stringCand)
13:             listCand ← Filter(listCand, threshold)
14:             if QueryCSO(listCand) ≠ ∅
15:                 dictConcept ← dictConcept ∪ QueryCSO(listCand)
```

The goal of fuzzy match is to match the candidate to a concept that is very close to but not exactly equal to the candidate. In our prototype system, we use the computer science ontology (CSO) as the domain-specific ontology. The CSO also incorporates links of the form *"sameAs"* (http://www.w3.org/2002/07/owl#sameAs), that connect to

DBPedia [198], a broader, but less strictly-defined and less domain-specific ontology. We use these links and leverage the DBpedia Lookup Service [153] to find related DBpedia concepts and link them back to CSO. After checking the Wu-Palmer similarity between the CSO concept detected in this way and the original candidate using WordNet [107], we add the concept to the dictionary if the similarity is above a threshold. This threshold is currently determined by trial and error.

4.4.3 Hierarchy Reconstruction

The concept dictionary constructed thus far does not yet incorporate hierarchical information. In order to retrieve and store the hierarchical information from the ontology, we query the paths from every detected concept to the root of the ontology and use them to restructure the concept dictionary as a tree. The final output of this algorithm—the concept tree—can be directly converted to a JSON file and used to automatically render the visualization.

4.5 Visualization

In this section we discuss the visualization design and the interactions built upon the visual interface of ConceptScope.

4.5.1 Visual Encoding

We choose Bubble Treemaps proposed by Görtler et al. [131] as our primary visualization. This visualization is originally designed for uncertainty visualization, but we find it suitable for our application in terms of hierarchy representation and space organization. We use the original layout algorithm of the Bubble Treemap, but adapt the visual encoding and interaction strategies to meet our design requirements.

4.5.1.1 Hierarchy Presentation

In a Bubble Treemap, the deepest levels of the hierarchy are represented as circles, with successive higher levels forming contours around their "child" levels. We use the circles to represent the terms that appear (or have corresponding synonyms) in the original document as well as in the ontology. The outer contours represent concepts

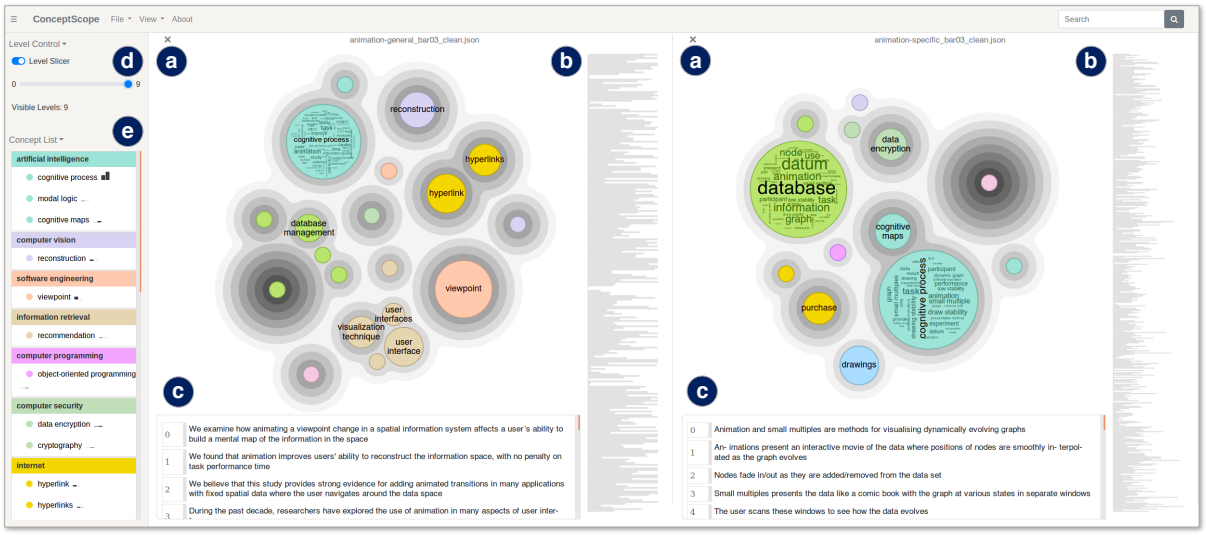


Figure 4.2: The ConceptScope interface representing two research papers discussing animation. The Bubble Treemaps (a) provide overviews, with the one on right showing a paper covering more specific topics than the one on the left. Supporting transcript (b) and text (c) views, along with a concept list (e) allow exploration and comparison between the documents.

that do not explicitly appear in the document, but still represent parent concepts from the ontology. These parent concepts are identified using the ontology query process demonstrated in Algorithm 1. The outermost contour forms the "root" of the ontology, with successive inner contours representing its child concepts. For example, in the computer science ontology (CSO) [305] we use for our case studies, the term "computer science" is the root concept in the ontology.

Inner Circles The function of the innermost circles—representing concepts that are present in the ontology *and* in the document—is to provide a clear representation of the terms that are directly connected to the document. The size of the circles are proportional to the frequency with which the corresponding term appears in the document. The fill color of a given circle corresponds to the highest "parent concept" it belongs to, just below the root. Although the Bubble Treemap layout already gathers together circles that share the same parents, we visually reinforce such relationship by assigning the same color to circles with common highest ancestor (besides the root). These "highest parent concepts", divide the root term into several subclasses and help users to better grasp the various areas the document covers. In order to make sure the circles' colors are perceptually uniform, we create the isoluminant palette [186] from

the CIELAB color space to ensure perceptual uniformity between the concepts shown.

Surrounding Contours The contours surrounding the circles show hierarchical relationships between the concepts that occur in the document. After exploring several encoding options for the contours to best represent related concepts while highlighting hierarchies, we chose fill colors of decreasing luminance to represent "deeper" contours in the hierarchy.

4.5.1.2 List Presentation

Effective as the Bubble Treemap is, it is not intuitive enough for the users to understand and grasp all necessary information at a glance. We therefore augment the visualization with a multi-function widget which combines concept list, legend, and bar charts representing term frequencies to solve this problem. Inspired by scented widgets [378], the multi-function widget presents important supporting information in a compact representation. As a concept list, this tool represents every concept detected in current-loaded document(s) as a list item, the background color of which is the same as the corresponding concept circle(s) shown in the Bubble Treemap. We group the concepts sharing the "highest super topic" together, with an additional list item showing the common "highest super topic" of each group. This concept list also acts as a legend showing the connection between each color and their corresponding "highest super topic". We also attach a sparkline for each list item to show the distribution of current concept across multiple documents (when multiple documents are loaded).

4.5.1.3 Incorporating Word Clouds

An unlabeled Bubble Treemap can be too abstract a representation for the user to comprehend. On the other hand, labeling every concept may result in a cluttered view which would also make comprehension difficult. We thus provide three levels of labeling for the concept: unlabeled (if the concept circle is too small), labeled (if the concept circle is large enough to fit its corresponding concept name), and labeled with context (where a word cloud of related terms from the document is combined with the concept label). The interactions to retrieve information from these views are discussed in the following section.

4.5.2 Interaction

ConceptScope provides linking between views and semantic overview and detail views to help analyze the document(s) and its concepts. These interactions support two modes of document analysis: exploration and comparison. We will first describe the overview and detail interactions and follow them with the modes of analysis.

4.5.2.1 Overview+Detail Interactions

To eliminate the potential confusion caused by the users' unfamiliarity with the Bubble Treemap, we introduce interactions to acquaint them with the visual schema and provide details on demand [322]. The Bubble Treemap provides a compact view of the domain-relevant concepts, their hierarchical structure in the ontology, as well as their context in the original document. In order to make this compact representation easier to understand, we design two interactions to present information that the user may seek: (1) a *level slicer* to "slice" the Bubble Treemap at any level to examine parent concepts, and (2) *semantic zooming*, which allows the user to zoom in to a concept circle to examine its corresponding word cloud (described in section 4.5.1.3. The users can chose and combine these two tools according to their preference.

The **Level Slicer** is designed to help novice users quickly build a connection between the nested layout of the Bubble Treemap and the hierarchical structure of ontology. This tool allows the user to choose the level of the parent concept that they want to see on the screen by sliding the slider bar. When the view initializes, all levels of the Bubble Treemap are shown to provide an overview, but the labels corresponding to parent contours are are concealed. Once the "child" concepts are sliced away by the slicer, the corresponding label of the newly exposed parent concepts are made visible. This tool facilitate users to inspect any cross section they are interested in from the whole hierarchical structure.

Semantic Zooming is designed to provide different granularity of information based on the users' need. As explained in section 4.5.1.3, users may see three levels of detail for the same concept circle: unlabeled, labeled, and labeled with word cloud. When users zoom in and our of the graph, the size of every circle changes and its appearance

transforms among the three based on the available space inside it.

ConceptScope also reveals more information about a concept including its thumbnail, definition, related concepts, and its context in the text. These views allow the exploration of concepts that do not themselves occur in the document, but are related to the ones that do occur.

4.5.2.2 Exploration Mode

The exploration mode—meant for inspecting a single document—provides conceptual overview and detail representations of the document using the ontology as a reference. With the static Bubble Treemap, it is almost impossible for novice users to build the connection between a circle in the graph and a word/phrase in the original text. Users might want to explore related knowledge in the domain ontology about the concepts shown in the Bubble Treemap. Following the information-seeking mantra [322], we design a set of small widgets which can be easily evoked and interacted with to the Bubble Treemap.

To *connect the Bubble Treemap and the original document*, we create a high-level transcript view and a raw text view. The high-level transcript view can be seen as a "minimap" of the document, with each sentence represented by a series of horizontal lines scaled to sentence lengths (Fig. 4.2 (b)). In the raw text view, the raw text is shown to provide a convenient context acquisition (Fig. 4.2 (c)). These two views as well as the Bubble Treemap view are fully coordinated, so that interacting with one view highlights related information in the other views. For example, if the users hover over a circle representing a concept in the Bubble Treemap view, the lines corresponding to the sentences that contain this concept in the transcript view and the text of the sentence in the raw text view are also be highlighted.

Interacting with a concept circle also reveals a tooltip that shows the concept definition, thumbnail, and a link to the relevant concept page on DBPedia. The tooltip also provides links to other related concepts that may not be present in the document, to provide context from an ontology point of view.

4.5.2.3 Comparative Mode

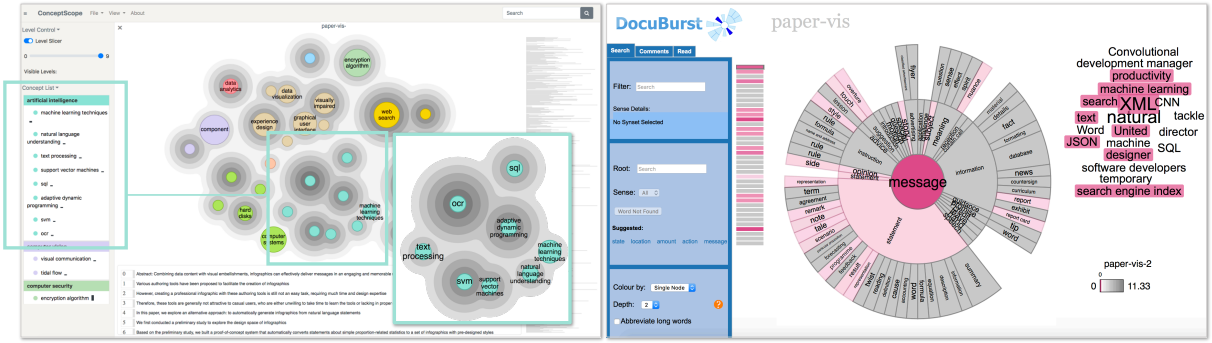
The comparative mode assists users in comparing multiple documents and explore conceptual similarities and differences between the documents. As the name suggests, loading multiple documents creates multiple, side-by-side Bubble Treemap views, one for each document. Concepts common to two or more documents are encoded in the same color across the Bubble Treemaps.

The comparative mode provides similar interactions as the exploration mode. In additionm the sparklines mentioned in section 4.5.1.2 can provide the users a quick overview of the relative frequency with which each concept occurs across the documents. The users can compare the concepts that interest them by hovering or searching. If they know where a concept is located in any of the Bubble Treemaps, the user can simply hover on the corresponding circle or contour, which highlights the concept—if available—across all the Bubble Treemaps. They can also directly search for the concept in the search field (top right corner in Fig. 4.2) to highlight all relevant circles and contours across the Bubble Treemaps. The users can thus quickly get an idea about where and how their concepts of interest are distributed across different documents.

The switchover between exploration mode and comparative mode does not require explicit user operation. Loading a single document shows the exploration mode, while loading additional documents sets ConceptScope to comparison mode. The exploratory features are always available regardless of the number of documents, as comparison also requires a degree of exploration. We also provide a "switch" for semantic zooming to make sure the users can explore or compare the Bubble Treemap(s) at whatever number of levels and size they want.

4.6 Use Case Scenarios

We briefly illustrate the use of ConceptScope for exploring and comparing documents with two use-case scenarios: exploring an academic paper, and comparing the transcripts of three TED talks.



(a) ConceptScope Interface (b) DocuBurst Interface

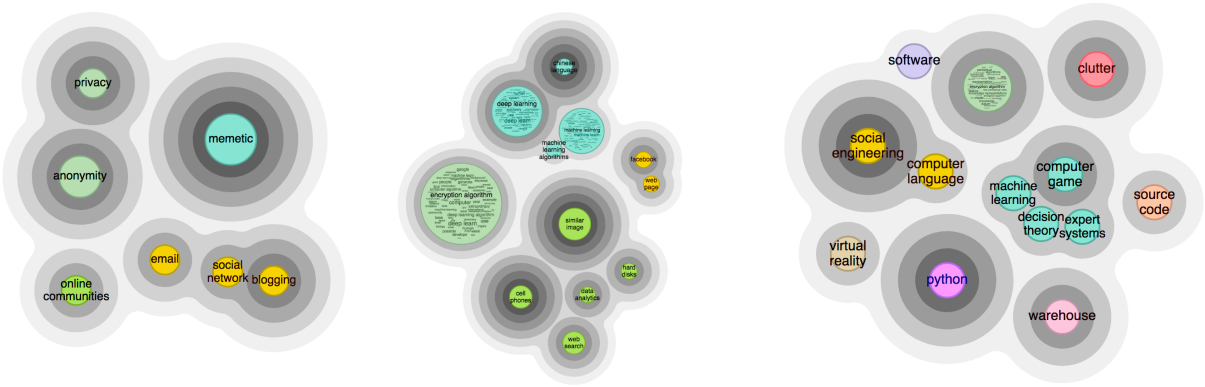
Figure 4.3: Overview of an IEEE VIS 2019 Paper [79] in (a) ConceptScope and (b) DocuBurst.

4.6.1 Exploring an Academic Paper

We first use ConceptScope to visualize an academic paper [79] on automatic infographics generation, published in IEEE VIS 2019. To ensure the accuracy of our natural language processing components, we only keep the natural-language parts of the original paper, and remove text in references, tables, formulas and figure labels. We use the computer science ontology (CSO) as the reference ontology for this work. Fig. 4.3 shows the visualization, with the same paper shown in DocuBurst [73] for reference.

The Bubble Treemap shows over 30 computer science concepts directly or indirectly mentioned in the paper (requirement **R1**). Inspecting the concept list on the left, we see that the highest parent concepts of the ones identified in the document range from "human-computer interaction" to "artificial intelligence" to "computer system". Zooming in, we click on the bubble representing "OCR" (optical character recognition) and a tooltip pops up with the definition of this concept as well as the recommendation of concepts related to this one (**R3**). We examine the definitions and where the concept appears in the word cloud to see that it points to the use of OCR to identify key text in existing infographics (**R2**). We also see that these and most concepts under "artificial intelligence" appear under the related work section. We thus infer that these concepts might only be mentioned as background or references to other work, and not as a fundamental contribution of the paper.

Figure 4.3 (right) shows the DocuBurst visualization using the root "message". We notice that almost all computer-science-related concepts identified by DocuBurst can



The case for anonymity online
Speaker profile: Organization founder
Tag: Internet origin stories
Duration: 11:09

The wonderful and terrifying implications of computers that can learn
Speaker profile: Philosopher
Tag: Artificial intelligence
Duration: 19:17

What happens when our computer gets smarter than we are?
Speaker profile: Data Scientist
Tag: Artificial intelligence
Duration: 16:03

Figure 4.4: ConceptScope visualizations comparing the transcripts of three TED Talks. The title of each talk is shown in red under each visualization, along with the speaker profile and talk metadata.

be detected by ConceptScope as well. In terms of space efficiency, DocuBurst has the advantage of providing more compact visualization with its Sunburst diagram. However, DocuBurst offers fewer options for contextual views. In ConceptScope, the word clouds in each concept circle provide a contextual overview, and aids concept exploration outside the realm of the document with our thumbnail views of concepts and the links to DBPedia.

4.6.2 Comparing Transcripts of TED Talks

To illustrate multi-document comparison, we load the transcripts of three TED Talks [35, 160,246], all of which are tagged under the "computers" category on the TED webpage. Fig. 4.4 shows the distribution and depth of concepts, along with information about each talk.

Loading all three documents into ConceptScope creates three panels (similar to that shown for two papers in Fig. 4.2), each containing the Bubble Treemap view, transcript view and raw text view for the corresponding transcript. The Bubble Treemap immediately illustrates the differences and similarities between concepts across the three talks, which can further be explored as all three views are coordinated. We notice that all three of the talks mention concepts under the parent topics of "internet", "computer security" and "artificial intelligence". One reasonable explanation is that

these topics covers many basic terms in computer science, so it is almost unavoidable to use them in a computer-science-related technical presentation. When inspecting the concept list and Bubble Treemaps, we notice that concepts that belong to "artificial intelligence" appear more in talk No. 2 and talk No. 3, which makes sense as the two talks have the additional tag of "AI" on the TED webpage.

Talk No. 1 discusses the issue of privacy on online forums, and concepts of privacy and anonymity fall outside the current version of the computer science ontology. In addition, the talk does not delve deep into computer science concepts. This results in a Bubble Treemap that covers very few concepts. Talk No. 2 is delivered by a data scientist who talks about computer science concepts, specifically "algorithms", "machine learning", and "deep learning", which are reflected in the Bubble Treemap. Finally, Talk No. 3 is presented by a philosopher who talks about broader implications of machine learning, also providing a historical perspective. This is reflected in the Bubble Treemap, showing the broadest concept coverage of the three talks, with no one concept being too dominant.

4.7 User Study

We conducted a controlled study to evaluate whether the visualization & interaction design and the use of a domain-specific reference ontology renders ConceptScope effective in exploring single documents or comparing multiple documents. Specifically, we intended to understand whether ConceptScope was effective in helping users: (1) summarize the content of a document effectively with a domain-specific concept overview (**R1**); (2) glean what a document says about any given concept in the context of the document (**R2**); (3) become aware of new concepts and their connections (**R3**); and (4) discover enough similarities and differences among multiple documents (**R4**). In order to provide a baseline, we used DocuBurst [73], the popular content-oriented document visualization tool that provides a non-domain-specific overview of documents using the WordNet [107] taxonomy. We thus conducted a between-subjects study comparing participants that used ConceptScope with participants that used DocuBurst.

4.7.1 Participants

We recruited 18 participants (10 female, 8 male) aged between 18 and 44 years. The participants comprised 16 Ph.D. students, 1 undergraduate student, and 1 employee of a technology company. Seventeen participants had computer science backgrounds, of which 12 specialized in visualization and HCI, 1 in high performance computing, 1 in natural language generation and multi-modal learning, while 3 didn't report their specialized field. The one remaining participant had a design and education background, specializing in learning and user experience design. Two of the 18 participants reported themselves as native English speakers.

4.7.2 Conditions and Task Design

We chose DocuBurst as a baseline for our evaluation because both tools visualize document content from the perspective of a human-curated knowledge base, instead of computationally-derived classifications such as topic modeling or latent semantic indexing. DocuBurst provides an overview of documents based on the non-domain-specific "is-a" relationship in WordNet, while our prototype is based on domain-specific ontologies, in this case the Computer Science Ontology (CSO). We asked each participant to perform the same tasks using the interface assigned to them (ConceptScope or DocuBurst) and compared interaction and behavior patterns across participants. Participants were given time to familiarize themselves with their assigned interface. They were then asked to perform the following tasks:

T1 Explore a single document: This task was divided into several sub-tasks, each aligned with a corresponding design requirement: (1) summarize the documents and provide relevant keywords (**R1**); (2) describe a specified concept based on its usage in the document (**R2**); (3) select (from a list of description) the context in which a given concept is used in the document (**R2**); (4) define several concepts before and after using the system, as well as rate confidence with the definition (**R3**); (5) identify concepts in the document related to given concept (**R3**); and (6) list the concepts (that the participants did not know before the study) in the document (**R3**). Participants were also asked whether they read the documents

before the study to account for potential confounds.

T2 Compare two documents: The participants were asked to compare two documents at a conceptual level (**R4**). Therefore, they were asked to identify common and unique concepts, as well as overall similarities and differences of the two documents. Again, they were asked whether they read the documents before the study to eliminate bias.

T3 Compare three documents: The questions that participants were asked to answer in this task were generally the same as task T2 but upon three documents. The participants were suggested to "identify a theme and explain their difference within the theme" when identifying the difference among three documents. Since DocuBurst was not capable of comparing more than two documents, this task was only assigned to participants using ConceptScope in the study.

In order to simulate participants' regular reading experience, we chose computer-science related academic papers or technical reports for all tasks of this study. For task T1 we used Munzner's nested model for validating visualizations [250]. Task T2 involved two papers discussing animation techniques: the first, a general evaluation of how animation could help users build mental map of spatial information [23], while the other focused on the role of animation in dynamic graph visualization [12]. To alleviate participant fatigue and manage their time, we chose to use relatively shorter transcripts of three 15–20 minute Ted Talks [35, 160, 355] in the "artificial intelligence" playlist instead of academic papers for task T3.

4.7.3 Study Setup

We conducted the study remotely owing to safety measures surrounding COVID-19. The participants were asked to access either of the tools from a remote server and participate in the study with their own machine and external devices. Fourteen of them used laptops with screen size ranging from 13 in. to 16 in. The other 5 used monitors with screen size ranging from 24 in. to 32 in. Fifteen participants used the Chrome browser, 2 used Safari, while one used Firefox for the tasks.

The setup, tasks, and durations were decided based on a within-subjects pilot of

the study described above with 2 participants: one native and one non-native English speaker. ConceptScope and DocuBurst employed different datasets in this study. The decisions to suggest time durations for the questions and to set up the final study as a between-subjects study were made based on the long duration of each session and on participant's fatigue toward the end of each session.

4.7.4 Procedure

Participants first responded to an online pre-survey providing their demographic and background information. Once they had finished familiarizing themselves with the interface, the participants performed the tasks described in section 4.7.2. Participants followed a concurrent think-aloud protocol while executing the tasks, with the moderator recording their verbalizations and their screen through a videoconferencing application. Finally, the participants were invited to finish a brief survey about the tool and share their feedback about their experience with the interface, both as open-ended responses and on the NASA TLX scale [149].

4.8 Results and Discussion

4.8.1 General Behavior Patterns

We categorized participants into two groups based on how they attempted to gather the information they needed to answer the questions: those that mainly used the visualization, and those that mainly used the raw text display. Seven of the 9 participants who used ConceptScope primarily used the main Bubble Treemap visualization to glean the required information, while the remaining 2 relied more on the raw text reading from the document. In DocuBurst, only 5 of the 9 participants used the main sunburst diagram as their main source of information, while 4 of the 9 chiefly relied on close reading of the text.

Participants using ConceptScope used the main visualization more than participants using DocuBurst. This was partly due to the raw text reading experience offered by the two interfaces, and partly due to the ability of the visualizations and the knowledge base in conveying a relevant overview. In ConceptScope, documents were split into

sentences and displayed in a relatively small vertical space (see Fig. 4.2c). Therefore, participants tended to read only a few sentences prior to and after the key sentence for a specific task instead of going through larger blocks of text. As participant $Pc7$ stated, "*because my resolution is small and my mouse is sensitive, so when I move it jumps between the text very easily (in transcript view). And this box (the tooltip showing the corresponding sentence) doesn't include the complete paragraph, so it's easy to get lost...*". In contrast, DocuBurst showed text as paragraphs in a view that used more vertical space, such that users were able to read the sentences more easily. "*One thing I like this system is when I click some words, they divide it as paragraph rather than the entire document...help me read more specifically*", said participant $Pd3$.

When answering a given question, 7 of the 9 participants using ConceptScope searched or explored related information in the interface and summarized their findings. The remaining 2 mainly attempted to recall the answer from earlier explorations, and then referring to the interface to confirm. For DocuBurst, this distribution was 5 participants chiefly exploring the interface, and 4 chiefly recalling the answer. Compared to ConceptScope, more participants using DocuBurst answered questions from memory, almost equal to the number of participants who explored the visualizations to find answers. Participant comments indicated that they felt they might spend too much time in locating the required information. For instance, when trying to find common concepts between two documents (task T2), participant $Pd9$ who used DocuBurst commented that "*it is really hard to see all of them (words in the sunburst diagram). And I really wanna expand one of those, but then I'm not sure if it will cover all the things that I wanna see. . . . It's hard to go back to where you came from*". Similar comments were also made by those using DocuBurst to first gather information before answering the question.

4.8.2 Task-Level Observations

We further separate task-wise participant behavior based on how they achieved specific objectives within tasks. This behavior was not restricted to any one task; rather, it characterized how certain participants chose to access information across tasks.

Document Sensemaking: When exploring the full document (T1), participants

across both interfaces attempted to use the visualization to quickly get a sense of what topics were addressed in the document. Ten of the 18 participants (6 using ConceptScope, 4 using DocuBurst) were able to quickly identify that the document was an "InfoVis paper". Certain participant behaviors were similar across both interfaces. Most of them explored the document using the main visualization first, and only later resorted to close reading of the text. Even after recognizing it as an academic paper, only 2 participants relied on the paper structure (e.g., abstract/introduction) to get a sense of the document.

However, DocuBurst users were more easily overwhelmed by the large number of words in the Sunburst diagram, many of which (they felt) were not closely related to the main theme of the document. Participant $Pd3$ observed, "*some words maybe appear really frequently, but it's actually not very important ... it's just because it's used very frequently by any document*." Another participant ($Pd2$) found it difficult to organize the words into themes, saying "*it is a little bit hard to place the information together, because you don't know what the correlation is between (among) these things (i.e. the concepts provided)*"

Participants' perception of the document when using ConceptScope was largely influenced by the extent of overlap between the document text and the ontology. For instance, the concept "visualization" being well-defined in the ontology, was successfully identified by 8 out of 9 participants in T1. However, the concept "animation" was not as well-defined in CSO, as a result of which 5 out of 9 participants failed to determine that task T2 involved papers discussing animation. In comparison, 8 of the 9 using DocuBurst were able to successfully identify the animation theme.

Concept Sensemaking: When making sense of a concept (**R2, R3**), most of the participants chose to locate it in the main visualization first, and only then looked at the other views to answer relevant questions. To locate a specific concept in the visualization, participants' strategies varied based on the solutions available in the interface and their preference.

In ConceptScope, 5 of the 9 participants used the search feature, while the others preferred to visually search the concept in the interface, i.e. looking it up in the concept

list or directly checking the Bubble Treemap. Since DocuBurst did not feature a search box available, all 9 participants set the concept to locate as the root word. However, eight of the 9 participants failed with this strategy and had to set alternatives of the original concept (e.g. the a parent concept, a synonym, or a substring of the target concept) as root words. One unique strategy that at least 3 participants used to search in DocuBurst was to start from higher level concepts and dive deeper towards their targets in the sunburst diagram. Once again, their success depended on their choice of parent concepts: they often lost their way as they could not retrace their steps. In comparison, participants found it more straightforward to locate concepts in ConceptScope.

While participants using either interface chiefly attempted to define a concept (**R1**) by referring to the context of its use (**R2**), their approach to identify the context was different across the interfaces. In ConceptScope, the concordance view was used the most, with all 9 participants using this view to identify the context at least once. This was followed by the close reading of the transcript (used by 7 participants), with the word cloud being used by 6 participants at least once. Although DocuBurst also provided a word cloud, only 2 participants used it for context. This was likely because DocuBurst's word cloud was not organized into concepts as in ConceptScope, and furthermore, the word cloud in DocuBurst—designed to supplement the main visualization—only featured proper nouns that would not otherwise be visualized in the Sunburst diagram. To find related concepts (**R3**), participants using ConceptScope chiefly referred to the Bubble Treemap while DocuBurst users referred to the raw text view.

Multi-document Comparison: We observed participants' behavior when comparing documents both at the conceptual level and the full-text level (**R4**). Participants using ConceptScope used several techniques including highlighting concepts in the Bubble Treemap, highlighting concepts in the concept list, checking the relevant sparklines, and comparing the word cloud within a concept group. Five of the 9 participants reported that these techniques were sufficient to answer all of the questions in tasks T2 and T3. Participant $Pc1$ observed, "*just looking at this (the Bubble Treemap for the third document in T3), you can see some colors are different, means some different concepts exist*

here... you can immediately see it". When the visual clues were not enough to aid them summarize the similarities or differences between/among the documents, the other 4 participants resorted to close reading of the document.

In contrast, most of the participants using DocuBurst mentioned that the visualizations and interactions were not sufficient to help them compare the concepts or full text of the documents. Participant $Pd5$ commented, "*the visual encoding (distinguishing concepts between documents) is confusing to me*". Participant $Pd9$ felt "*it is really hard to see all of them (concepts)*" when they tried to identify unique concepts of one document. Both participant $Pd1$ and $Pd8$ were distracted by general words like "*part*" and "*paper*", because they were the only few words marked as being shared by both documents. As a solution, they chose to read the document text closely to make sure their responses to the questions were accurate enough.

4.8.3 Overall Feedback

Fig. 4.5 shows the difference in participant experience for the study between ConceptScope and DocuBurst. We can see from the figure that participants' experience was more or less similar between the two interfaces with the exception of frustration: participants using ConceptScope were less frustrated ($Md = 2, IQR = 1$) than those using DocuBurst ($Md = 4, IQR = 4$). Observation and feedback indicated that participants using DocuBurst found themselves distracted by less relevant concepts. Participant $Pd3$ stated that the interface didn't provide "important" keywords as expected: "*When I click 'person'... it (the corresponding sector in sunburst diagram) is really big, means that it is important. However, I don't think it is important based on what I've seen*". Participant $Pd4$ mistook the document in task T1 for a medical paper and participant $Pd9$ mistook those in T2 as related to chemistry, based on their (mistaken) interpretation of proper nouns in the word cloud.

As a general feedback, most participants using ConceptScope considered it suitable to provide an overview for unfamiliar documents, while those using DocuBurst felt it was better suited as a supplementary tool when exploring familiar documents. Typical comments about ConceptScope included "*these multiple views are nice and easy*

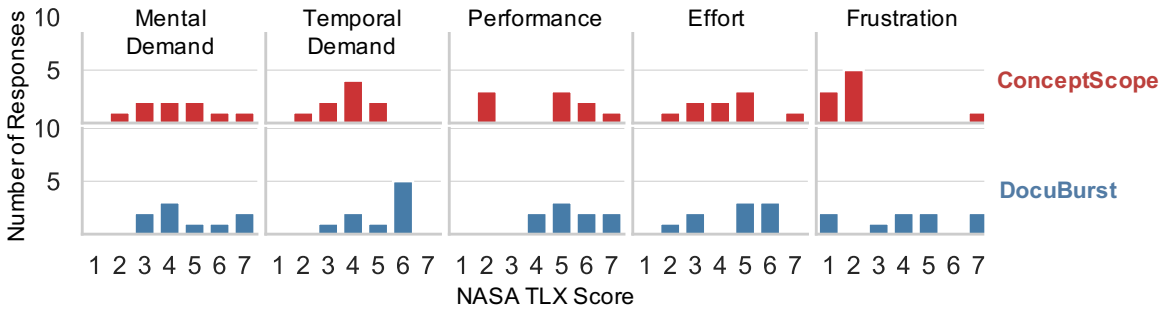


Figure 4.5: Distribution of NASA TLX responses showing participant feedback towards ConceptScope and DocuBurst (Rate 1 represents "very low", while rate 7 represents "very high").

to understand"(participant $Pc2$), "*it seems like a pretty useful tool especially for exploring large set of documents to get an idea of what the main topics are, what kind of researchers are active*" ($Pc7$). With DocuBurst, participant $Pd2$ suggested that "*the tool should be used as a supplementary tool ... doesn't help too much with understanding the document*". In addition, participants also reflected that the the learning curve for both tools were relatively long. "*It was hard at the beginning, but not so hard later*", commented by participant $Pc4$.

When regarding the features of each interface, the Level Slicer (section 4.5.2.1) in ConceptScope was marked as least useful by participants. Participant $Pc1$ observed that "*the level slicer is probably useful if the document is extremely complex ... but this dataset is relatively simple*". Three of the participants thought the list view (section 4.5.1.2) was the most useful feature. We also observed 7 participants used it for comparison tasks and 4 participants used it to search target concepts in the study. Only 2 participants rated the Bubble Treemap (section 4.5.1.1) as the most useful feature, while one marked it as the least useful one. Yet, we did see 7 participants used it as their major source of visual clues when comparing multiple documents. It is likely that the participants used the Bubble Treemap as providing supplementary information to the concept list view, which they found to be most useful.

4.9 Summary

In this work, I propose ConceptScope, an interface that aids a knowledge-based exploration and comparison of documents based on a reference domain ontology. I

present the use of a Bubble Treemap visualization as the primary overview visualization to show the distribution of concepts for a document of interest, and describe our approach to translate document content into appropriate queries that best reflect the concept spread and show their hierarchical relationships in the domain ontology.

I illustrate our approach using the computer science ontology as our reference. I demonstrate the use of ConceptScope for document exploration and comparison, and then evaluate ConceptScope against DocuBurst, the first and most popular overview visualization based on human-curated knowledge. Based on participant behavior and feedback, I illustrate that ConceptScope's ontology-based visualization and grouped word clouds help participants define and contextualize concepts and explore concepts related to one given concept. On the other hand, ConceptScope's domain dependency makes it unsuitable for reviewing text that covers more than one discipline. In contrast, DocuBurst's domain-agnostic reference allows it to be applied more widely, though the overviews are less useful when highly domain-specific content is visualized. In addition, DocuBurst's interface is more amenable to close reading of the document.

For future work, I plan to address issues relating to the ontology lookup. One main disadvantage is the dependence on ontologies that may or may not be mature. I currently use DBPedia to "broaden" our lookup, but using DBPedia detracts from the strict definitions and relationship requirements to which domain ontologies need to adhere. Our Bubble Treemap visualization as well as our ontology lookup can currently support only one ontology. This makes it difficult to view documents of an interdisciplinary nature. I also intend to explore the application of our approach to real-time visualizations of online forums or technical communication in the form of emails or instant messengers.

Chapter 5
Knowledge Exploitation for Document Summarization

In this chapter, I introduce ConceptEVA, a mixed-initiative approach to generate, evaluate, and customize summaries for long and multi-topic documents by *exploiting knowledge* from an existing knowledge base, DBpedia, and human input. This application is inspired by the inefficiency of existing natural language processing and artificial intelligence approaches to summarize long and multi-topic documents—such as academic papers—for readers from different domains. ConceptEVA incorporates a custom multi-task longformer encoder decoder to summarize longer documents. Interactive visualizations of document concepts as a network reflecting both semantic relatedness and co-occurrence help users focus on concepts of interest. The user can select these concepts and automatically update the summary to emphasize them. In this chapter, I present two iterations of ConceptEVA evaluated through an expert review and a within-subjects study. We find that participants' satisfaction with customized summaries through ConceptEVA is higher than their own manually-generated summary, while incorporating critique into the summaries proved challenging. Based on our findings, I also make recommendations for designing summarization systems incorporating mixed-initiative interactions.

5.1 Introduction

The notion of automated text summarization—compression of long text passages into shorter text without losing essential information—has been an open problem since over half a century ago [223]. The main goals of automated text summarization are to present the salient concepts of a given document in a compact way, and to minimize

repetition of the presented ideas or concepts [97]. Earlier techniques fall under the umbrella of extractive summarization where summaries are generated by extracting terms, phrases, or entire sentences from the source text using statistical techniques [139]. With advances in machine learning and specifically sequence-to-sequence language models, abstractive summarization—an approach that generates paraphrased text that still retains concepts from the original text—has gained recent popularity as it mimics summaries created by humans [317].

However, significant challenges in abstractive summarization remain, such as the summarization of long, complex, documents that span multiple knowledge domains. While approaches have been proposed for summarizing domain-specific text [210] and others for summarizing long documents [382], the challenge remains that there is no one "ideal" summary for such long and multi-domain documents. Automated summarization systems typically do not fare well when the source document spans multiple topics regardless of approach, i.e., extractive [140], abstractive, or hybrid [97].

Academic papers, especially those in the fields of design or human-computer interaction (HCI) where research tends to be cross-disciplinary, tend to fall under this category of long, multi-topic documents. For instance, a research article might span the fields of wearable technologies, privacy, and social justice. A summary of this article that is deemed useful by a researcher in wearable technology would be different from one deemed useful to a researcher in security and privacy. Yet, both summaries may still be perfectly valid summaries of the article. This subjectivity means that purely automated, black-box approaches to summary generation will not work. Instead, a human-in-the-loop approach is needed to allow the user to steer the automated summary generator to interactively generate a summary relevant to the user's interests.

To address this challenge, we present ConceptEVA, a mixed-initiative system for academic document readers and writers to generate, evaluate, and customize automated summaries. We build a multi-task Longformer Encoder Decoder (LED) [24] from a pretrained LED trained for scientific document summarization by fine-tuning it on two downstream tasks—paraphrasing and semantic sentence embedding—to handle

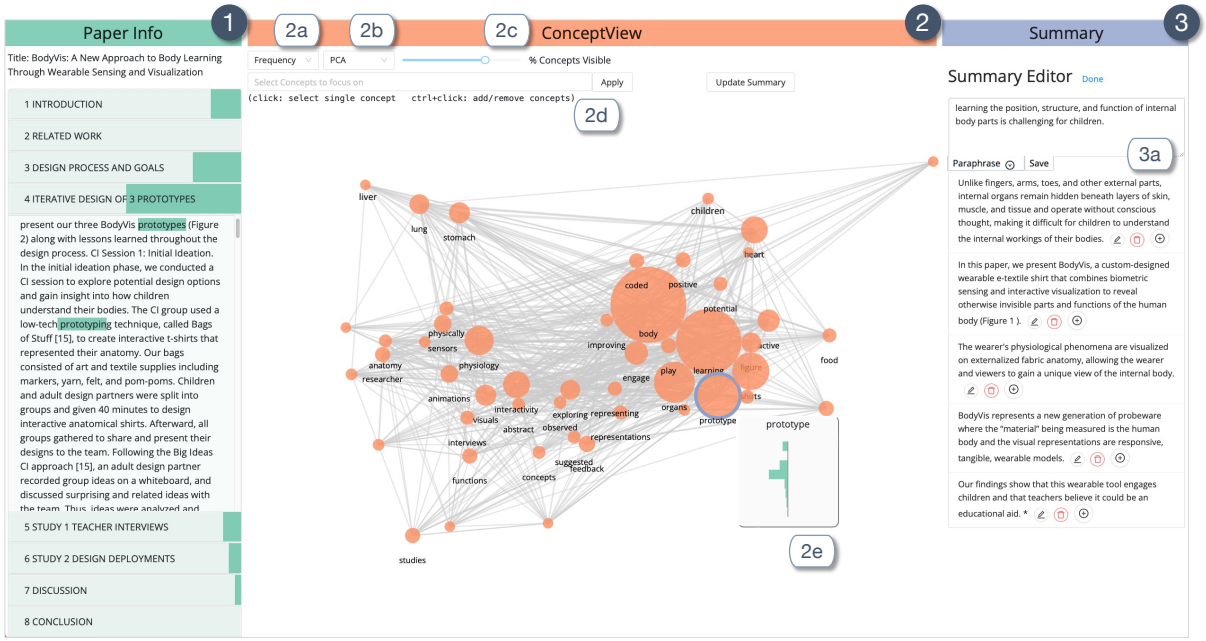


Figure 5.1: The ConceptEVA Interface shows a multi-disciplinary research paper [258] and its auto-generated summary. The interface can be separated into three main panels vertically. The panel on the left (1) shows a section-wise collapsed view of the research paper, while the panel on the right (3) shows the generated summary in the summary editor. At the center is the concept view (2), which displays the concepts extracted from the paper based on a reference ontology. The circles represent concepts whose layouts are decided by their text embedding and co-occurrence relationship with other concepts. The links indicate whether and how frequently the concepts they connect co-occur in the same sentence. A *concept glyph* comparing the concept's appearance distribution across the document and the summary is shown in (2e) upon user request. Users can also adjust the concept layout in (2) according to their interests by changing (2a) the information encoded with circle size, (2b) the text embedding projection algorithm, (2c) the percentage of concepts visible, or focusing on the concepts they are interested in (2d). They can also interactively edit (3a), reorder (with drag and drop), or delete any sentences in (3). The user can select concepts of interest for the summary generator, which then generates a new summary incorporating the selected concepts.

long documents. This approach uses the notion of attention mechanisms from transformers [363] at local levels to reduce memory usage, and at global levels to preserve information fidelity in longer documents.

In addition, ConceptEVA also supports summary customization for the user by visualizing the concepts—in this scenario, topics explicitly defined in an ontology or knowledge graphs—occurring in the document. The concepts are identified using a multi-domain ontology [16], and visualized as a force-directed layout of a graph network using metrics such as concept relatedness and concept co-occurrence in the document. We introduce a function that we call "focus-on", that allows the user

to select concept(s) of interest to surface and highlight other concepts related to the selected ones. The user can identify concepts to focus on and use them to steer the automated summarizer to generate a summary text in which the concepts of interest feature prominently. The user can also further edit the summary at the concept level by navigating the original document and selecting text to emphasize using the concepts of interest as a filter. They can also edit it at the sentence level by selecting alternative paraphrasing and sentence ordering.

The design of ConceptEVA informed by an initial survey of eight research practitioners, and refined through two stages of development and evaluation:

Iteration 1. A hierarchical summarization approach with a glyph-based visualization of concepts embedded in a two-dimensional projection, evaluated through an expert review of 3 participants;

Iteration 2. The final LED approach to summarization described above with concepts visualized as a force-directed layout that preserves both semantic relatedness and concept co-occurrence within the document. This version is evaluated by a within-subjects study of 12 participants using manually-generated summaries as a baseline.

Findings from the user study indicate that ConceptEVA is seen as helpful for participants in examining and verifying ideas, and using specific concepts of interest to explore related concepts and how they are addressed in the source document. ConceptEVA was also reported as more useful when the participants evaluated and customized a summary of a document that lay outside their domain of interest, while it was seen as less useful when the participant was knowledgeable about the domain or had a specific idea of what the summary should include. Using ConceptEVA for summarization allowed participants to address content-specific aspects of the summary, but inexeperienced participants found it more difficult to incorporate critique such as limitations and implications into the summary.

The chief contribution of this work is ConceptEVA, a mixed-initiative system that integrates interactive visual analysis and NLP techniques for evaluating and customiz-

ing long document summaries. Specifically, we fine tune an LED trained for scientific document summarization for paraphrasing and semantic sentence embedding, identify and visualize concepts from a given academic document using a reference ontology, and provide an interactive visualization system to identify concepts of interest and use them to customize the summary. I also present insights from a user study on how well users are able to follow summarization guidelines when using ConceptEVA. Finally, this chapter includes recommendations for future development and analysis of mixed-initiative summarization systems such as maintaining the user's mental map of the original document by preserving its layout, allowing users to create custom groupings of concepts that will help them add critique to the summary, and minimizing interactive latency for a more fluid interface.

5.2 Related Work

ConceptEVA introduces a human-in-the loop, mixed-initiative approach to evaluate and customize document summary generation. In this section, we review prior work in the domains of summary generation, summary evaluation, and text and document visualization on which we build to create ConceptEVA.

5.2.1 Summary Evaluation

Summary evaluation techniques can be divided into two main categories: intrinsic [145] and extrinsic [252]. Intrinsic evaluation methods evaluate a summary based on how well its information matches the information in a reference summary, which is typically human-generated. Some examples of intrinsic evaluation of summarization include ROUGE [209] and BERTScore [398]. Bommasani and Cardie [31] propose separate intrinsic scores for compression, topic similarity, abstractivity, redundancy, and semantic coherence. In contrast, extrinsic evaluation methods evaluate summaries based on their suitability to specific tasks such as following instructions, assessing topic relevance, or answering questions [91, 155, 252]. In extrinsic approaches, humans subjects are asked to use different summaries to perform a task and uses metrics for their performance—such as completion time and success rate—to evaluate the summaries.

My work incorporates the principles behind extrinsic summary evaluation methods. By effectively revealing and comparing the important concepts in a document and its summary, readers can gain confidence in a qualified summary by confirming that it includes all interested concepts, or see which concepts are missing in a "poor" summary.

5.2.2 Summary Generation and Customization

Advances in deep learning and AI has made the automatic generation of good-quality summaries for long document text possible, featured by the success of Transformers [363] with its innovative architecture and attention mechanism. Unsupervised pre-training methods—Masked LM (MLM) and Next Sentence Prediction (NSP)—proposed by Devlin et al. [85] for their Bidirectional Encoder Representations from Transformers (BERT) enables modeling natural language on a huge corpus, and then fine tuning the model on downstream tasks like summarization. Inspired by BERT, other researchers [200, 284, 285, 397] propose different pre-training methods and improve the quality of summarization. For instance, Li et al. [204] propose a multi-task training framework for text summarization that trains a binary classifier to identify sentence keywords that guides summary generation by mixing encoded sentence and keyword signal using dual attention and co-selective gates. Wu et al. [382] use a top-down approach to recursively summarize long articles like books. In this work, we use the Longformer Encoder Decoder (LED) [24] for long scientific document summarization, which turns a full attention mechanism—computing relationships between every pair of words in the document—to a local attention mechanism—computing relationships between a more "local window" of limited words that precede and succeed any given word. This has two benefits: faster computation and lower memory usage, which makes it more capable of processing longer documents without a significant drop in the summary quality.

For the summary customization task, most existing NLP techniques utilizes memory to adjust the auto-regressive language model's output distribution such that the models can retrieve external information given the input prompt. Nearest-Neighbour Language Models [181] merge the retrieved information into the output distribution

and boost up the language model's perplexity without training. Borgeaud et al. [32] show that by incorporating a large-scale explicit memory bank, a smaller language model can achieve performance comparable to models like GPT-3 with 25 times more parameters, and can update its memory bank without additional training. Inspired by these methods, we apply Johnson's method [168] to retrieve the k-nearest sentences for each sentence relevant to a chosen concept, and we customize summaries given these sentences as context.

Besides fully automated approaches, there are also semi-automatic solutions that incorporate humans in the loop. Post-editing [192, 247] is a common semi-automatic approach for summarizing text, which allows humans to edit AI-generated summaries to ensure accurate and high-quality summarization. Compared to post-editing, ConceptEVA's approach better exploits human-AI collaboration and iteratively improves the summary by leveraging such collaboration. In contrast to post-editing which only allow human to edit the summary at the end, ConceptEVAsupports users to iteratively evaluate and refine the summary by inputting their intention on what should be summarized to the AI models. In ConceptEVA, users can also edit the AI-generated summary. But instead of direct manual editing, ConceptEVAleverages AI models to provide aids, such as connections to the concepts, and suggestions for paraphrasing.

5.2.3 Interactive Visual Analysis for Text Data

This work involves designing interactive visualizations of word embedding and thematic infographics to facilitate summary evaluation and customization. Visualization of word embeddings [152, 214, 330] has been used for supporting text data analysis, such as selecting synonyms, relating concepts, and predicting contexts. In a different way, thematic visualizations are useful for exploring document and conversational texts. For instance, ConToVi [95] uses a dust-and-magnet metaphor [331] to visualize the placement of conversational turns (dust) in relation to a set of topics (magnets). NEREx [96] provides a thematic visualization of multi-party conversations by extracting and categorizing named entities from transcripts. The conversation is then visualized as connected nodes in a network diagram, allowing a visual, thematic exploration of

the conversation. TalkTraces [53] uses a combination of topic modeling and word embeddings to visualize a meeting's conversation turns in real time against a planned agenda and the topics discussed in prior meeting(s). VizByWiki [208] automatically links contextually relevant data visualizations retrieved from the internet to enrich new articles. Kim et al. [182] introduced an interactive document reader that automatically references to corresponding tables and/or table cells. All these works exploited visualizations to provide contexts or additional information for helping readers to better comprehend text contents.

The application of concept-based clustering is not limited to text analysis: Park et al. [273] cluster neurons in deep neural networks based on the concepts they detect in images, and in addition create a vector space that embeds neurons that detect co-occurring concepts in close proximity to each other. Berger et al. [25] propose cite2vec, a visual exploration of document collections using a visualization approach that groups documents based on the context in which they are cited in other documents, creating a combined document and word embedding. Closest to our own proposed work is VitaLITy [254], an interactive system that aids academic literature review by providing a mechanism for serendipitously discovering literature related to a topic or article of interest. VitaLITy uses a specialized transformer model [70] to aid academic literature recommendations that use additional data such as citations. These recommendations are presented via a 2-D projection of the document collection embeddings generated from the transformer model. This work also uses word embeddings to project a view of relevant concepts onto a 2D space, but is different from VitaLITy in the purpose: our focus is on interactively exploring the concepts of a generated summary as well as generating summaries that emphasize concepts of interest within an academic publication.

In this work, we use visualization of word embeddings to provide overviews of all the important concepts in a document and identify which concepts are missing in the summary for evaluation. Thematic infographics is used in the visualization of word embedding to show the details and occurrences of a concept in both the document and summary for comparison.

5.3 Design Requirements

In order to better understand the different requirements and motivations when summarizing an academic article, we conducted a preliminary survey of 8 higher education professionals: one professor, 4 associate professors, and 3 assistant professors (7 male, 1 female, all between 25–44 years of age). The survey covered open-ended questions concerning how they motivated and guided students' paper summaries, how they evaluated such summaries, and what they consider to be a good summary and why.

Based on the experts' responses, we grouped their remarks and suggestions under three categories: ***process***, representing approaches they use or suggest students to follow in order to summarize an academic document; ***content***, representing what should be included in the summary; ***requirements***, representing attributes that make for a "good" summary. Each remark or statement below is suffixed with a count showing the number of experts who shared the corresponding opinion.

- Process: Approaches to follow when summarizing.
  (1) Prioritize referring to abstract, conclusion, introduction, and title (7 experts).
  (2) Use the abstract & introduction as a "backbone" for the summary (1 expert).
  (3) Familiarize oneself with background and context, then identify strengths & weaknesses (1 expert).
  (4) Find parts of the paper relevant to one's context or interest and focus on them (1 expert).
- Content: What the summary should include.
  (1) An Explanation of what the paper is about and what its contributions are (5 experts).
  (2) The major ideas of the proposed solution and its difference from prior work (3 experts).
  (3) The results generated by the solution, and how they address the problem/research question (3 experts).
  (4) The problem addressed by the paper and the research questions it answers (2 experts).

(5) An outline of existing approaches to address the research question or problem, their advantages and limitations, and the challenges (2 experts).

(6) The advantages/disadvantages of the solution and the strengths/ weaknesses of the paper (2 experts).

- Requirements

(1) The summary should have an indication that the summarizer has not simply paraphrased the paper but also thought about and understood the underlying ideas (3 experts).

(2) The summary should show reflection on the ideas and discuss implications for practice/research. (3 experts)

(3) The summary should include a figure if possible (2 experts).

(4) The summary should have a clear structure & emphasis (2 experts)

(5) The summary should be specific and provide details, paraphrasing where necessary and quoting from the paper where necessary (1 expert).

While the above responses are relevant for manual summarization, we also examined existing approaches of evaluating automated summarization techniques, such as fluency, saliency, novelty, and coherence [344]. Saliency is an especially complex issue as saliency of a given summary may vary across readers depending on each reader's background and research focus. Based on the responses and on prior work on automated summarization, we synthesized the following requirements that we prioritize for mixed-initiative approaches that help the user evaluate and customize summaries of scientific articles:

R1 Accuracy Evaluation: The technique should help the user efficiently verify whether a summary accurately reflects the content of the original document based on the criteria established by the user (see R4: Flexibility below). This requirement is synthesized from participant responses categorized under *"criteria"* and *"structure"*.

R2 Provenance Evaluation: The technique should show direct or indirect contributors to a summary to help the user verify whether the summary reflects the

structure and key components of the original document. This includes the parts of the original document—a research article in this case—that contribute to the summary. It also includes external references (see R3: Contextualizations) that influence parts of the summary. This requirement is synthesized from responses under *"topics"*, *"structure"*, and *"strategies"*.

R3 Contextualization: The technique should be able to provide some context in which the work presented in the paper exits. Such a context includes the contribution of the work, as well as the significance of the work, its strengths, weaknesses and so on. This can include information presented within the paper itself but should not be restricted to it. This requirement is based on the participant responses under *"criteria"*.

R4 Flexibility: The technique should be flexible enough to change the summaries based on the priority of the user. For instance, the summary may focus on the relevance of the paper to a concept of interest to the user. Alternatively, the summary may also be one that examines the paper's contributions, approach, and methods—or any combination thereof. The requirement is based on participant responses under *"topics"* and *"strategies'*.

5.4 Methodology

In ConceptEVA, we support summary evaluation and customization by empowering the exploratory visual analysis (EVA) with multiple natural language processing (NLP) techniques. In this section, we first introduce the data processing and visual analysis framework of ConceptEVA, then describe the major NLP techniques backing the functionalities of the system.

5.4.1 Framework Overview

ConceptEVA leverages knowledge graphs, NLP, and EVA techniques to facilitate summary evaluation and customization for academic document readers. We bridge the original document and the summary with a concept view visualizing all of the concepts identified from the document. As shown in Fig. 5.2, we start by extracting concepts from

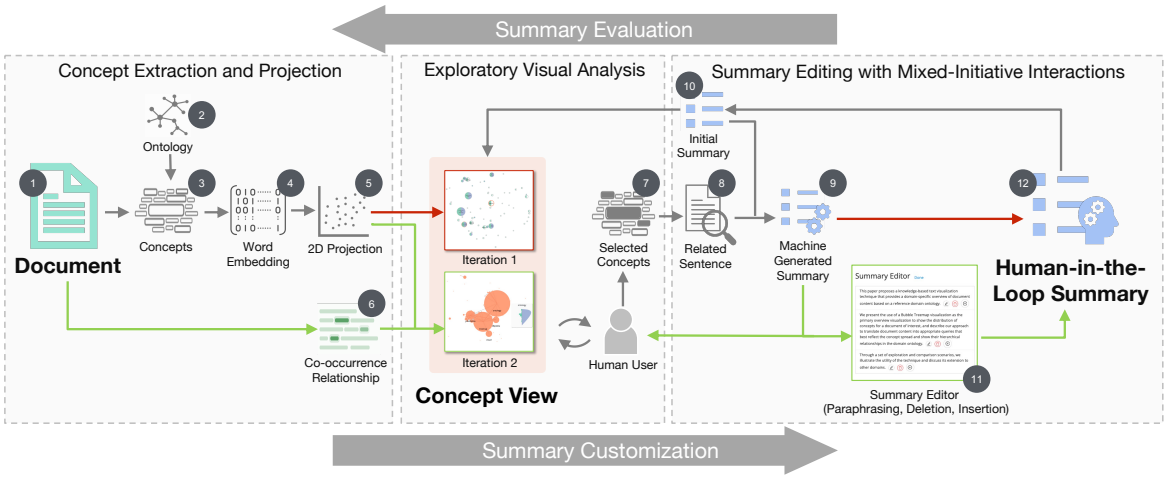


Figure 5.2: The framework of ConceptEVA. The core idea is to bridge the document and the summary with a concept view. In iteration 1, the concept view shows an embedding-based layout that allows users to select concepts to include in the machine-generated customized summary (red boxes & arrows). In iteration 2, the concept view also includes the co-occurrence information in a force-directed layout, and a summary editor with mixed-initiative interactions is added (green boxes & arrows). In both iterations, the user can repeat the human-in-the-loop summary customization for multiple rounds till they are satisfied with the result.

an academic document according to a reference ontology, converting them into text embeddings and projecting them onto a two-dimensional space (Sec. 5.4.1.1). After that, we present the semantic and contextual information of the concepts in an interactive visual interface that supports flexible concept exploration and customized concept(s) prioritizing (Sec. 5.4.1.2). Finally, we provide an interactive summary editor to facilitate dedicated refinement of a new version of the summary we generated according to the user-specified concepts of interest (Sec. 5.4.1.3). In this way, we help the users evaluate the quality of an AI-generated summary and see how well it addresses the readers' focus of interest in the paper, as well as support them customizing the summary to alter their specific requirements if the automated one is not satisfied enough.

5.4.1.1 Concept Extraction and Projection

In order to effectively extract the key concepts from a large body of texts, knowledge graphs, such as DBpedia [16], Freebase [30], and Wikitology [286], can be used to look up established concepts in specific domains. We use DBpedia-Spotlight [238] to extract concepts and rank their importance by term frequency. We then visually highlight concepts to show which ones are included or missed in the AI-generated or customized

summary. To vectorize these concepts, ConceptEVA leverages text embeddings to represent concepts, sentences, and descriptions of the concepts as high-dimensional vectors. Two-dimensional projections of these "concept vectors" are computed using dimensionality reduction techniques, such as PCA [352], t-SNE [359], or UMAP [234]. Semantically similar concepts are placed closer together in the projections, while different concepts are placed farther apart.

5.4.1.2 Exploratory Visual Analysis

To allow readers to explore and reason about the concepts, ConceptEVA provides interactive visualizations to help trace these concepts back to the source document text as well as to the generated summary. A visual representation (see Sec. 5.5 for details) is designed to show the importance of the concepts and help the user compare their occurrences in the document text and the summary. Readers can use ConceptEVA's interactive visual interface to explore and understand each concept, as well as selecting concepts that are relevant to their research interests. The selected concepts are used to recompute the importance and relevance of each concept in the high-dimensional embedding and recreate the projection, allowing the readers to "steer" the exploration.

5.4.1.3 Summary Editing with Mixed-Initiative Interactions

While generating a good summary that can satisfy the user's needs and interests cannot solely rely on NLP techniques, ConceptEVA provides a set of mixed-initiative interactions for quickly customizing and editing an AI-generated summary. From the user interface, users can easily select which concepts in the document are important or match their interests. If the generated summary did not provide enough context or description of these concepts, the user can indicate where in the summary that they want to add a sentence about a particular concept, then ConceptEVA will immediately generate a list of sentences that describe that concept for the user to choose. In addition, ConceptEVA allows users to paraphrase any of the sentences based on its NLP models.

5.4.2 Natural Language Processing: Multi-Task Longformer Encoder Decoder

As shown in Fig. 5.2, ConceptEVA uses several NLP techniques at various stages of summary generation and customization. At the center of these techniques is a multi-task Longformer Encoder Decoder (LED) [24] that we develop for iteration 2. We describe in this section the motivation to use LED and its functions at specific stages in summary generation and customization.

In the first iteration of ConceptEVA, we developed a a hierarchical summarization method with BERT Extractive Summarizer [239] and a Pegasus abstractive summarizer [397] for summary generation and customization of long documents. However, this approach could easily incur high interaction latency caused by sentence clustering and iterative summarization of long documents. To alleviate these issues, we develop for the second iteration a multi-task Longformer Encoder Decoder (LED) [24], capable of processing longer documents. In addition, we take advantage of weight sharing, i.e., every task shares weights on the common parts of the network's memory, thus optimizing the time and space efficiency of ConceptEVA and speeding up the system's responses to human input.

Our multi-task LED is employed in ConceptEVA for four functionalities: scientific document summarization, paraphrasing, semantic text encoding, and summary customization (see Fig. 5.3). We describe these functionalities below.

Scientific Document Summarization: The LED model was trained on the ArXiv dataset of scientific papers [69]. Due to its local self-attention mechanism, the memory complexity of LED grows linearly, making it capable of handling up to 16384 tokens, which is typically long enough for handling academic papers. These factors render the LED suitable for generating summaries of academic papers. These automatically-generated summaries (see item '10' in Fig. 5.2) act as a starting point for users to evaluate and customize upon according to their interests.

Text Paraphrasing: One of the functions in ConceptEVA's mixed-initiative interactions is the ability to paraphrase text, or specifically, generate alternative summaries

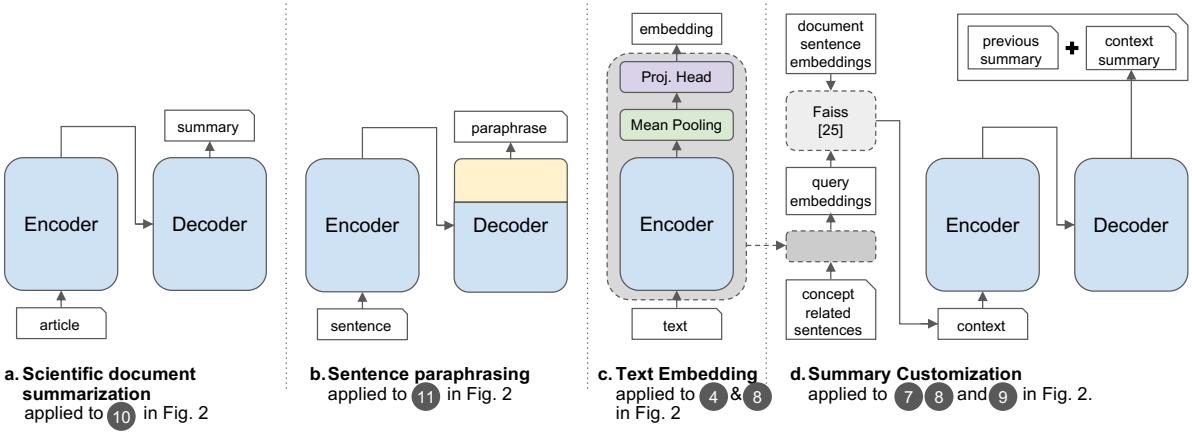


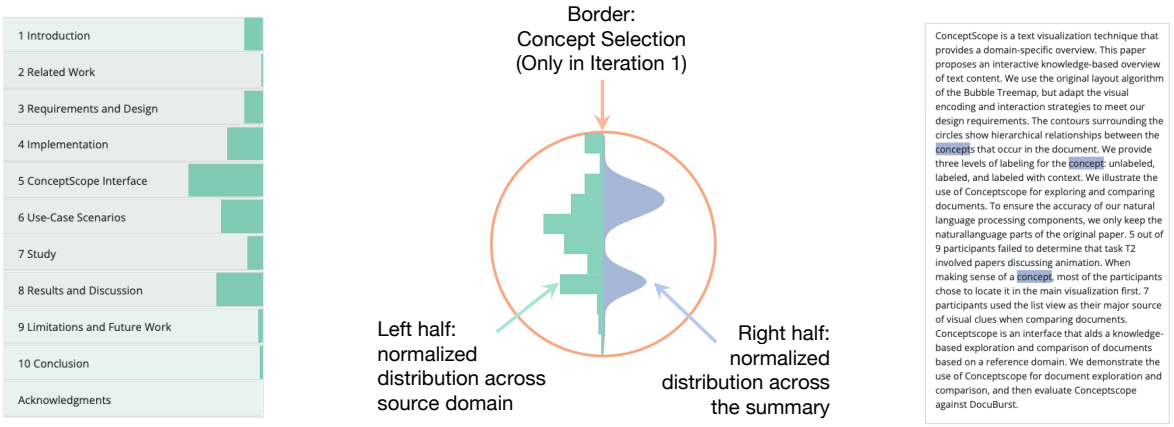
Figure 5.3: ConceptEVA uses a multi-task LED model [24] to help generate, evaluate, and customize summaries. Specifically, LED performs four functions, shown above as subfigures a–d with detailed explanations in Sec. 5.4.2. The text below each subfigure indicates the corresponding function in Fig. 5.2 for which the model is used. In each subfigure, the blue rounded boxes represent the weights from the LED trained for summarizing scientific papers and shared across all tasks. The yellow, purple, and green rounded boxes represent fine-tuned layers for downstream tasks. The functions are: (a) **Scientific document summarization:** The LED's training data, local self-attention mechanism, and high memory complexity make it suitable to summarize academic papers. (b) **Sentence paraphrasing:** We fine tune the last two decoder layers (shown in yellow) with a set of "paraphrasing datasets"—datasets that contain multiple paraphrases of a given set of sentences. This helps in generating alternative sentences for a given sentence when editing a summary. (c) **Text embedding:** To generate the concept layout (see Fig. 5.1-2) and fetch relavant context for summary customization, we compute text embeddings—vector representations of concepts or sentences in a high-dimensional space. This is done by adding a mean pooling layer (green) and a projection head (purple) to the encoder and fine-tuning it (see Sec. 5.4.2 for details). (d) **Summary customization**. Pre-computed embeddings of every sentence in the source document are queried using vector representations—retrieved from the text embedding shown in (c)—of user-selected concepts. Nearest sentences are appended to provide 'context' for the selected concepts, and then summarised. The resulting summarized sentences are appended to the existing summary (see details in Sec. 5.4.2).

for a selected sentence. To achieve this capability, we fine tune the pre-trained model on relatively small datasets with small learning rates. We "freeze all the layers" of the model, i.e., we keep all model weights the same during training except for the last two decoder layers. The decoder takes a sequence of tokens as the input and generates the next token based on its weights. We train these two decoder layers on a dataset that contains 147,883 sentence pairs, with each pair containing two alternative paraphrases of one sentence (Fig. 5.3b). We build this dataset by merging three other datasets: PAWS [399], MRPC from GLUE [367], and TaPaCo [310]. Once fine-tuned, this model is capable of taking as input one sentence and providing a paraphrased sentence as an

output. In item '11' in Fig. 5.2, this model is accessed via the summary editor when the user opts for automated paraphrasing of a selected sentence.

Text Embedding: To generate the concept layout (see Fig. 5.1-2) and fetch relevant context for summary customization, text embeddings—representing the relationships between concepts or sentences in a high-dimensional space—need to be computed. To compute sentence embeddings, we follow the siamese network architecture from SentenceBERT [292], an approach to generate sentence embeddings, i.e., vector representations of sentences that preserve semantic relationships. We add a 'mean pooling layer'—a function that averages the embeddings of input tokens—and a 'projection head'—a function that computes a high-dimensional space that captures semantic similarities between all sentences—on the LED's encoder (Fig. 5.3c). We then fine-tune the encoder for learning meaningful sentence embeddings by freezing all layers of the encoder and only training on the projection head. For the training data, we once again follow SentenceBERT: we combine the SNLI [36] and MultiNLI [379] datasets, and format each data sample as a triplet of an 'anchor sentence', a 'positive sentence', and a 'negative sentence'. The training involves fine-tuning the embedding such that in each triplet, the positive sentence ends up closer to the anchor sentence than the negative sentence. We also follow data augmentation approaches. inspired by those followed in SentenceBERT [292]. The resulting model is used in two main functions of ConceptEVA: generation of the "concept view" (Fig. 5.2), the "focus-on" function (Sec. 5.5.2), and subsequent summary customization (see items '7' and '8' in Fig. 5.2).

Summary Customization: ConceptEVA customizes a generated summary by updating it to include concepts of interest selected by the user. To achieve this, we pre-compute embeddings for every sentence in the source document. When a user selects a concept or concepts of interest, we retrieve corresponding text embeddings using the model described in the previous paragraph. We then use these embeddings as 'queries' to search for sentences in the pre-computed embeddings that are closest to the query vectors (see Fig. 5.3). We apply Faiss [168]—a similarity search library of dense vectors in large scale—to implement this approach. The nearest sentences are



(a) Document Sections (b) Concept *Glyph* (c) Summary

Figure 5.4: The *concept glyph* extends the concept circle to support the in-place comparison of concept distribution between the document and the summary. This glyph is shown for all dominant concepts in iteration 1 and in a floating tooltip upon request in iteration 2.

concatenated in the order of their appearance in the original document and included in the input to the summarizer as 'context' for the selected concepts. The resulting, newly-summarized sentences are then appended into the previously-generated summary. In this form of summary customization, new concepts add to the existing summary but do not result in the erasure of parts of the existing summmary. The summary editor provides the option for the user to manually delete the sentences.

5.5 Interface Design

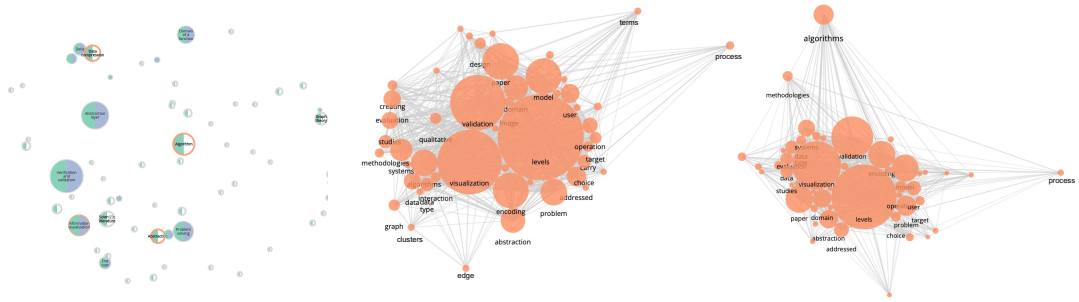
The ConceptEVA interface (Fig. 2.2) consists of three main panels: a document view on the left (with green header & accents) that collapses into a section-wise overview, a summary view (blue header & accents) on the right displaying the generated summary and associated metadata, and a central concept view (orange header & accents) showing the relative dominance and associations between the concepts found in the document. Additional controls for visualizing and filtering the concepts are also provided on top of the concept view. The interface design has gone through two iterations of development, incorporating feedback and insights from the expert review (Sec. 5.6). We detail the visualization and interaction design choices of the final version of the system and the underlying rationale in this section.

5.5.1 Concept View: Document-Summary Relations

In the concept view, we provide an overview of the document-summary relation from the perspective of concepts. We represent each of the concepts occurring in the documents as a node—a "concept circle"—the size of which shows the dominance of the concept in the source document. User-specific metrics of dominance, such as "frequency" and "tf-idf" are available for the user to choose.

To convey information about the structure of the document and of the summary (**R2**), we incorporate the user's orientation to the interface—the document on the left and summary on the right—into the concept view to represent concepts that are present in the document and concepts present in both the document and the summary. We design the *concept glyph*—a pair of histograms representing the distribution of the concept across the source document and the summary respectively (see Fig. 5.4). The histograms are oriented vertically and share a common axis. This way, the histogram on the left indicates the source document and the curved line on the right (histogram smoothed with a kernel density estimation) represents the summary. The number of bins on the histogram on the left matches the number of sections in the source document, while the right one maps to the number of sentences in the summary. For instance, the concept "prototype" is missing in the summary shown in Fig. 2.2 because the right half of the glyph is missing. To further reinforce this connection between the histogram and the document view, we create an echo of the histogram overlaid on top of the section headers (Fig. 5.4-a). This allows the user to identify the sections of the document in which the concept is most dominant, and examine the content of those sections closely if needed.

When determining the two-dimensional(2D) layout of these concepts on the concept view and the amount of information to reveal for each of them, we started with an embedding-based layout in iteration 1 where the *concept glyph* of every concept were displayed and distributed according to the text embedding (Fig. 5.5-a). While this layout was designed to help the user efficiently compare the occurrence of concepts in the original document against those in the summary, the expert review results (Sec. 5.6)



(a) Concept View in Iteration 1

(b) Default Layout of Concept View in Iteration 2

(c) "Focus-on" Layout of Concept View in Iteration 2

Figure 5.5: A comparison of the embedding-based layout in iteration 1 and the context-augmented latyout in iteration 2 for the concept view. All three figures show 80% of the concepts from the paper [250]. The circle size represents frequency (The size scale and ontology query parameters are slightly different between iteration 1 and 2). The "focus-on" layout in (c) focuses on the concept "algorithm".

indicated that showing such a comparison for all the concepts in one visualization was too overwhelming to the users. To reduce such perception load, we shifted to a more intuitive visualization design in iteration 2 where the visual representation of the concepts were simplified to solid circles (Fig. 5.5-b) in a force-directed layout. The coordinates of these circles are initialized by a 2D projection of the concepts' semantic word embedding and adjusted by links representing the co-occurrence relationship of two concepts in the same sentence per experts' request for more co-occurrence information support. In this way, we created a context-augmented layout with the coordinates of each concept influenced by both its semantic meaning and its co-occurrence relationship with the other concepts in the specific academic document (**R3**). For instance, Fig. 5.1-2 shows that the concepts "organ" and "prototype" are semantically remote but co-occur frequently in [258], while aligns with the fact of this document. Our context-augmented layout could capture such document-specific concept co-locations and adapt the initial text embedding in concept view to reflect the document context. To efficiently support the user to evaluate the summary quality from the perspective of concept appearance, we move the *concept glyph* with detailed document-summary information for each concept to a tooltip which can be triggered by hovering in iteration 2. This provides an effective overview and detail-on-demand exploration of the concepts in a document using interactive visual analysis.

5.5.2 Summary Evaluation

To facilitate the users to get an intuition about concepts from the document that are included in the summary compared the the concepts excluded from the summary (**R1**), we designed the *concept glyphs* (Fig. 5.1-2e) as described in Sec.5.5.1. Users can quickly filter out all but the "important" concepts, and then compare their distribution and context in the document and in the summary using the *concept glyphs* and the linked view to the document on the left (**R3**). To cater to user-specific analysis requirements (**R4**), we allow users to (1) choose the criteria (frequency or tf-idf) by which concepts should be considered "important" (Figure 5.1-2a), (2) choose the dimensionality reduction method (PCA, tSNE, or UMAP) to project the concepts (Figure 5.1-2b and 3), and filter them to only show the top K percent of concepts based on ConceptEVA's importance metric (Figure 5.1-2c).

Inspired by the experts' attempt to locate concepts with the "focus-on" function and their significant interest in it, we enhanced the "focus-on" function in iteration 2 to allow the user to switch perspectives and evaluate how well the current version of the summary addresses their specific areas of interest (Figure 5.5-c). When the user triggers the "focus-on" function, they will be able to select from full list of the concepts sorted by their appearance frequency in the original document (Figure 5.1-2d). Users can select one or multiple concepts based on their research interests and trigger a corresponding update of the concept view layout. The concept they choose to focus on will "float to the top" of the concept view, i.e., move to the top of the view, and the rest of the concept will "sink" to the bottom, with semantically or contextual-wise more relevant concepts pulled higher towards the top and less relevant concepts pushed lower towards the bottom. Meanwhile, the horizontal layout remains to reflect the concepts semantic and contextual distance determined by the user-chosen projection method. For instance, the layout in Fig. 5.5-c was focused on the concept "algorithm". We can see the related concepts including "methodologies", "validation", and "systems" are also pulled upwards. Meanwhile, the layout of the remaining concepts Fig. 5.5-b is locally maintained, continuing to reflect their semantic and contextual closeness

in the document. This will further facilitate the concept selection and inform the customization task described in Sec. 5.5.3.

5.5.3 Summary Customization

Reflecting on the requirements we collect for a "good" summary (Sec 2.3), we approach summary customization in two ways: at a concept level, we see summary customization as determining what concepts are included when generating the summary, while at a structural level, we see it as inserting, reordering, and rewriting content. Users can achieve the concept-level summarization by selecting a group of concepts from the concept view to prioritize for the next version of the summary. Based on user selection, the summarizer extracts relevant sentences from the document as described in Sec. 5.4.2 and inputs them to the summarization pipeline for a customized summary that better addresses the concepts of interest.

The AI-generated summarization approach focuses more on the content than the flow of the summary, and was seen in the expert review as compromising the logical and narrative connection from one sentence to the next (see Sec. 5.6 for details). To address these concerns about the summary quality, we extended the interactions supported in ConceptEVA with an interactive summary editor to facilitate better human-AI collaboration in iteration 2. With the AI-generated summary as a starting point, the summary editor (Figure 5.1-3) helps users iteratively customize or extend the summary (**R1** & **R2**) by: (1) choosing from a list of candidate sentences for all user-selected concepts categorized by concept name, and inserting them into the summary, (2) updating a particular sentence in the summary with automatically paraphrased sentences generated with the paraphrasing model in Sec. 5.4.2 (Figure 5.1-3a), and (3) interactively editing, reordering, or deleting any sentences. In this way, a human-in-the-loop summary will be generated as the final output of the summary customization process in which user knowledge and judgments are effectively cooperated with the NLP techniques described in Section 5.4.2.

5.6 Expert Review of Iteration 1

Iteration 1 of ConceptEVA was evaluated through expert review with three participants (2 male, 1 female). Given our prototype was backed with a NLP model more suited for scientific document analysis, we invited three experts with Ph.D. degrees in computer science with InfoVis as their research focus. Participant details are listed below, with years of experience in reading/reviewing academic papers included in parentheses.

- **E1:** software engineer (5–10 years).
- **E2:** senior applied scientist and former academic (10–20 years).
- **E3:** data scientist (5 to 10 years).

The review was conducted online via a video conference setting. Participants were first introduced to ConceptEVA's functions and features and given trial tasks with a test dataset to familiarize them with the interface.

Participants then used ConceptEVA to finish two open-ended tasks while following a concurrent think-aloud protocol: (1) verify the auto-generated summary for a given document, and (2) generate a customized summary according to a set of requirements provided to them. Since the participants were experienced researchers in infovis, we also collected their feedback and recommendations on the system as suggestions to incorporate into iteration 2. Iteration 1 was received positively in general, especially idea of evaluating a document summary by examining the concepts (E1, E2, E3), context and support views to compare the document and the summary (E2, E3), but the quality of the generated summary was not considered sufficient (E1, E2, E3). Specific feedback is listed as follows:

- *Concept extraction & separation:* Concept identification through fuzzy matching between document terms and the reference ontology sometimes produced results that the experts (E1, E2, E3) found confusing. Iteration 1's implementation of the "focus-on" interaction was also not deemed helpful likely due to the issues concerning the fuzzy matching (E1, E2, E3), though all experts expressed considerable interest and pointed out potential ways for improvement. E1 and E2 also expressed that they expected a better-functioned "focus-on" tool with more

intuitive interaction. E3 also suggested providing concept searching functions, showing the frequency of the concepts, and sorting the searching list accordingly.

- *Information support:* The visual representation of the concepts and the way they supported the comparison of the summary against the document was deemed helpful (E2, E3). Showing co-occurrence information of concepts was recommended (E1, E2, E3).
- *Summary quality and presentation:* An initial paragraph-like summary shown to E1 & E2 was deemed to not have a logical flow, while a bullet-point format change with E3 was received well. However, E3 was uncertain on how well they could "trust" the summary if it were of an unfamiliar paper, and recommended showing additional information to increase the user's confidence in the summary.

5.7 User Study of Iteration 2

Lessons learned from the expert review helped focus the redesign of ConceptEVA and focus its evaluation through tasks that reflect how a researcher may approach summarizing an academic paper. Specifically, we decided to focus our study on whether and how a participant is able to generate a summary of a paper with which they are familiar using ConceptEVA such that the summary is relevant to their research interests.

While comparing the use of ConceptEVA with an existing summarization tool would be ideal, to our knowledge there is no existing summarization tool designed for research documents. We thus chose human-generated summaries by each participant as the baseline for that participant. While this means there is no "standard" baseline across all participants, this approach gives us better ecological validity as each participant would generate a summary that is relevant to their own interests and research contexts. Therefore, the current baseline for researchers would be to generate a summary by themselves—unaided by other tools. This would serve two purposes. Firstly, by generating their own summary manually, they gain familiarity with the document and are able to use ConceptEVA as a tool to refresh their memory, navigate the concepts relevant to the document, and be able to compare the summary they generate using

ConceptEVA against their own manually-generated summary. Secondly, the process serves to emphasize our idea that ConceptEVA is *not* intended as a replacement for reading the document; it is intended to augment the way the document is explored.

This necessitated a study with a within-subjects component where each participant first generated a summary manually before attempting the same task on ConceptEVA. For the same reason, there was no counterbalancing: asking all participants to perform the manual summarization task first allowed us to ensure they were familiar with the document before they used ConceptEVA. It also allowed participants to critically examine the extent to which they could create a summary that was relevant to their own interest in the document. We used two test papers [258, 347], one for six participants who participated in our study.

5.7.1 Participants

We recruited 12 participants (4 female, 8 male, aged 25–44 years), comprising 10 Ph.D. students, 1 university faculty, and 1 research engineer from a technology company. Seven participants reported they had been actively reading academic papers for 5-10 years, and the remaining five reported less than 5 years. And 10 participants reported they had written a summary/abstract/short description for an academic paper more than 10 times before the study, and the remaining two did it for 3-10 times. Two of the 12 participants reported themselves as native English speakers.

5.7.2 Experimental Setup

We conducted the study remotely considering the varied geographical locations of the participants and a safety measures surrounding the uncertain conditions of COVID-19. Instructions for the offline study task **T1** were shared with participants no less than 12 hours before the online study session began. For the online study session, the participants were asked to access ConceptEVAfrom a remote server and participate in the study with their own machine and external devices. Six participants used the Chrome browser with the Windows operating system, four used Chrome with MacOS, and the remaining two used the Safari browser with MacOS for the tasks. The setup,

tasks, and durations were decided based on a pilot study with three participants: one native and two non-native English speakers.

We asked the participants to follow the "think aloud" protocol and audio- and video-recorded them during the task. Each participant received a $10 Amazon gift card as a compensation for their participation.

5.7.3 Summarization Guidelines

Based on findings from our survey of research practitioners explained in Sec. 2.3, we constructed a set of guidelines for participants to follow when generating a summary manually or using ConceptEVA. The guidelines were presented in the form of the following list of questions that participants could try and answer in their summary.

- **G1 Content.** What is the paper about? What are the contributions?
- **G2 Approach.** If the paper addresses a problem, how does it do it?
- **G3 Comparison.** If the paper addresses a problem, how does its approach compare to existing approaches to address the same problem?
- **G4 Insights.** What insights does the paper offer from its analysis or evaluation of the approach?
- **G5 Critique.** What are the strengths and weaknesses of the approach?
- **G6 Implications.** What are the implications of the work to your own interests and/or research?

We made it clear to participants that they were free to choose some, all, or even none of the guidelines below when generating the summary. In the procedure below, we would ask the participants which of the guidelines they followed for each summarization process: manual and using ConceptEVA.

5.7.4 Procedure

Each participant was provided with a research paper a few days in advance of the scheduled session with the study moderator, along with the guidelines listed in Sec.5.7.3. Each participant was then assigned the following tasks:

T1: Manual summarization.

- We asked participants to read the paper and manually generate a summary between a minimum of 100 and a maximum of 150 words reflecting what they found interesting in the paper. This summary was to be sent to the moderator in advance of their scheduled session. This represents the baseline for each participant, indicating the summary they would generate without ConceptEVA. It also ensures that participants read the paper before the start of the study.
- After their summary was received, participants were also asked to fill in a survey relating to their background and demographics. They were also asked to respond on a 7-point Likert scale (one for each guideline in Sec. 5.7.3) the extent to which they followed the guideline.
- Participants were also asked to report on their experience of the summarization task on the NASA TLX scale [149].

T2: Automated summarization.

- Participants were shown the automated summary generated without human intervention and asked to read through it.

T3: Human-in-the-loop summarization.

- Participants were introduced to the ConceptEVA interface and allowed to explore it through mini-tasks that reflected the process they would follow in their main task. This training/exploration session used a paper different from the one used for their tasks.
- Participants were then instructed to generate a summary of the same paper as in T1, following the same prompts and guidelines, but this time using ConceptEVA to explore and focus on concepts of interest and choosing relevant concepts to steer the summary generated. Throughout this exploration participants were instructed to follow a concurrent think-aloud protocol where they verbalized their thinking during their exploration.
- At the end of this process, they responded to a 7-point Likert scale (same as

in T1) showing the extent to which they followed each of the guidelines from Sec. 5.7.3.

- Participants reported on their experience of the summarization task on the NASA TLX scale.

T4: Rating all summaries.

- Participants finally rated on a 7-point Likert scale their satisfaction with (a) their manually-generated summary from T1, (b) ConceptEVA's automated summary with no human intervention from T2, and (c) the summary they generated in T3 using ConceptEVA by focusing on concepts of interest. They were allowed to re-read all three summaries before reporting on their satisfaction. The reason behind choosing "satisfaction" as a metric and for having participants rating their own summaries as opposed to others' summaries are related. Recall that the reason behind proposing ConceptEVA was that different readers of the same research article may emphasize different aspects when generating a summary of the paper. A participant with their own concepts of interest in a given paper would have takeaways that are influenced by these interests, which would in turn be reflected in their summary of the paper. We deemed that it would be less insightful for them to evaluate a summary generated by a different participant with different interests and takeaways. Instead, having the participant examine the summaries they have themselves created through three approaches could potentially reveal more insights into how well the human-in-the-loop approach has worked, as each participant can examine all summaries through the lens of their interest in the paper. For the same reason, "satisfaction" as a measure along with participant responses explaining the reasoning behind the rating allows us a way to understand what aspects of human-in-the-loop summarization are valuable for participants, albeit at the expense of specific insights more objective measures may provide.

The study did not focus on speed or quality of task performance, but on partic-

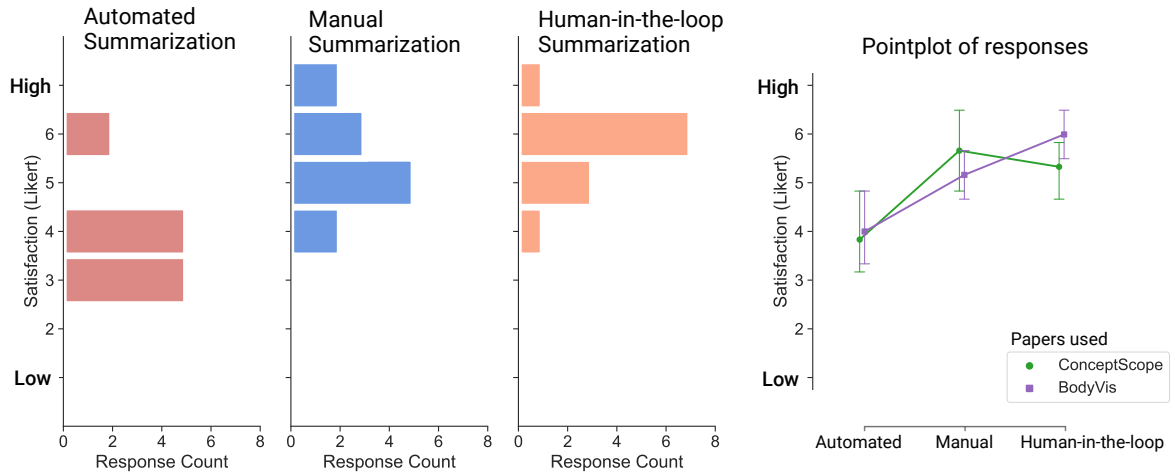


Figure 5.6: Distribution of participant responses on a 7-point Likert scale showing their level of satisfaction with the summaries from the automated approach in task **T2**, manual approach from task **T1**, and the human-in-the-loop approach in task **T3** created with ConceptEVA. The three charts on the left show the distribution as response counts for each summarization approach. The right chart shows average values for each approach for the two papers used in the study, ConceptScope [347] and BodyVis [258], with error bars indicating 95% confidence intervals.

ipants' own satisfaction with their experience and outcome. Thus task times were not restricted, and we did not track the time participants spent on Task 1, only their self-reported experience in writing the summary as described above. Participants in general spent between 60 and 90 minutes on tasks T2–T4.

5.8 Results and Discussion

5.8.1 Summary Satisfaction

As mentioned in Sec. 5.7, we used each participant's manually-generated summary (**T1**) as a unique baseline for that participant. Ten of the 12 participants rated the automated summary (task **T2**) *lower* than the baseline, 8 out of 12 participants rated the summary generated using ConceptEVA's human-in-the-loop approach (task **T3**) *higher* than the baseline (Fig. 5.6). Recall that two papers were used in the study—6 participants summarized ConceptScope [347] and 6 summarized BodyVis [258]. Fig. 5.6 also includes a pointplot showing average ratings split across both papers. While the small participant pool makes it difficult to state with sufficient confidence whether participant satisfaction with the human-in-the-loop summarization using ConceptEVA

is equivalent to their satisfaction with their own manually-generated summary, Fig. 5.6 suggests such an equivalence. In addition, a chi-squared test of independence showed a significant association between summarization approach and summary satisfaction rating, $\chi^2(8) = 23.5, p < 0.01$. On the other hand, a chi-squared test of independence showed no significant association between the paper used and summary satisfaction rating, $\chi^2(4) = 0.87, p = 0.93$. This indicates that the differences seen in Fig. 5.6 are more likely to be due to the summarization approach rather than the paper used in the task.

Participants who gave a higher rating for the human-in-the-loop approach reported being able to locate and focus on concepts more efficiently (P4, P6, P9), and on the content of the summary itself (P7). P7 observed that "*the contribution of this paper, was also well described in the* (human-in-the-loop generated) *summary.*" Participants who preferred the manual version of their summary to the human-in-the loop approach (P1, P3, P11) explained that they had their own idea of a summary that they wanted the generated version to reflect. For instance, P11 wanted the summary to focus on the paper methodology, so they deleted all sentences from the automated summary, directing the system to pull new sentences from the paper focusing on "visualization", "concept", and "ontology". They proceeded to edit these new sentences based on their recall of the document and even manually wrote some text from scratch. These participants also reported a lower level of trust in the AI component of ConceptEVA through the study.

Participants' level of trust in the generated summary also appeared to be influenced by their confidence in their knowledge of the domains addressed in the paper. For instance, BodyVis [258], one of the papers used in the study, covers domains like participatory design, physiological sensing, and tangible learning, which the participants were relatively unfamiliar with. Their response to the summary generated by ConceptEVA was more positive. P4 reflected that *"in terms of ... describing the (BodyVis) system, maybe the one generated by ConceptEVA is kind of better... In the manually generated summary, although I put my focus there, I didn't do a good job like mentioning it. I don't think if I mentioned it."* P10 noted the automated summary addressed some of their own

omissions: *"In my manual summary. I actually skipped some details, like I didn't really mention ... the feedback from children and the teachers (about BodyVis)."* In contrast, for a topic they were knowledgeable in, participants seemed to prefer their own interpretations and emphases, as P1 states: *"For the papers, if I already know that area, I have a certain expectation of what I need to look at. Then I would still prefer to write the summary by myself."*

In terms of the process, all participants reported being able to follow guidelines G1 (content) and G2 (approach) i.e., they rated themselves above 4 on a 7-point Likert scale. Six out of 12 participants reported being able to follow G4 (insights) and G6 (implications) as shown in Fig. 5.7. Participant ratings on being able to follow G3 (comparison) and G5 (critique) were skewed heavily toward the lower end of the scale. Participants P4 and P8 found it the most difficult to address these two guidelines, and they had a common approach: they attempted to find concepts related to "limitations" or "cons" to see the weaknesses reported in the paper itself and found this approach difficult to critique the paper and compare it with existing work. A low chance of success is expected with this approach as it is difficult to critique a paper by only examining the paper without a general sense of the related work. A summary that features such critique is difficult to automate as it would need knowledge as well as critical thinking about related work.

5.8.2 Summarization Experience

When responding to the NASA TLX scale (see Fig. 5.7) and rating their summarization experience, participants described the experience of using ConceptEVA as "*helpful*"(P1, P3, P7, P9, P11), "*useful*"(P1, P4, P6, P8, P9), "*amusing*" (P5) and "*enjoyable*" (P5). Eight participants reported that the concept view provided useful information such as the importance, appearance frequency, and co-occurrences of concepts. P4 and P6 also reported finding the focus-on function helpful to explore relationships with less dominant concepts. "*sometimes a concept is kind of minor...sheltered by those big circles...but by lifting it up you can see all the relation to other concepts. you can also like, and identify it directly.*" (P4).

The glyphs from the earlier iteration that were redesigned to be revealed only on

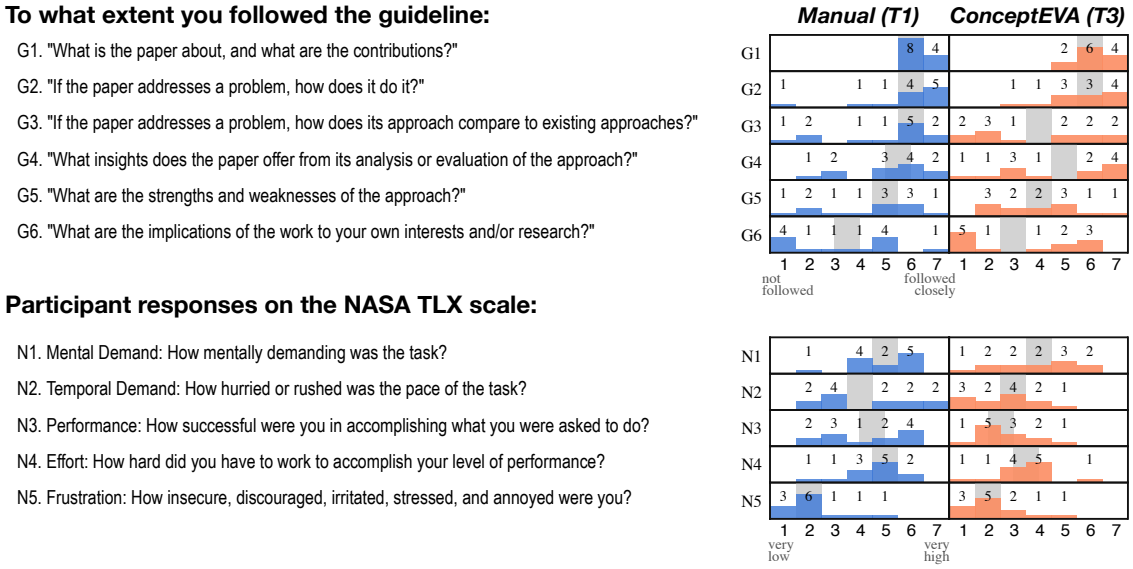


Figure 5.7: Ratings for manually-generated summary in T1 and human-in-the-loop summary in T3. Median ratings are in gray.

detailed inspection were also deemed helpful by 6 participants, indicating perhaps that the glyph in isolation was helpful but several together were distracting. Since ConceptEVAwas implemented for the browser, we also observed participants incorporating built-in browser functionalities such as search, translation (for bilingual/multilingual participants), and grammar checkers.

Participants also expressed their frustration when they "*can't find anything useful about the word they identified*"(P3) or "*lose the full picture of the paper*"(P8). Identifying relevant concepts is a function of the ontology, and a balance between the specificity of domain ontologies and the breadth of a general ontology such as DBpedia. On the other hand, issues related to identifying strengths and weaknesses of the work that may not be explicitly stated in the paper—echoing participant experiences described in Sec. 5.8.1—may be possible to address by additional visualization of document affect and sentiment [188].

5.8.3 Influence Factors on User Experience

When conducting tasks **T3** and **T4**, we observed four dominant factors that appeared to influence participants' use and preferences of certain functionalities in ConceptEVA.

- *Academic reading experience and skill influences exploration.* Participants such as

senior PhD students and faculty/researchers preferred to read the original text of the paper. P5, a graduate student with 5–10 years of experience reading academic papers, said they preferred to read the original text of the paper, but also said that the concept view *"is actually really good with the way my brain is... I just think of words, and then it (the focus-on function) has the words I want. This kind of maps with my thinking, which is very amusing."* In contrast, participants with either less academic experience or from a different domain found direct text reading difficult. For example, P7 thought the paper reading process was"*very overwhelming*" while P8 reported that they "*don't have the full picture of the paper in this way*". They preferred to use the visualizations—the projection view or the Focus-on too—to get a high-level overview, and then "*grab information based on the concept that I'm giving*"(P11). While this is part of the intention behind designing the visualizations (esp. **R3**), a longitudinal study may be needed to explore how ConceptEVA may be used as a way to scaffold students' ability to read and understand academic text. Note that P5, P7, P8, and P11 are all graduate students, but P5 identifies as a native English speaker while the others do not. While this may not be the reason for the difference, it brings up the issue of reading skill, a factor that was not evaluated in the study.

- *Academic writing experience influences summarization.* An extension of the above observation means that participants' academic writing experience would influence how they used ConceptEVA to summarize text. P5 found the workflow afforded by ConceptEVA useful, and that it was "*doing most of the work for me*", such as "*constructing sentences I would put in my paper, or something letting me take what either my problem is or what I'm thinking about looking at the paper, and like merging these things together*". They also appreciated "*the freedom of allowing more editing*" in the summary editing panel (**R4**), and used it to directly edit the summary sentences. P10 reported finding it useful to "*pull out the related sentences categorized by each of the concepts you selected*" (**R3**). Other experienced participants like P11 reported that ConceptEVA "*doesn't encode (sic) their standard of generating the*

summary." Note that P11 is also the participant who heavily edited the generated summary (Sec. 5.8.1).

- *Domain familiarity influences use of ConceptEVA.* Participants' reflections indicated that their knowledge of the domain covered in the document would influence how they would use ConceptEVA. P1 mentioned that "*if I'm reading a machine learning paper or deep learning one that I'm not quite familiar with (the domain)*", they would prefer to use the concept view to "*understand what kind of concepts they (the paper) have*" and would like to see definitions of the concept in ConceptEVA. On the other hand, for documents in their own domain, they said they would *"have a certain expectation of what I need to look at. Then I would still prefer to write the summary by myself."* This was also seen in P8's approach in the study: they were unfamiliar with the paper they were asked to read and requested more information support as they did not have "*a general picture of the paper.*"
- *Mental map of document influences use of visual interface.* Eleven of the 12 participants reported being happy with the visual interface for the summary customization task. While distributing their time to the three panels in ConceptEVAin different ways, 11 out of the 12 participants embraced the visual interface for the summary customization task in our study. Participants preferred different aspects of the interface depending on the way they approached ideas in the paper. P5, quoted earlier in this section, stated how the concept view layout mirrored the way they think. P11, on the other hand, preferred the "paper info" panel to the concept view *"because I can see I know where it (the concept) is (in the paper)."* They even chose to search the concept directly in the PDF version of the paper after briefly exploring the Focus-on function in the concept view panel, explaining that "*it's quite a huge number of information ... it's a little bit hard to draw the connection between the information inside the original paper and the (concept view) exploration panel. That's why I just ignore the exploration panel."* Others found the paper info panel disorienting as it provided a view of the paper that was different from the PDF layout they had initially read, stating, *"I don't have, like the mental map of the*

original pdf. It's gone" (P5), *"Here everything's like um very flat. So I don't know where it is."* (P12), and *''I didn't use this. Yeah, this part was well overwhelming"* (P7).

5.8.4 Limitations and Future Work

One of the issues that came up through the iterations is striking the right balance between the use case scenarios of ConceptEVA, specifically its use to explore a paper as an alternative to reading. Similar "distant reading" approaches in the social sciences have received criticism for being suggested as objective alternatives to close reading, a practice considered integral to scholarship [14]. In our studies, the expert review evaluation for the first iteration of ConceptEVA did not require participants to read the paper in advance. Thus they spent more time using the system to understand the paper content—which was not the main focus of the system—than to generate and evaluate the summary. The study setup following iteration 2 ensured that participants were already familiar with the paper, which allowed them to focus on the summary evaluation and customization tasks. Participant reflections we saw in Sec. 5.8.1 and Sec. 5.8.3 show that participants still used ConceptEVA as a way to check if they missed any important concepts, especially if they were unfamiliar with the domain of the paper. Participant P3 suggested using ConceptEVA as a way to skim through papers so that "*if frequent concepts are not what I care, I can just leave this paper and turn to others.*" On the other hand, comments about the disorienting effect of the paper layout in the paper info panel (see "mental maps" in Sec. 5.8.3) indicates that a better application of ConceptEVA would be toward supporting and summarization and *verification*, rather than exploration. Integral to this approach would be to design a paper information view that preserves the appearance of the PDF view, thus preserving the reader's mental map and allowing them to build upon their close reading of the paper.

The two test papers we chose for the user study were corresponding to the two different conditions—highly interdisciplinary papers spanning at least five domains and relatively typical CHI papers describing the algorithm, user study, and visualization design. Because of the authors' limited knowledge background, we chose two CHI papers in which we had a better understanding and control of the content for our

user study. We will eliminate this limitation by testing ConceptEVA on more diverse papers in the future. Besides, we are aware of the different summarization complexity for papers from different domains [164, 289, 382, 391], but consider it more of an NLP research problem rather than our main focus.

Participants also made suggestions for additional functions and features. The most popular suggestions fell under the category of richer view coordination between the panels. Specifically, participants suggested being able to support concept provenance and filtering within a selected section, or a direct linking between the summary text and the paper information panel. However, this would also mean that ConceptEVA becomes more of an exploration tool providing an alternative to reading the paper rather than a support to summarize a paper, which is a different scope of work altogether, and a requirement that needs closer examination in terms of benefits and pitfalls. On the other hand, other suggestions such as the one by P1 about being able to group concepts into groups relevant to the summary such as "definition", "pipeline", and "preprocessing method". While the groups listed by P1 might work for a data science or data visualization domain, other domains might require entirely different groups than can then be examined to summarize contributions, offer critique, and present other salient ideas. Allowing the user to create custom groups aided by additional NLP approaches like sentiment analysis and topic modeling could help users reflect on and critique the paper, and can be a helpful function to consider integrating into ConceptEVA in a future iteration of this work.

Finally, a limitation of our study include technical issues such as network delays, rendering performance issues, and back-end computations to update concept embedding, sentence paraphrasing, or summary generation itself. These, when they occurred, resulted in latency that influenced participants' experience and potentially their responses to questions like the NASA TLX scale. While the focus of this work is not engineering or optimisation of the system, our future iterations will attempt to cut down performance or networking issues relating to latency.

5.9 Summary

This chapter presents ConceptEVA, an interactive document summarization system aimed at long, and multi-domain documents of the kind seen in academic publications. I show the iterative development and evaluation of ConceptEVA through two iterations. The first iteration incorporates a hierarchical summarization technique with an interactive visualization of concepts extracted from the document using a reference ontology. The second iteration, developed after evaluating the first iteration through an expert review, incorporates a multi-task longformer encoder decoder pre-trained for scientific documents that we fine-tune for paraphrasing and sentence embedding to handle long documents, and concepts visualized using a force-directed network that preserves semantic as well as co-occurrence relationships of document concepts. I also introduce a "focus-on" function that allows users to choose concepts of interest, examine their relationship with co-occurring concepts, and choose relevant concepts that will then be incorporated into a custom summary. An evaluation of ConceptEVA's second iteration through a within-subjects study using manually-generated summaries as baseline shows that ConceptEVA was helpful to participants for content-specific aspects of summarization, but participants with less experience struggled with critique-related aspects of summarization. Participants largely preferred the summary created through ConceptEVA's human-in-the-loop approach over their own manually-generated summaries. I also discuss the implications of our findings and suggest future development and evaluations of mixed-initiative summarization systems.

Chapter 6
Knowledge Exploitation for Technical Text Annotation

In this chapter, I present LabelVizier, a human-in-the-loop workflow that *exploits domain knowledge* and user-specific requirements to reveal actionable insights into annotation flaws, then produce better-quality labels for large-scale multi-label datasets. This application is inspired by the rapid accumulation of text data produced by data-driven techniques and the increasing importance of extracting "data annotations"—concise, high-quality data summaries from unstructured raw text. The recent advances in weak supervision and crowd-sourcing techniques provide promising solutions to efficiently create annotations (labels) for large-scale technical text data. However, such annotations may fail in practice because of the change in annotation requirements, application scenarios, and modeling goals, where label validation and relabeling by domain experts are required. We implement our workflow as an interactive notebook to facilitate flexible error profiling, in-depth annotation validation for three error types, and efficient annotation relabeling on different data scales. We evaluated our workflow in assisting the validation and relabelling of technical text annotation with two use cases and four expert reviews. The results show that LabelVizier is applicable in various application scenarios, and users with different knowledge backgrounds have diverse preferences for the tool usage.

6.1 Introduction

Building on the discussion in Chapter 2, data-driven approaches have pervaded manufacturing in the age of Industry 4.0, producing a large amount of digitized data in the form of unstructured technical text [296]. For example in machine maintenance,

machine operators and repairing technicians often create maintenance work orders (MWOs) to record their maintenance activities. However, the rich text of asset management history in MWOs usually sits untouched because of the potential inconsistency, incompleteness, or incorrectness [43] in the descriptive text. Compared to raw unstructured text, a set of high-quality annotations summarizing the content is preferred for "robust and reproducible" [43] analysis of large-scale technical text. In particular, these annotations can be utilized for the systematic problem identification and classification, root cause analysis, and product life cycle prediction [111], which provides precious insights and facilitate the key performance index (KPI) assessment and budget planning process. For instance, the statistics of the label "*too_hot*" in a heating, ventilation, and air conditioning (HVAC) system maintenance log dataset (see Sec. 6.6.1) could indicate how well the air conditioning system has been maintained and thus inform maintenance budget planning. This is also a critical research topic in technical language processing (TLP) [87].

However, it is not easy to create quality annotations and many important annotated datasets are riddled with labelling errors [259]. Given the exponentially increasing volume of unstructured text, researchers have gradually discarded conventional manual annotation approaches and turned to more efficient state-of-the-art machine learning (ML) techniques or commercial crowd-sourcing [165] platforms. Particularly, recent advances in weak supervision [287, 362] promise efficient large-scale text annotation. However, it is necessary to sufficiently validate and improve the annotations generated by such methods before delivering them to down-streaming tasks. Limited research efforts have been devoted to validation and relabeling of such large-scale technical text annotation. To facilitate this process, we developed LabelVizier, a human-in-the-loop workflow encapsulated as a visual analytic solution that supports reliable and efficient annotation validation and relabelling for domain practitioners to meet their specific application requirements.

LabelVizier helps identify and correct three types of annotations errors: (1) duplicate, (2) wrong, and (3) missing labels. Inspired by the practice of debugging in software

engineering, we profile the potential errors in the existing labels and devise visual analytics procedures to facilitate an efficient skimming of the labels and their context based on the domain expert's annotation preferences. We supplement this validation process by training a surrogate model to approximate the agnostic annotation process, visualizing the prediction metadata to expose potential errors, and providing LIME explanations [294] for root cause analysis. For the user-identified annotation errors, we support flexible relabelling of the dataset on the corpus, sub-group, and record levels. We implement this workflow as a web-based interactive notebook containing editable function blocks and an interactive visual analytic interface designed in close collaboration with the two domain experts on our team. We demonstrate how LabelVizier can benefit different application scenarios in two use cases and evaluate them with expert reviews from four domain practitioners. The results show that the domain experts appreciated the efficiency and accessibility of LabelVizier and are interested in using LabelVizier for their text-based analysis tasks. This work has the potential to impact a number of data-driven fields that emphasize annotation quality and, in particular, benefit multidisciplinary areas that deal with critical problems such as maintaining the vital infrastructure and ensuring community resilience.

The main contributions of this work can be summarized as follows:

1. Proposing a human-in-the-loop workflow that supports domain practitioners to efficiently conduct validation and relabeling tasks for large-scale technical text annotations from weak supervision.
2. Encapsulating this workflow as a web-based interactive notebook with a visual analytic interface that facilitates the identification of annotation errors and relabeling for different scales of data.
3. Distilling insights from domain experts in different domains and observe various preferences corresponding to their backgrounds, which could shed light on directions for improving LabelVizier and fulfil the needs of diverse domain practitioners.

6.2 Related Work

6.2.1 Technical Language Processing

Process monitoring, diagnostics, and prognostics have gained prevalence with the increased emphasis on smart manufacturing, and reduced machine downtime. This trend—coupled with lower cost, more accessible sensors and data storage solutions—has increased the volume of maintenance data [44]. Despite the potential benefits, companies frequently struggle to adopt advanced manufacturing technologies due to cost of and lack of technical expertise in data analysis [166]. Simple yet powerful solutions for data analysis are necessary to aid manufacturers improve their practices. There has been an increasing focus on sensor data and predictive maintenance using AI techniques [49, 342]. However, these works often neglect a large part of maintenance data: natural language contained short-text maintenance logs, which leads organizations to turn to NLP.

Technical text, however, poses challenges to commonly used NLP methods. Technical fields are often low-resource settings from an NLP perspective; they lack available resources such as annotated data and algorithms appropriate for specific analyses [88]. Transfer learning is the traditional strategy for addressing low-resource domains in machine learning [87]. Models that were generated from annotated data from resource-rich domains are adapted for the low-resource domain. Transfer learning approaches often assume limited differences between two different domains. But the technical text that appears in industrial information systems deviates considerably from "standard" English [87], full of expressions like "1 W Mech Insp Ball Mill BM001" and "DSHT Cons Thkner rplace bed press".

The lexical, grammatical, and terminological differences between "standard" English and industrial technical text have spawned bespoke domain-specific NLP adaptations that are largely outside of mainstream NLP [87]. TLP is a human-in-the-loop, iterative approach that addresses perceived shortcomings of applying standard NLP (natural language processing) to technical text data [87]. Originating with manufacturing maintenance, it is an adaptation of NLP that focuses on the technical text

communicated within specialized domains. TLP emphasizes the practical importance of semantic information and extends its system boundaries beyond algorithms and pipelines to include human input and community resources [43]. The short-text from maintenance work orders (MWOs) are important analysis corpora for TLP [224, 313]. They record in detail the maintenance history of equipment and collectively capture vital information about inspections, diagnoses, and corrective actions [43]. Annotation methods for MWOs have been the subject of recent research in TLP. Tools, such as Nestor[1] have been developed to support the manual injection of critical real-world knowledge by allowing for the annotation of the MWO text descriptions via tagging to facilitate automated categorization and analyses. Machine learning systems can then use these tags as a signal to help ensure correct outcomes [87].

6.2.2 Large-Scale Text Annotation

The exponential growth of text data has made the current manual text annotation approaches, e.g., crowd-sourcing [165], deficient in meeting the pressing demands for high-quality large-scale annotations [212, 394]. As an alternative, researchers have developed the weak supervision techniques [115,212,287] that leverage human-defined labeling functions (LFs) [287], small labeled datasets [362], or existing text paradigms with multi-type metadata [237] for more efficient text annotating. However, most of these approaches trade off labeling speed or cost with annotation quality [212, 237, 287, 362], and the generated labels are mainly evaluated by numerical performance matrices, such as accuracies [287], F1 scores [362]. Without human review, it is uncertain whether such annotations are of sufficient quality for real-world applications. In light of the deficiencies of manual and automatic text annotation approaches, a series of semi-automatic text annotation frameworks have been proposed, allowing humans to annotate large-scale text data with the help of automatic modules, which can be coordinated labeling modules [400] or deep learning techniques such as attention model [64], human-validated labeling functions [100,295], and transductive semi-supervised learning [84]. However, the annotation quality of such frameworks still lacks human validation—they either

[1]https://nist.gov/services-resources/software/nestor

only verify quantitative performance matrices [400], or sample a small subset for humans to inspect results [84, 100]. Although there are a few works for improving the annotation quality [22, 215, 346], they are mainly designed for image or video data and hence not directly applicable to technical language datasets. Given the importance of high-quality annotations [94, 232], a human-centered tool is needed to support the validation of large-scale text annotations.

6.2.3 Technical Text Visualization

In the past decade, the idea of applying visualization and visual analytics to technical text analysis has been broadly embraced. Manufacturing enterprises are becoming aware of the value of maintenance records they collect and are supporting visualization research [8, 54, 59, 138]. Academic researchers have developed visual analytical strategies for maintenance records [348] and error logs [180, 205, 255, 318]. In particular, La VALSE [137] and MELA [318] are scalable visualization tools with multiple visualization interfaces incorporating different logs for interactive event analysis. ViBR [51] provides a visual summary of large bipartite relationships by via minimum description lengths and is used for vehicle fault diagnostics. However, existing solutions have prerequisites on either the text format or the quality of the labels. Some assume that there exists a well-defined set of labels [143] to train a classification modelfor the annotation task or assume a trivial effort to define these labels in the pre-processing stage [86, 404] when they are not available as input. Others expect that the text can be generated from grammar or rules so that the labels can be derived from clustering [137].

In this work, we address inconsistent technical text created by human maintainers that contains domain jargon and labels of unknown reliability. Unlike other approaches, we do not have prerequisite text formats nor do we make assumptions about the labels or their quality. We also do not rely on the text's grammatical structure

6.3 Validation and Relabeling for Text Annotations

In this section, we define the problem we wish to solve and clarify our assumptions. We also derive the annotation error types and design requirements based on the industrial

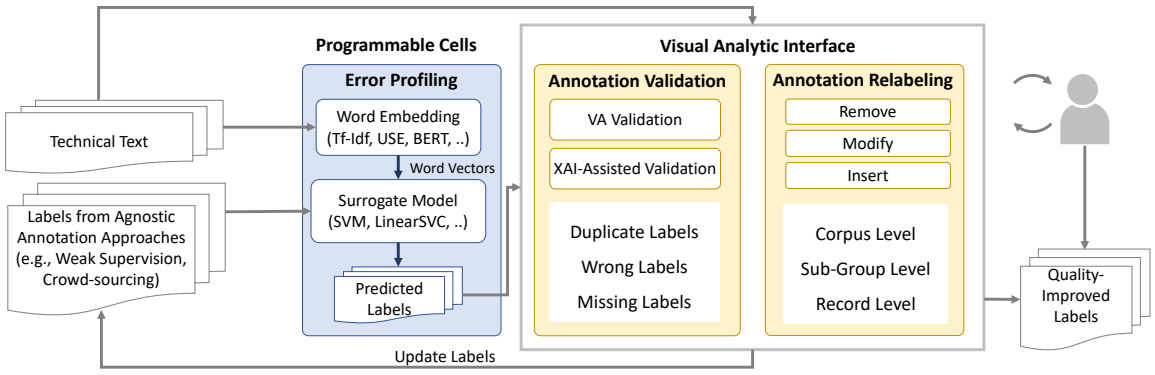


Figure 6.1: The LabelVizier workflow consists of three phases: Error Profiling, Annotation Validation, and Annotation Relabeling. It is implemented as a web-based interactive notebook which takes technical text and the corresponding labels as input. With the actionable insights provided by the surrogate model, the XAI method, and the visualization, users can identify three error types and improve annotation quality at three different data scales.

experience of two coauthors and an exploration of a machine maintenance log dataset.

6.3.1 Problem Definition

Like software development, TLP often requires machine-assistance in the validation of *large*, *complex*, and *context-specific* sets of text. To fulfill this need, we aim to facilitate the validation of annotations and the correction of labelling errors by designing visual analytic techniques for domain practitioners.

Our target users are domain practitioners and data analysts in need of assessing and improving the annotations for large-scale technical text data. We expect they have the analytical skills to interpret the model performance metrics and interact with LabelVizier. We also make two assumptions about the technical log text and labels:

1. There exists a finite set of labels $L = \{l_1, l_2, ..., l_n\}$. And the mapping from each record s_i to the labels is defined by $l_i = f(s_i)$, where $l_i \subseteq L$. In the context of this chapter, $f(s_i)$ is agnostic and the quality of L requires expert verification.
2. There exists a finite set of label categories $C = \{c_1, c_2, ..., c_n\}$. Each label l in L belongs to one label category in C. Note that the "label category" in our context is a higher-level taxonomy of labels. For instance, label "*air-conditioner*" belongs to "Item" and "*too_hot*" belongs to "Problem" in Sec. 6.6.1.

Given there is no formalized taxonomy of labeling errors in the TLP domain, we target three dominant types of annotation errors distilled from two coauthors' long-

term industrial experience:

- **E1 Duplicate Labels** share duplicated words (e.g., "*temperature*" and "*room_temperature*") and/or express semantic meanings (e.g., "*too_cold_building*" and "*temperature_too_cold*").
- **E2 Wrong Labels** involve labels with conflicting meanings (e.g., one record is labeled with "*too_cold*" and "*too_hot*" simultaneously). It can also refer to an unreasonable label name (e.g., "*building_building*") where the label refers to a non-existent class.
- **E3 Missing Labels** refer to labels that should be assigned to technical records but are absent (see examples in Sec. 6.6.1). Missing labels are relatively hard to detect if there are already other labels assigned to the record.

The output of our workflow is a set of labels with improved quality.

6.3.2 Design requirements

After much discussion on a weekly base, our team, which included two TLP domain experts, agreed to four design requirements for LabelVizier to address the problem defined in Sec. 6.3.1:

- **R1 Label Overview**: As the first step of label debugging, LabelVizier needs to provide users with a summary overview of all technical text and labels. The visual interface needs to present label distribution in the finite label set L and illustrate their categories c (if available) intuitively. The visual interface also needs present the currently assigned labels of one or multiple record(s) s_i in different levels of detail and from different perspectives per user demand to support intensive context comprehension.
- **R2 Label Quality Screening**: LabelVizier need to support an efficient evaluation of the quality of existing annotations. In particular, the visual interface needs to allow users to quickly locate labels that potentially fall into the three types of errors (see Sec. 6.3.1). After that, it should help users confirm the error by providing sufficient context information about the related labels and explaining how they were assigned to specific records.
- **R3 Interactive Relabeling Support**: Once the errors are identified and confirmed, LabelVizier needs to interactively collect the user's relabeling suggestions and

apply them to specific scales of the dataset per user request. In particular, users should be able to make suggestions to remove or modify an existing label or insert new labels according to their best judgment. After that, such modifications should be applied to entire corpus, a sub-group, or an individual record per user demand.

R4 **Accessibility and Flexibility**: LabelVizier should be accessible to domain practitioners of varying backgrounds. On the one hand, the basic functionalities of the visual interface should be intuitive enough for users without a computing background during the validation and relabeling tasks. On the other hand, LabelVizier should provide users with in-depth information on demand and the flexibility to adjust the data processing or model training settings so that the analysis process also satisfies experts with more computing experience and special analysis needs.

6.3.3 Datasets

We involve two TLP datasets, **HVAC** and **NLU**, in this work:

HVAC is an internal dataset from our industrial collaborators with over 21,000 pieces of maintenance records from an HVAC system. Each record contains two text fields: "LONG_DESCRIPTION" and "DESCRIPTION". "LONG_DESCRIPTION" describes the detailed maintenance information, including the problem, the solution, the maintainer, the corresponding machine, etc., while "DESCRIPTION" is a concise version, which is often a sentence or a set of keywords. There are also eight categories of labels available for each record, including "P" (Problem), "S" (Solution), "I" (Item), "PI" (Problem Item), "SI" (Solution Item), "X" (Irrelevant), "U" (Unknown), and "NA". For example, the category "P" includes labels such as "*too_hot*", "*leak*", and the category "SI" includes labels such as "*adjust thermostat*", "*replace valve*", etc. These labels were produced by a weak supervision method, and their quality remains agnostic.

NLU [37] contains over 25,000 human-robot interaction records and the corresponding labels, collected from a voice AI agent serving in an intelligent home system. Each record includes three text fields: "question" is a pre-designed human-robot interaction

question; "answer" and "answer_normalized" contain the original and normalized user answers, respectively. There are three categories of labels, including "scenario", "intent", and "suggested_entities". For example, the category "scenario" includes labels such as "*weather*", "*music*", and the category "intent" includes labels such as "*request*", "*send email*", etc. These labels were generated from a crowd-sourcing platform, and their quality requires validation as well.

6.4 Methodology

6.4.1 Workflow

We designed the LabelVizier workflow as an iterative framework with three major phases: (1) Error Profiling, (2) Annotation Validation, and (3) Annotation Relabeling. A regular analysis process starts from the **Error Profiling** phase, in which we train a surrogate model with the technical text and their existing labels to approximate the prior annotation process. Then, users can conduct the first round of **Annotation Validation** through the integrated visual analytic interface, where multiple coordinated views are provided to assist an efficient investigation of labels (6.3.2) and detection of three types of errors. After that, users can move on to the **Annotation Relabeling** phase and relabel the identified results at three different levels: corpus level, sub-group level, and record level (6.3.2). A more detailed description of our visual and interactive support on these three levels is provided in Sec. 6.5. After the first pass of the three phases, users can iterate between **Annotation Validation** and **Annotation Relabeling** for multiple rounds till the annotation quality converges with their standard of satisfaction. It is also worth mentioning that LabelVizier simplifies the input and output of phase(1) so that users only need to make minor hyperparameter adjustments to execute different use cases with various analysis purposes (Sec. 6.3.2).

6.4.2 Surrogate Model for Error Profiling

In the first phase of the LabelVizier workflow, we train a surrogate ML model [78] to approximate the generation process of existing labels in the dataset. To ensure that the surrogate model can achieve satisfactory performance and reflect potential

annotation issues, we tuned the model architecture with the interactive notebook to fit the specific dataset. Then, by visualizing the model's intermediate results (e.g., prediction probability [170]) in the second and third workflow phases, we help users uncover potential annotation flaws. In this way, users start their label validation from those suspicious labels related to unusual model behaviors (Sec. 6.3.2) and locate a group of potential labeling errors for inspection. After addressing these labels, users can retrain the surrogate model with the better-quality dataset to obtain a reusable model incorporating domain knowledge from human experts and save it for future annotation tasks.

The error profiling phase of LabelVizier require the surrogate model to be: (1) lightweight, so that the model tuning is time-effective; (2) accurate in producing similar results to existing annotations. For (1), we utilize lightweight and time-effictive word-embedding and ML methods to process text data and train the annotation classifier. For instance, to process the input technical text data, we adopt computationally efficient and widely-used word embedding techniques, including TF-IDF (term frequency-inverse document frequency) [377] and truncatedSVD (Singular value decomposition) [129], to encode the original text into real-valued vectors. For (2), we iterate multiple processes with different model training settings, and audit quantitative performance matrices in the validation split until reaching the best result. Thus, we ensure the surrogate model achieves satisfactory performance, i.e., the alignment between the predicted labels and the existing annotations is reasonable for the surrogate model to simulate the annotating process and provide hints for users. Specifically, the average hamming loss is 0.02, the micro f1 score is 0.8044, and the average macro f1 is 0.6703, where smaller hamming loss, and larger micro & macro f1 scores indicate better performance. Besides, the predicted probabilities $predProba$ of the fitted LinearSVC $fittedModel$ is obtained to act as the clue of finding suspicious labels.

The data processing pipeline of the Error Profiling phase is illustrated with Algorithm 2. The inputs of the Error Profiling phase of LabelVizier are the technical text data and their finite set of labels (refer to 1) annotated by automatic tagging tools

Algorithm 2 Surrogate model for simulating label generation

Input:

Technical text data, $techText$;

Programmatically-created labels in text format, $labelText$;

Label categories as a list, $categList$;

Output:

Word vector encoded from text, $wordVector$;

Surrogate models for each category, $modelDict$;

Predicted probabilities of the models, $predProba$;

$vectorTemp \leftarrow$ TF-IDF($techText, stopwords$)
$wordVector \leftarrow$ truncatedSVD($vectorTemp$)
$labelVectorTotal \leftarrow$ OneHotEncoding($labelText$)
$labelVectorDict \leftarrow$ Categorize($labelVectorTotal$)
$modelDict \leftarrow \{\ \}$
$predProba \leftarrow [\ ]$
for $category$ **in** $categList$ **do**
    $labelVector \leftarrow labelVectorDict[category]$
    $fittedModel \leftarrow$ LinearSVC($wordVector, labelVector$)
    $modelP \leftarrow$ Pipeline(TF-IDF, truncatedSVD, $fittedModel$)
    $modelDict[category] \leftarrow modelP$
    $curProba \leftarrow fittedModel.pred\_proba(wordVector)$
    $predProba \leftarrow predProba.concat(curProba)$
end for
return $wordVector, modelDict, predProba$

or weak supervision techniques (refer to Fig. 6.1). The technical text data is fed into a word embedding process, whose main goal is to encode the text into real-valued vectors [191]. Consistent with 2, the labels are grouped into different pre-defined categories according to its semantic meanings, e.g., labels "*too_hot*", "*too_cold*", "*noisy*" in a machine-maintenance dataset are categorized as "P" (Problem). Then, a machine

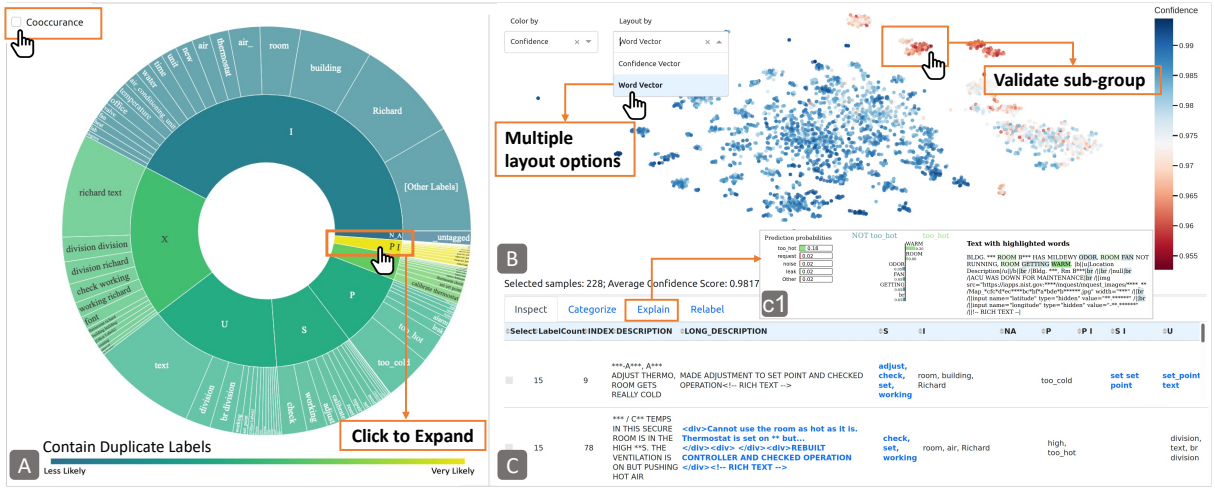


Figure 6.2: The LabelVizier interface with the HVAC dataset. (A) **Label Investigation View** visualizes the label and category hierarchy relationships; each category can be expanded to present label co-occurrences (see Fig. 6.3 (A)). (B) **Record Projection View** presents the record distribution oto support sub-group validation, layout by model confidence vectors or input word vectors. The color represents "model prediction confidence" or "record info density". (C) **Inspection & Operation View** includes multiple tabs, "Inspect" for record and label inspection, "Categorize" for category-based investigation, "Explain" for model behavior interpretation (c1), and "Relabel" for relabeling operations.

learning classifier is trained by taking the word vectors as the input and the labels from each category as the ground-truth, which means we will get a multi-label annotation classifier for each category. Finally, LabelVizier will automatically pack the fitted word embedding tool together with each of the classifier respectively as callable functions, allowing the integrated visual interface to easily access the functions and visualize the models' behaviors.

6.4.3 Model Behavior Explanation

LabelVizier utilizes one of the state-of-the-art eXplainable Artificial Intelligence (XAI) techniques–LIME (Local Interpretable Model-Agnostic Explanations) [294]–to support the annotation validation. For each of the constructed surrogate models, LIME performs perturbation-based analysis over a given text record and presents the explanation by highlighting the rationale behind the model's prediction. It exposes the weakness of the model and the pitfalls of the input technical text and thus could help users more accurately inspect a potential annotation error and make a reasonable relabeling decision. The LIME explanation is integrated into the "Explain" tab of Record Projection

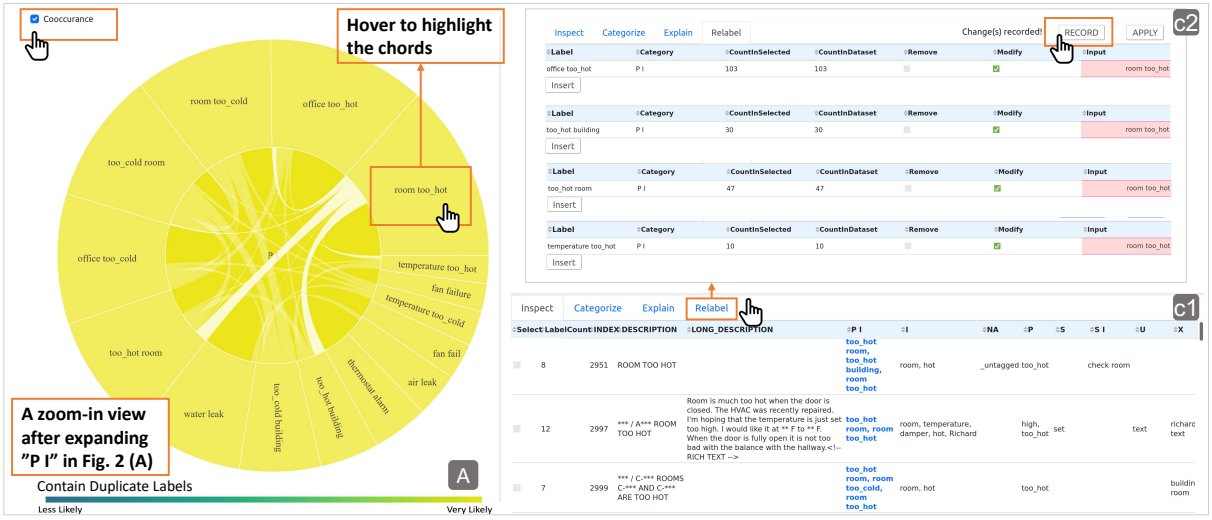


Figure 6.3: Duplicate label validation with the HVAC dataset (Sec. 6.6.1). The chord diagram (A) shows that labels "*office too_hot*", "*room too_hot*", "*too_hot building*", and "*too_hot room*" co-occur frequently. Duplication is confirmed via record details in (c1) and fixed in "Relabel" tab (c2).

View LabelVizier and is triggered when users select a record from the "Categorize" tab for further inspection (more details in Sec. 6.5 and examples Sec. 6.6).

6.4.4 Implementation

To maximize its accessibility for sharing and flexibility for customization (**R4**), we implement the LabelVizier workflow as a computational notebook. We use multiple Python data analysis libraries including Pandas [269], Numpy [146], and Joblib [167] for data processing and intermediate metrics analysis. In addition, the word embedding techniques (TF-IDF and truncated SVD) and ML methods (LinearSVC) discussed in Sec. 6.4.2 are implemented with scikit-learn [279], and the LIME technique is with LimeTextExplainer [294]. To ensure smooth integration and faster rendering speed, we embed the visual interface (Fig. 6.2) in the computational notebook with Plotly's JavaScript Graphing Library and Plotly Dash. And we use the t-distributed stochastic neighbor embedding (t-SNE) algorithm for dimensionality reduction when visualizing the high-dimensional word vectors and confidence vectors for the Record Projection View. We also deliberately separate the functions in the notebook so that users can easily plug in any word embedding and dimensionality reduction algorithms for their specific analysis needs with minor programming.

6.5 Visual Analytic Interface

To fulfill the design requirements in Sec. 6.3.2 and concretize the LabelVizier workflow in Sec. 6.4.1, we design a visual analytic interface (see Fig. 2.2) that contains three major components: (A) Label Investigation View, (B) Record Projection View, and (C) Inspection & Operation View. In this section, we demonstrate how we can coordinate these views to locate the three types of errors and perform multi-level validation and relabeling on annotations.

6.5.1 Annotation Validation

With the coordination among different components of LabelVizier, users can efficiently validate the annotation quality and identify the three major types of error introduced in Sec. 6.3.1 (**R2**).

6.5.1.1 Duplicate Label Detection

To support duplicate label detection, we designed the Label Investigation View (Fig. 6.2 (A)) and the "Inspect" tab of Inspection & Operation View. We choose the sunburst diagram for Label Investigation View to provide an overview of the hierarchical relationship between labels and their category (**R1**), as well as the distribution of the labels across categories – the size of the label sectors at the outermost layer represents the number of records in the dataset assigned with the corresponding label. We also encode the possibility of duplicate labels into the sectors colors to provide a priority recommendation for the user inspection. This duplication possibility is the average of a ratio of co-occurrence number $Co(l_i,\ l_j)$ to the total appearance $Num(l_i)$ of each label l_i in the category:

$$P_{duplication} = \frac{1}{n_{categ}} \sum_{i=1}^{n_{categ}} (\frac{1}{n_{occur}} \sum_{j=1}^{n_{occur}} \frac{Co(l_i,\ l_j)}{Num(l_i)}) \tag{6.1}$$

The sunburst diagram is expandable per user request. And in the zoom-in view, we embed a chord diagram to illustrate the label co-occurrence in the same record, which is a strong indicator of duplicate labels (e.g., Fig. 6.3 (A)). In this chord diagram, the co-occurring labels are connected with white chords, with their thickness representing the

co-occurrence frequency. For example, the chord between the label "*room too_hot*" and "*too_hot room*" is thicker than that between "*room too_hot*" and "*water leak*", indicating a heavier co-occurrence pattern and potential duplication of the former pair. A larger number of thicker chords also indicates a higher possibility of existing duplicated labels in this category, which corresponds to the brighter sector color described above.

Users can further inspect the context of any suspect labels by clicking on them and checking the updated data table under the "Inspect" tab in the Inspection & Operation View (Fig. 6.2 (C)). This data table presents all the records across the dataset assigned with the selected label. In this way, users can efficiently locate and evaluate the correctness of potential problematic labels.

6.5.1.2 Wrong Label Detection

Wrong labels can be detected with the Record Projection View (Fig. 2.2 (B)) in coordination with the "Categorize" and "Explain" tabs of Inspection & Operation View. To provide a two-dimensional (2D) overview for all records (**R1**), we apply the t-SNE [358] algorithm to project the customized record vectors onto the 2D space and visualize each of them as a dot. The customized record vector can be a "word vector" or a "confidence vector". When the "word vector" is used for layout, the distance among the dots indicates the semantic closeness of descriptions in their corresponding records. When the "confidence vector" is used for layout, the distance among dots indicates the model behavior towards similarity when predicting labels for the corresponding records. We provide two options to color the record projections — "information density" and "confidence score". The "information density" is more useful in locating missing labels, so we will discuss it in Sec. 6.5.1.3. The "confidence score" is the mean value of all dimensions of the aforementioned "confidence vector", which could expose the records containing more labels predicted with low confidence, and thus provide hints to locate sub-groups that potentially contain labeling mistakes (Fig. 2.2 (B)).

Once a cluster with low confidence is identified and selected, a heatmap under the "Categorize" tab in Inspection & Operation View (Fig. 6.4 (c1)) will be triggered, where each row refers to a single record; each column represents one label category, and the

color indicates the model's average confidence score.

To support deeper understanding of the model reasoning process, we provide LIME explanations under the "Explain" Tab (Fig. 6.2 (c1)). The explanation includes three parts: the left bar chart visualizes the top five predicted labels and their prediction probabilities; the middle bar chart visualizes the "score of contribution" of the input words to the top label; the right side shows more context information and the original text record, where the positive and negative contributors are highlighted with different colors. Combining these three kinds of information, we aim to help users verify whether the rationale behind the model's decision aligns with their knowledge (**R2**).

6.5.1.3 Missing Label Detection

The detection of missing labels also involves the Record Projection View (Fig. 6.2 (B)) and "Inspect" tabs of Inspection & Operation View. We designed the "information density" metric to highlight records more likely to have missing labels. This metric is determined by the ratio of label count and the input text length:

$$D_{Info} = log(\frac{Count(labels)}{WordCount(text)}) \tag{6.2}$$

Once the users locate and select a cluster of records with low "information density", the "Inspect" tab of Inspection & Operation View will be updated for verification of the label missing issue. It is also worth mentioning that higher "information density" can insufficiently indicate the existence of duplicate labels, but further verification is required with the process in Sec. 6.5.1.1.

6.5.2 Annotation Relabelling

After confirming a labeling error, users can improve the annotation quality of the dataset (**R3**) on three data scales: corpus level, sub-group level, and record level:

1. **Corpus level relabeling** updates the label across the entire technical text dataset. It is achieved by clicking on a label from Label Investigation View and use the "Relabel" tab of Inspection & Operation View to "remove", "modify", or "insert" it. For example, the label "*building_building*" is considered to be a "wrong label"

error at the corpus level. Users can select it from Label Investigation View and remove it from all affected records.

2. **Sub-group Level relabeling** involves a sub-group of records within the dataset. It is achieved by selecting a sub-group of records with the lasso tool in Record Projection View and relabeling them with the "Relabel" tab. The number of records in one sub-group can vary from a dozen to hundreds.
3. **Record level relabeling** updates individual record(s) associated with a specific label error. For example, a user looks over the records through the "Inspect" tab and notice two records missing the label "alarm". They can select these two record(s) with the checkbox and relabel them under the updated "Relabel" tab. These relabeling operations are only applied to the selected records.

To eliminate waiting for dataset updates and projection re-rendering, the visual interface is only re-rendered when the user request to apply the changes. To achieve this, we record the relabeling operations as a history list and sequentially apply them to the database per request.

6.6 Use Case Scenarios

This section describes two use case scenarios where LabelVizier assists domain experts in validating the quality of technical text annotations and conducting efficient relabeling for the incorrect annotations.

6.6.1 Case 1: Maintenance Management for HVAC System

This use case involves Amy, a maintenance manager who monitors machine maintenance records to track maintenance-related issues and to plan for future maintenance resources (e.g. budgets, maintainers, etc.). With the HVAC dataset, we demonstrate how she uses LabelVizier to validate the label quality of MWOs and make maintenance management based on the more accurate labels.

After finishing data processing and surrogate model training in the programmable cells of LabelVizier, Amy starts validating labels through the interface. Because the frequency of similar labels reflects the prevalence of a maintenance issue and influ-

ences decision priorities and budgets, Amy chooses to screen duplicate labels at first. She notices from the Label Investigation View (Fig. 2.2 (A)) that the category "PI" is the most likely to contain duplicate labels, so she expands it to check for label co-occurrance (Fig. 6.3 (A)). As indicated by the chord thickness, "*office too_hot*", "*room too_hot*", "*temperature too_hot*", "*too_hot building*", and "*too_hot room*" co-occur very frequently. Because these labels have similar semantic meanings, Amy further inspects their context in the "Inspect" table (Fig. 6.3 (c1)) and confirms that they are duplicates. Such duplication will overemphasize air-conditioner-related "*too hot*" issues and may cause excessive allocation of maintenance resources. Amy removes the redundant labels and unifies the rest with "*room too_hot*" with the "Relabel" (Fig. 6.3 (c2)). Now that Amy has created more accurate and consistent labels for temperature-related problems, she counts their frequency, evaluates the problem severity, and decides to arrange for regular examination of all air-conditioners across the company.

Amy moves on to the Record Projection View (Fig. 2.2 (B)) to screen for wrong labels which are another type of error that can mislead maintenance planning. Amy notices that there are several clusters with lower confidence scores (Fig. 6.4 (B)), indicating that the surrogate model performed worse and might predict the wrong labels for the corresponding records. After selecting one, she learns from the updated "Inspect" tab that the labels "*Richard*" in category "I" and "*br richard*" in category "X" appear in many records. Amy browses the context to check if this is related to a maintainer's name or is an improper description but finds the phrase "br richard" doesn't appear to be semantically relevant to either "DESCRIPTION" or "LONG_DESCRIPTION". Seeing this issue also appears in many other clusters, Amy decides to further investigate its cause with the "Categorize" and "Explain" functions. Under the "Categorize' tab, she clicks on the cell indicating lowest prediction confidence in category "I" and "X" (Fig. 6.4 (c1)) to trigger a LIME explanation. As shown in the left bar chart in Fig. 6.4 (c2), the model's top prediction for the selected record is the label "*br richard*". However, the middle and right part shows that most positive contributors to this prediction are HTML metadata such as "TEXT" and "RICH". Amy realizes that the wrongly-predicted

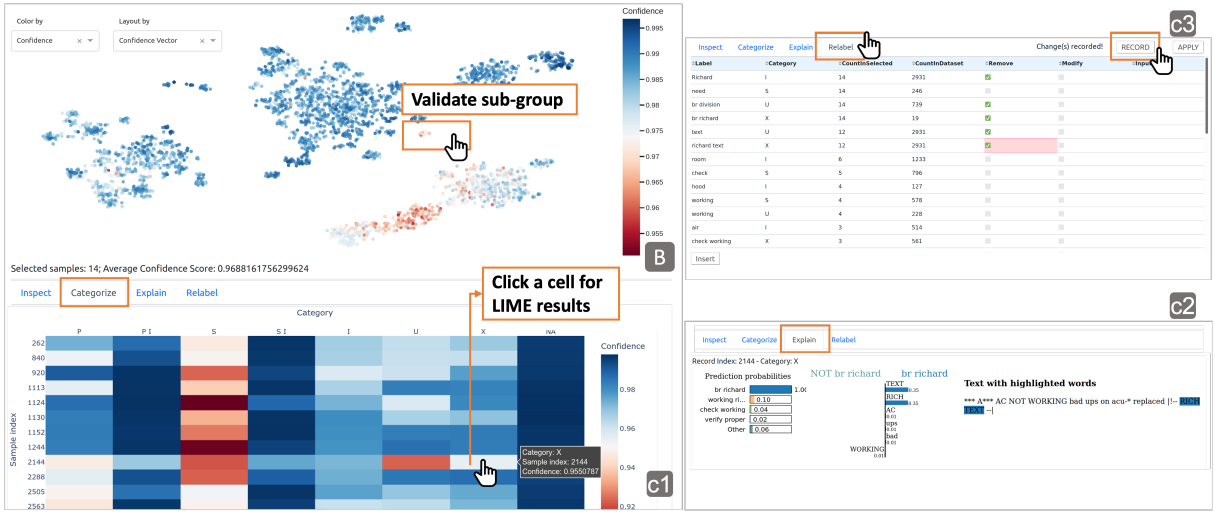


Figure 6.4: Wrong label detection in HVAC dataset using LabelVizier (Sec. 6.6.1). Users can select a sub-group with lower confidence in (B) and inspect model confidence of each category in (c1). Then they can click on cells in (c1) to activate LIME explanation in (c2) for model behavior interpretation. After confirming the error, users can remove the wrong labels with (c3).

label "*br richard*" might originate from the presensce of HTML tags. After seeing similar LIME explanations for more records in this cluster, her hypothesis is confirmed – the model referred to HTML tags to predict the wrong labels. Amy removes these wrong labels (Fig. 6.4 (c3)) and decides to conduct a thorough cleaning of the dataset later to remove HTML tags.

Amy also checks the info density and does not find any severe label missing issues. Then she reloads the updated labels into LabelVizier and confirms that the new annotations are satisfactory. Finally, Amy executes the code cells of the computational notebook (Sec. 6.4.4) to re-train the surrogate model with the relabeled dataset. In this way, she preserves her domain knowledge in the model that can be used to annotate any future maintenance records.

6.6.2 Case 2: Data Cleansing for NLU Model Training

In this section, we demonstrate how LabelVizier can help modify the annotations of the training data for a specific application scenario. This use case involves Steven, a data engineer working in a company that provides conversational agent (CA) services. Steven coordinates the large-scale crowd-sourcing process to provide high-quality training data for the natural language understanding (NLU) model [37] embedded in the CAs,

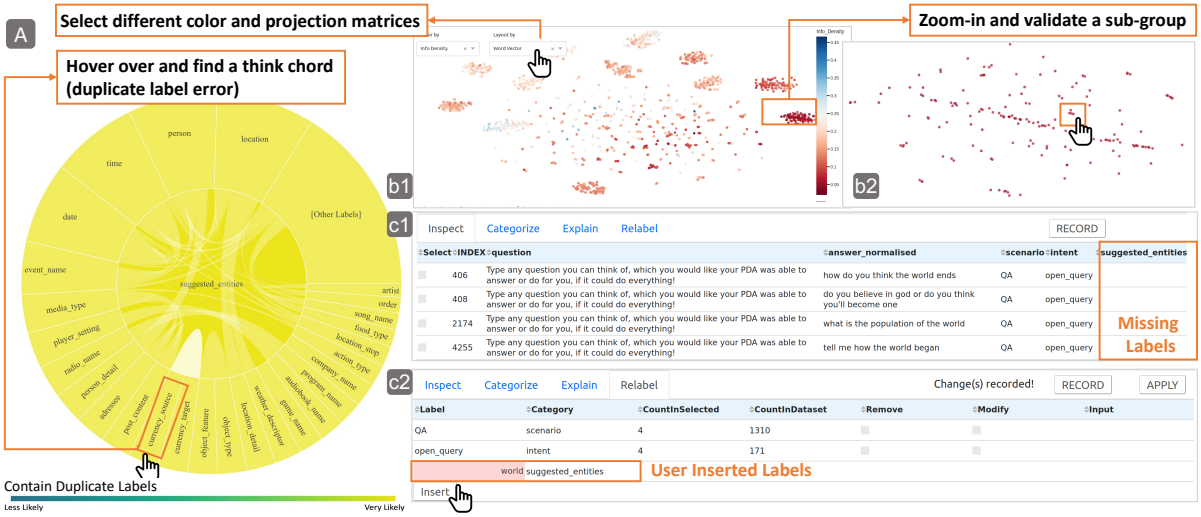


Figure 6.5: Finding duplicate and missing labels in NLU dataset using LabelVizier (Sec. 6.6.2). The chord diagram in the Label Investigation View (A) reveals the duplicate labels "*currency_source*" and "*currency_target*". The clusters with low $info\_density$ in the Record Projection View (b1) highlight records with missing labels. The "Inspect" tab (c1) shows that the "suggested_entities" for selected records in (b2) are missing. User-suggested labels can be inserted through the "Relabel" tab (c2).

which summarizes the semantic content of user utterances by mapping it to structured, abstract representations (labels) that support the decision making process.

Steven uses LabelVizier to validate and debug the crowd-sourced annotation results before delivering the dataset for the downstream machine learning tasks. The NLU model requires all labels to be independent and accurate so that the voice AI agent can exclusively query them in the search engine and provide correct answers to the users. To remove those dependent labels with semantic duplication, so Steven starts by looking for them via the Label Investigation View. According to the coloring of the categories, the category "suggested_entities" is most likely to include duplicate, so Steven expands this category to inspect the label co-occurrences. The thick chord indicates that the labels "*currency_source*" and "*currency_target*" heavily overlap (Fig. 6.5 (A)). Steven inspects the detailed context of the corresponding records with the Inspection & Operation View and finds that most of these records are related to currency exchange questions. Although the two labels "*currency_source*" and "*currency_target*" appear reasonable, Steven still decides to merge them into the single label "*currency_source_and_target*" to facilitate the down-streaming task.

When using the "Inspect" tab to investigate the duplicate label issues above, Steven

notices that missing labels are a common with this dataset. He moves on to the Record Projection View to facilitate find more missing labels. He chooses the options of "Color by Info Density" and "Layout by Word Vector" to highlight clusters with similar semantic meanings and lower info density (Fig. 6.5 (b1)). Then he uses the lasso tool to select the most notable cluster and observes from the Inspection & Operation View (Fig. 6.5 (c1)) that all the records in this cluster were labeled as "*QA*" under the category "scenarios" and "*object_query*" under the category "intent", but not assigned with any labels under the category "suggested_entities". With a second look at the input "questions", "answers", along with the model's reasoning process from the "Explain" tab, Steven figures out the cause — the model only captured the information from the records' shared "question" but ignored the "answers". To fill the missing labels, Steven zooms into this cluster in the Record Projection View (Fig. 6.5 (b2)) and uses the lasso tool to select each sub-clusters with similar semantic meaning. After selecting one sub-cluster (Fig. 6.5 (b2)), Steven notices similar semantic meanings of the selected records – for "answer_normalized", those records have sentences such as "*how do you think the world ends*", "*tell me how the world begin*" and "*do you believe in god*", etc. Considering these questions were asked by their CA users, Steven believes "*apocalypticism*" or "*philosophy*" would be proper labels for the category "suggested_entities". He inserts them into the selected sub-group of records with the help of the "Relabel" tab. Steven conducts the same operation to the few other low-info-density clusters, and fix the label missing issues accros the entire dataset.

Finally, Steven applies his relabeling operations to the dataset and updates the interface. After confirming the quality of the annotations, he delivers the dataset to the machine learning engineers for the downstream training tasks. With LabelVizier, Steven optimizes the annotation quality from crowd-sourcing results and avoids the potential flaws that can bias the training of the voice AI agent.

6.7 Expert Reviews

LabelVizier was developed with the participation of TLP domain experts (Sec. 6.3.2) over the course of two years. To evaluate the generalisability of our workflow and reveal insights from or practical value to domain practitioners, we invited another two TLP domain experts (E1 and E2) and two experts from other domains (E3 and E4) into our expert review studies.

6.7.1 Expert Demographics

All experts were experienced in data analysis and performed data annotation tasks in their daily work. E1 and E2 was research engineers from the TLP community, who were familiar with and had worked on the analysis of the HVAC dataset (Sec. 6.6.1) before the study for several years. E3 was an economist and statistician who analyzed large-scale tabular datasets to gain insights into community resilience. E4 was a research social scientist who manually annotated large-scale datasets about risk perception and evacuation decision-making, and was in need of speeding up this process.

6.7.2 Tasks and Setup

We conducted two pilot studies to simulate the remote setup, test the LabelVizier execution environment (online Colab via a local Jupyter Notebook) and adjust the content of the tutorial sessions. Then we finalized a semi-structured, open-ended expert review in which each expert was asked to explore one of the two datasets described in Sec. 2.6. Based on their domain of expertise and familiarity with the dataset, E1 and E2 used the HVAC Dataset while E3 and E4 used the NLU Dataset. We shared the original dataset and its documentation[2] with E3 and E4 before the study so that they could get familiar with it in advance. Because the existing annotation in the NLU Dataset is relatively clean, we used an adapted version with two manually inserted errors for each error type in Sec. 6.3.1 in the study.

The study was conducted online via a video conferencing where the domain experts accessed LabelVizier from Google Colab Notebook[3] via their personal computers. We

[2]https://github.com/xliuhw/NLU-Evaluation-Data
[3]https://colab.research.google.com/

shared the tutorial document with the experts no less than two days before the study.The online study session started with a 25-min tutorial session that combined an introductory presentation, a live demonstration, and the mini-tasks. An example mini-task for the HVAC dataset was "Please use LabelVizier to find one pair of duplicated labels under the category 'PI' (Problem Item), and then suggest how to modify it with the 'Relabel' tab". After the tutorial session, the experts were asked to freely explore their assigned dataset to validate and relabel the annotations (20-25 minutes). During this process, the experts followed the think-aloud protocol to verbalize their thinking and suggestions. Finally, the experts responded to a questionnaire with their demographic information and general feedback of LabelVizier.

6.7.3 Observations

In our study, all experts appreciated the value of LabelVizier for facilitating annotation refinement and expressed willingness to use it in their daily work, describing it as *a very good tool* (E2) that was "*helpful at a high level of quickly and...pleasantly...identifying issues than just scrolling through a spreadsheet*" (E4). Meanwhile, we observed that domain experts with differing backgrounds interacted distinctly with LabelVizier and sometimes provided divergent comments towards the same features. We categorize the their behaviors and feedback during the exploratory and describe them below.

Learning Curve. Based on their familiarity with the dataset and the tool, domain experts required differing times to overcome the learning curve (**R4**). For instance, E1 and E2 had previously worked on the HVAC dataset with other annotation tools, so they spent less time grasping LabelVizier compared to the other two experts (E3, E4). E2 expressed great interest in the methodology "*under the hood*" and asked many technical questions to understand the underlying mechanism during the tutorial session. Although E3 and E4 needed more hands-on instructions about using our tool, they were capable of replicating the moderator's operations and accomplishing the exploration task after the tutorial sessions. They praised our tutorial session design, saying "*it helped very much...after (the moderator) demonstrated, it very easy to replicate*" (E2).

Functionality. In all four studies, the experts were able to efficiently evaluate the

quality of the annotations (6.3.2) and successfully accomplish the relabelling (6.3.2) by coordinating information from the three major views of LabelVizier (Fig.2.2). Moreover, E1, E2, and E4 successfully mastered the relatively complex "Categorize" and "Explain" functions and utilized them to understand the root cause of a potential wrong label. We also received requests for more delicate annotation manipulations and more complicated information support from experts with shorter learning curves, such as modifying the name of a label category (E2) or showing the percentage value of the duplicated labels (E1, E2). However, experts with longer learning curves requested simpler operations and more exploration guidance from the tool, such as simplified projection view (E3) or "*pop-up reminders...to remind people what these different tools are for in a really obvious way*" (E4). How to support more delicate label manipulation as well as ensure the accessibility of LabelVizier ((6.3.2)) is an inspiring topic that we will discuss in Sec. 6.8.

Visualization. Interestingly, experts with different backgrounds and experience using (semi-) automatic annotation tools also showed different preferences towards our two major visualization components – the Label Investigation View and the Record Projection View. Though all experts expressed their favor of the Label Investigation View, saying they "particularly like the chord diagram" (E3) because it was "*very helpful*"(E1), "*intuitive enough*" (E2) and they "*haven't seen labels presented in this way*" (E4), E2 mentioned "*the co-occurrence is less useful because I don't have enough flexibility to dive down into why there's that co-occurrence*." For the Record Projection View, experts knowing more about machine learning (E1 and E2) picked it up faster and appreciated its value in finding wrong and missing labels better. "*I think the projection view is super useful*," said E2. They "*would like to see even more options of projection spaces and be able to play around with those*." In contrast, E3 felt the same function was too complex and required "*a lot of playing around*." E4 didn't even get a chance to try out the different projection options because of the time constraint.

Interaction. During the study, we received precious interaction improvement suggestions, including more cross-view coordination, operation history tracking, and typ-

ing suggestions. E1, E2, and E4 suggested more flexible interactions, such as cross-view Boolean operations and subset highlighting. For example, when E2 inspected the label "*time*," they mentioned that this label might involve different types of redundancy according to their prior knowledge about the dataset. As a result, they requested Boolean operations between the Label Investigation View and Record Projection View to sift those records of their specific need. E3 and E4 suggested providing ways to keep track of the editing history, such as an undo function (E3), a history list (E4), and some hints of what the user has just clicked (E4). E2 suggested adding typing suggestions for relabel tab, such as auto-complete or alternative recommendation functions. These suggestions reflected the experts' tacit knowledge gained from their long-term annotation practice and will direct us to a more accountable annotation tool in the next development iteration.

6.8 Discussion

The observations and feedback from the expert reviews indicated that LabelVizier provides a means for domain practitioners to validate and relabel the technical text annotations "*quickly*" and "*pleasantly*". They also suggest potential future work for our workflow and tool. Below, we organize the lessons learned.

Accessibility v.s. Functionality. We observed in our expert review that users with less machine learning and (semi-) automatic annotation tool experience may go through longer learning curves with LabelVizier. They requested more exploration guidance or relabelling recommendations when using the tool, while the other group of users requested more complex functions, saying LabelVizier was "*good to find gross errors, but not for perfectionism*"(E2). This is understandable because we required domain experts with diverse backgrounds to learn a relatively complex system within a limited time. We plan to alleviate this problem by leveraging user modeling techniques [141] to analyze the user behavior and guide them to start from different levels of complexity. This way, it will also be safe to extend LabelVizier with more intricate functions, as recommended by the domain experts.

Automation v.s. Human Trust. As computer science researchers, we tended to incorporate more automation in LabelVizier during the development process, which was discouraged by our collaborators with technical text annotation backgrounds. We observed that most of the data analysts tended to be "*over conservative*" (E2) and had to closely check the raw text before "*starting to believe the systems is working*"(E1). They also said that many cases were ambiguous, so they tended to examine more context before making relabelling decisions. Because of this, LabelVizier currently still involves considerable manual work, as demonstrated in Sec. 2.6. LabelVizier also only provides recommendations and explanations instead of one-step relabelling suggestions to supply users with a comfortable amount of information. Indeed, there were no complaints about too much manual work during the expert review but we did receive praise that our tool helped "*focus their energy*"(E4).

Application Domains. LabelVizier was originally designed to serve as a component of technical language processing, but it is generalizable to other annotation verification tasks. The error profiling process can take any natural language descriptions and their labels as input and allow users to perform validation and relabelling via the interface. If label categories are available, as was for our use cases (Sec. 6.3.1), there will be two layers in the Sunburst diagram of Label Investigation View. Otherwise, the Sunburst diagram will devolve into a pie chart, with other functionality remaining unchanged.

6.9 Summary

In this chapter, I presented LabelVizier, a human-in-the-loop workflow that can help domain experts efficiently validate and improve the quality of multi-labeled technical text annotations. LabelVizier utilizes a web-based interactive notebook to enable flexible data processing and model training, and integrates a visual analytic system to leverage human knowledge in annotation relabeling. The interface coordinates different visual components for multi-type error detection (duplicate, missing, and wrong labels) in different dataset scopes (corpus level, sub-group level, and record level), and provides a human-centered solution targeting the quality enhancement for large-scale

text annotations. I demonstrate the usability of LabelVizier via two use case cases, and four experts evaluated the effectiveness of our workflow through a study consisting of one-on-one qualitative evaluations. I believe this work will encourage the design of visual analytics for other domain-driven problems and inspire future research efforts in creating higher-quality annotations for larger-scale text datasets.

Chapter 7
Knowledge Exploitation for Machine Learning Model Validation

Real-world machine learning applications need to be thoroughly evaluated to meet critical product requirements for model release, to ensure fairness for different groups or individuals, and to achieve a consistent performance in various scenarios. For example, in autonomous driving, an object classification model should achieve high detection rates under different conditions of weather, distance, etc. Similarly, in the financial setting, credit-scoring models must not discriminate against minority groups. These conditions or groups are called as "*Data Slices*". In product MLOps (Machine Learning Operations) cycles, product developers must identify such critical data slices and adapt models to mitigate data slice problems. *Discovering* where models fail, *understanding* why they fail, and *mitigating* these problems are therefore essential tasks requiring the knowledge from domain experts to steer the MLOps life-cycle. In this chapter, I present SliceTeller, a novel tool that allows users to *leverage their domain knowledge* for debugging, comparing, and improving machine learning models driven by *critical* data slices. SliceTeller automatically discovers problematic slices in the data, helps the user understand why models fail. I also present an efficient algorithm, *SliceBoosting*, to estimate trade-offs when prioritizing the optimization over certain slices. Furthermore, our system empowers model developers to compare and analyze different model versions during model iterations, allowing them to choose the model version best suitable for their applications. We evaluate our system with three use cases, including two real-world use cases of *product development*, to demonstrate the power of SliceTeller in the debugging and improvement of product-quality ML models.

Figure 7.1: SliceTeller applied to the comparison of two machine learning models (*ResNet50* and *GroupDRO*) for hair color classification (gray hair, not gray hair), trained on the CelebFaces Attributes Dataset (CelebA). (A) Slice Matrix: The data slices (represented as rows), slice descriptions (encoded as columns), and slice metrics (Support and Accuracy). Slices are sorted by model accuracy. (A - *Tooltip*) Confusion matrix for Slice 1. (B) Estimated effects of optimizing the model for two data slices (Slices 1 and 2, highlighted in blue). (C) Accuracy comparison between the two models, *ResNet50* and *GroupDRO*. (D) Slice Detail View containing image samples from a data slice. (E) Slice Detail View containing the comparison of two data slices using the *MatrixScape* visualization. (F) System menu, containing options for model selection, effect estimation of focusing on a slice during model training, and data slice summarization.

7.1 Introduction

Recently, Machine Learning (ML) has been used in a variety of critical applications, including autonomous driving, medical imaging, industrial fire detection, and credit scoring [150, 185, 236, 270]. Such applications need to be thoroughly evaluated before deployment to assess model capabilities and limitations. Unforeseen model mistakes may cause serious consequences in the real world: for example, a false sense of security in ML models may cause safety issues in driver-assistance [150] and industrial systems [185], misdiagnoses in medical analysis [270], and biases against minorities [236].

During our collaborations with MLOps (Machine Learning Operations) engineers for product-quality model development, we have identified that the evaluation of crit-

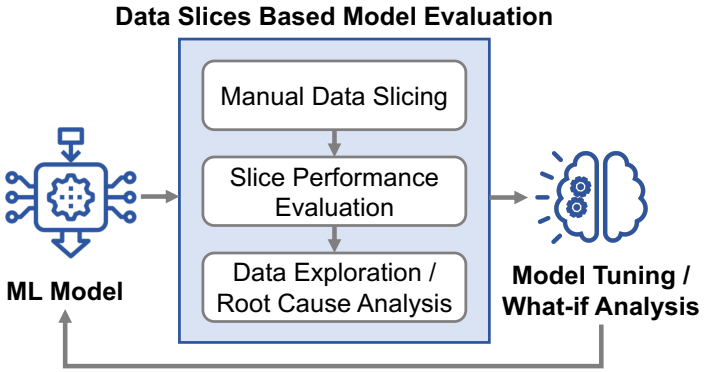


Figure 7.2: Product MLOps engineer's workflow for model validation and iteration over critical data slices. Experts sliced their data based on product and domain requirements, computed model performances per slice, and explored the data to identify the root causes for potential model mistakes. Based on these observations, they would iterate over the model, by retraining while re-prioritizing certain data slices over others.

ical ML models is usually conducted beyond the aggregated level (e.g., a single performance metric). Instead, they need to thoroughly evaluate model performance on carefully specified usage scenarios or conditions in order to meet important ML product requirements. Based on this analysis, experts can then take actions to 1) attempt to make the model more robust to various conditions and 2) make customers aware of model limitations in certain conditions, aiding in the development of mitigating measures. During the evaluation of ML models, model developers often have to slice their data based on the specified product usage conditions, to ensure satisfactory performance under such critical conditions. For example, in the autonomous driving setting, experts need to ensure high detection rates for multiple environmental conditions, such as sunny weather and rain, and specific object types, such as cars and pedestrians. Figure 7.2 shows a common workflow for critical model analysis and iteration.

Challenges on Model Evaluation and Iteration. While the slice-based analysis is essential for the critical applications, this approach has several limitations. 1) Manually slicing the data is a very time-consuming task. In our interviews, experts mentioned that this task involved manually creating rules to slice the data, running evaluation scripts on the data subsets, and comparing the results on various data subsets. 2) ML experts cannot explore all possible data subsets to identify relevant failures cases for their application. Data slices can be created by any number of interpretable meta-data

(e.g., weather and temperature for autonomous driving), resulting in an exponentially large search space. Therefore, they must rely on domain-specific priors to select what meta-data they will slice the data based on. 3) Once the critical failures are identified, experts have the options to either collect more data to cover the weakness scenarios or retrain their models by prioritizing the critical slices. While the former requires additional investment on data collection, the latter is usually time-consuming, particularly for training neural network architectures. Moreover, it is unclear how the new model will trade-off performance on other slices and whether the result can still meet the product requirements.

Our Approach. We develop SliceTeller, a novel data slice-driven model validation tool that automates slice finding, enables slice-based model validation and comparison, and allows what-if analysis for slice prioritization. Our tool takes as input the data (non-interpretable features), metadata (interpretable properties that can be used to slice the data), dataset labels and model predictions. A state-of-the-art slice-finding algorithm is adapted to find slices on the data for which a performance metric (for example, accuracy) is significantly different from the overall model metric. Once these slices are found, we use a binary matrix graphical encoding to show the data slices compactly, as well as metrics for these data slices (4.2). We also provide various visualization to help users understand and interpret these data slices. After a data slice of interest is found, users can estimate the performance impact of optimizing the model for this data slice using our effect estimation algorithm, SliceBoosting. Finally, users can quickly find problematic data slices on their model and derive actionable insights to improve the results according to their product requirements in a next training iteration.

The main contributions of this work include the following:

1. A novel Visual Analytics (VA) tool, SliceTeller, for the evaluation and comparison of ML models using a data slicing approach. SliceTeller combines data slice finding, together with effective visual representations for data slices and model metrics, to facilitate the iteration and comparison of ML models.
2. An efficient algorithm, SliceBoosting, for the quick estimation of data slice trade-

offs during model training to improve model iteration efficiency. This algorithm estimates the performance effect of focusing on one or more data slices during model training, highlighting potential trade-offs between data slice optimization and overall model performance.

3. Three use cases, including two real-world use cases of product development, to demonstrate the effectiveness of our approach in assessing, comparing, and iterating over ML model results. We believe our method and design process together with product R&D partner and MLOps Engineers can benefit the practice and research of using VA for model validation at large.

In summary, SliceTeller, is a novel VA tool for ML model validation with a data-slice driven approach. This tool is model-agnostic and can be easily plugged in MLOps life-cycles. This work also resonates with recent data-centric AI trends focusing on data instead of models. We hope this work can inspire more research questions innovating VA approaches to address data-driven model validation challenges.

7.2 Related Work

7.2.1 Slice-Oriented Model Validation

Finding the subgroups of specific quality is one of the classical combinatorial optimization problems, named Constraint Satisfaction Problem (CSP) [354]. This problem is defined as slice finding in the context of ML model evaluation, the aim of which is to identify the data subgroups over which the ML subgroups underperforms [19].

Most of the existing solutions in commercial tools use greedy heuristics to balance the tradeoff between searching speed and accuracy. For instance, the FreaAI in IBM IGNITE [3], Slice Finder in Tensorflow [67, 68], Amazon Sagemaker [207], and RobustnessGym [128] are built upon heuristic techniques such as clustering, self-defined metrics (e.g., highest posterior density), model-based, and rule-based data slicing. Efficient as they are, heuristic solutions could miss critical data slices if they fall into the blind area of the heuristic rules (e.g., the slice size is too small). Thanks to the development of parallel computing devices, researchers could solve this problem by using

exhaustive searching with a reasonable time cost. For example, SliceLine includes size and score pruning to boost the performance of exhaustive search, which can be easily deployed onto the parallel devices [303]. DivExplorer [274–276] enumerate the data lattice to look for all candidate itemsets with the highest divergence and support the customization of itemset size. Since such flexibility would allow us to let the users decide the slices they are interested in, we choose DivExplorer as the slice finding tool for our system. The research work in this direction is still in the exploratory stage, and there is an increasing trend of introducing ML for solving the classical combinatorial optimization problems [163, 230].

However, these approaches mainly focus on search efficiency and scalability, but largely ignore how to understand and interpret the impact of these data slices. Also, it is a non-trivial task to customize and select data slices of interest based on domain-specific requirements. This is where our VA-based approach can help.

7.2.2 Model Robustness over Data Groups

ML robustness research focuses on achieving consistent model performance under various conditions. Due to data collection, sampling or annotation biases, ML models trained on real-world data tend to identify *spurious correlations*, connections that appear extensively in the training data, but do not hold in novel scenarios [193, 253]. Under distribution shift, models that rely on these spurious correlations often degrade significantly in performance [122, 240].

One powerful approach to alleviate this issue is group distributionally robust optimization (DRO) [161, 268]. Group DRO utilizes prior knowledge of spurious correlation to define groups over training data, and optimizes the worst-case loss among the groups, instead of the expected loss of the entire distribution. As a result, the trained model is capable of significantly improving minority group performance while maintaining similar performance over the majority groups. In [304], the authors investigate group DRO in the context of over-parameterized deep neural networks (DNNs). They discovered that as opposed to shallow ML models that can directly benefit from group DRO, DNNs require strong regularizations during training to achieve minority group

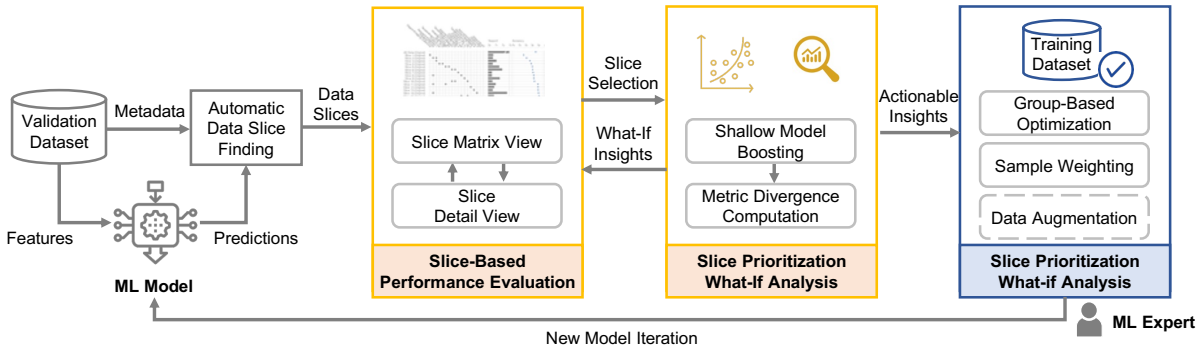


Figure 7.3: Workflow of model analysis and improvement using SliceTeller. The validation data, together with the model predictions, are used for automatic slice identification. The produced data slices can be explored using our VA solution (Slice Matrix and Slice Detail View). Users can prioritize groups of data slices and quickly evaluate the effect of this action on the rest of the model slices. Finally, experts can use the insights gained from the system to fine-tune the model and continue the analysis with SliceTeller.

improvements. Recent works along the direction of group DRO research incorporates instance-level weighting to tackle imperfect group partition [405], leverages human annotation to discover and optimize over unmeasured variables [334], and scales up the optimization method for large-scale problems [199].

In this work, we utilize distributionally robust deep neural network described in [304] as the slice optimization model. Our VA framework addresses the following core limitations of group DRO for practical usage: a) Group DRO requires that the group information of training data is known beforehand. Although empirical results have provided evidence of spurious correlations in popular public datasets [193, 268], such insights are not readily available for most real-world ML problems. b) Group DRO training is time-consuming. It is therefore infeasible to exhaustively apply group DRO on all combinations of data slices in order to determine the ideal slices for optimization. By integrating slice discovery, effect estimation and model optimization under the same VA framework (Figure 7.3), we help users identify problematic slices, understand the optimization trade-off among slices, and eventually optimize the model, thus significantly speeding up model iteration.

7.2.3 Visualization for Slice-Based Model Optimization

There are various VA techniques supporting the creation and analysis of data slices [83, 177, 392]. In the context of model exploration, CoFact [174] is a VA system that helps users explore counterfactual subsets and thus understand the confounding facts in large and complex datasets. CoFact is close to the first part of our work, but its focus is on spurious feature correlations, not providing any model or data optimization recommendations. FairVIS [45] and Manifold [396] also allow data subgroup analysis based on human interaction, but do not provide automated methods for data slicing. Finally, Errudite [383] provides a domain-specific language for data grouping with unstructured text data analysis, while we mainly focus on structured data.

Most of these works focus on the data exploration stage and rarely provide optimization solutions to close the entire loop. Therefore, we provide a VA solution for the entire loop of slice-based model optimization that supports slice finding, model optimization, and model comparison with a Matrix visualization inspired by UpSet [201] and PipelineProfiler [263], along with carefully-designed interactions driven by the domain requirements described in Sec. 7.3.1. To the best of our knowledge, our system is the first of its kind to provide an end-to-end solution for users to identify, understand and optimize model performance on problematic data slices.

7.3 SliceTeller

In this section, we describe SliceTeller, a system for data slice-driven validation of ML models. We first present a desiderata for our system, distilled from interviews with four industry partners from the product R&D team. Next, we describe the visual components of our system. Finally, we present SliceBoosting, an algorithm that can estimate performance divergences after a slice-based model optimization.

7.3.1 Data and Domain Requirements

SliceTeller was developed based on a collaborative product project with three industry MLOps engineers over the period of six months. We received continuous feedback from our partners, who worked on tabular and image-based classification problems. These

experts work on products in critical applications, including autonomous driving and fire detection for building security. Therefore, they often conduct extensive validation of their models before deployment to production.

In the context of autonomous driving, experts were interested in modeling the ultrasonic sensors to understand the car surroundings. It is a critical modality in the sensor-fusion pipeline to enhance the overall system robustness. The raw ultrasonic sensor data are not directly interpretable by human. However, every sample also contained metadata describing the experiment setup, for example, the object type, distance, sensor location, time of day, etc. Experts had trained a tree boosting model to classify nearby objects' heights (as "high" or "low") using the sensor-derived tabular features. While evaluating their models, they wanted to make sure that certain critical objects had a low error rate. In some cases, they would trade-off between the performance of non-critical objects for the performance of the critical objects. For instance, children, curbstones, and nearby cars should have the highest priority. Therefore, in every evaluation iteration, they had to slice the data, evaluate the model on the data subsets, and retrain the model with different parameters to mitigate critical mistakes. Experts mentioned that these tasks were tedious and time-consuming.

For the fire detection application, experts trained a deep neural network to detect smoke and fire on video frames. In this setup, every video segment was associated with interpretable metadata that described the video collection process in detail, for example, the recording location, time of day, the smoke density, and whether there were blinking lights in the scene. While the overall performance of this model was high, experts were interested in identifying situations where it failed. Therefore, they spent a large amount of time inspecting the model and using the video metadata to identify these situations. Transparency with customers was a high priority for model release: they wanted to clearly communicate the model capabilities, where it was effective and where it failed. Furthermore, they wanted to understand why the model was failing, and what were the possible confounding features on their dataset.

While these two applications used different techniques and data types, the experts

developing them shared a common goal. 7.2 shows the model evaluation workflow derived from our interviews. In both cases, experts desired to slice the data into various scenarios, thoroughly evaluate their models, understand the failure cases, and develop strategies to tune the models to improve performance. They noted that this workflow usually took several iterations, requiring significant effort and trial-and-error. Based on these observations, we have compiled the following desiderata for a data slice-driven model validation system:

1. Data Slice Finding and Overview: Users often spend a significant amount of time slicing the data based on certain heuristics to learn the boundaries of the model. Our system should automatically identify these data slices in the validation dataset and present an overview to the user.
2. Slice-Based Data Understanding and Valuation: Users should be able to explore the data slices in order to understand them and value how critical they are. The system should allow the user to explore the data, model metrics, and distributions to explain these scenarios.
3. Slice-Based Model Optimization: In critical applications, experts need to trade-off the performance of certain scenarios in order to focus on critical use cases. To do so, they need to train models from scratch, which can be very time-consuming. The system should enable the quick experimentation with the slice-based model optimization, highlighting possible trade-offs in the data.
4. Slice-Based Model Comparison: Users need to train and evaluate multiple models in order to tune parameters and mitigate problems. However, this comparison is usually done at the aggregated level (e.g., a single metric value). The system should allow the comparison of model performances at the slice level, facilitating the identification of trade-offs between data slices.

7.3.2 System Workflow

In order to fulfill the requirements identified in the previous section, we developed SliceTeller, a system that tells a story about the evolution of ML models from the perspective of data slices, allowing their evaluation, exploration and comparison. Section

7.3 shows the general workflow for model analysis and improvement with SliceTeller. The input to SliceTeller is the validation dataset consisting of validation data (e.g., raw images or tabular features extracted from the sensor signals), metadata (interpretable features that can be used to slice the data), and ground truth labels (e.g., object classes or obstacle height). Note that we use a validation dataset for model analysis instead of training data since it is unseen by the model. In the case of model overfitting, we observe all slices in the training data having high accuracies. Given the input, the system works as follows:

1. First, the system uses an automatic slice finding algorithm to identify data slices where the performance measures (e.g., accuracy) are the most different from the overall model performance. We use the DivExplorer algorithm [275], a Frequent Pattern Mining-based approach for this task (Section 7.3.3).
2. Next, a VA system allows the users to quickly visualize and summarize the produced data slices using the Slice Matrix View (where rows correspond to slices, and columns, to slice descriptions and associated metrics). The user can select slices to view its details using the Slice Distribution View, which can present metadata distributions and correlations to the user. These two views allow the user to identify critical slices in the data, i.e. slices where the model performance has issues (Section 7.3.4).
3. When a critical slice is found, the user can test mitigating measures using the 'Slice Prioritization - What-If Analysis' tool. This tool uses our SliceBoosting algorithm to evaluate the effect of optimizing the model for particular data slices. SliceBoosting fits a shallow boosting model on top of the original model to estimate the effect of prioritized optimization (Section 7.3.5).
4. Finally, when the user found a group of slices to optimize, they can export the selected slices back to their programming environment, make changes on data, hyperparameter or model, and insert the new model back into SliceTeller to compare models (Section 7.3.6).

7.3.3 Data Slicing

We begin our analysis by automatically finding problematic data slices using the DivExplorer algorithm [275] (illustrated in 7.4). The algorithm takes the model predictions and the meta-data (interpretable features of the dataset) as input, and executes an exhaustive slice search by frequent pattern mining [144]. The *minimum support* (i.e., minimum slice size) is defined as a parameter by the user. Then, a model metric such as accuracy is computed for every data slice found.

To reduce the searching time for datasets with a larger number of metadata features, we conduct a two-iteration slice finding procedure using DivExplorer. First, we run DivExplorer with a large minimum support to identify the relevant metadata features which are most correlated with poor performance. Then, we run DivExplorer again using the relevant metadata features, this time with a lower minimum support to perform a more fine-grained search. The level of granularity (minimum support) can be fine-tuned by the user in order to find the relevant slices for their model. For example, users can fine-tune the parameter to find slices with sufficiently high support and low performance (such attributes are problem-specific and user-defined).

The DivExplorer algorithm can find an exponential number of data slices (exponential in the number of unique feature-value pairs). Therefore, we use a summarization approach to reduce the number of slices to be explored by the user 1. We allow users to summarize data slices with a redundancy pruning approach [275] (slider in 4.2(F)). If the introduction of an item α (e.g., $Weather = Sunny$) in a slice S will cause an absolute performance change below a redundancy threshold ϵ, only the slice without α (denoted $S \setminus \{\alpha\}$) will be presented to the user. This guarantees that the more general slices can be investigated first. More specifically, let p be the function that computes the performance score on a data slice. A data slice S is pruned if: $|p(S) - p(S \setminus \{\alpha\})| < \epsilon$.

7.3.4 Visualization Design

4.2 demonstrates the visualization design of SliceTeller. The main visualization components of SliceTeller are the Slice Matrix (A) and the Slice Detail View (D-E). The Slice Matrix shows a summary of all the data slices with a performance metric that diverges

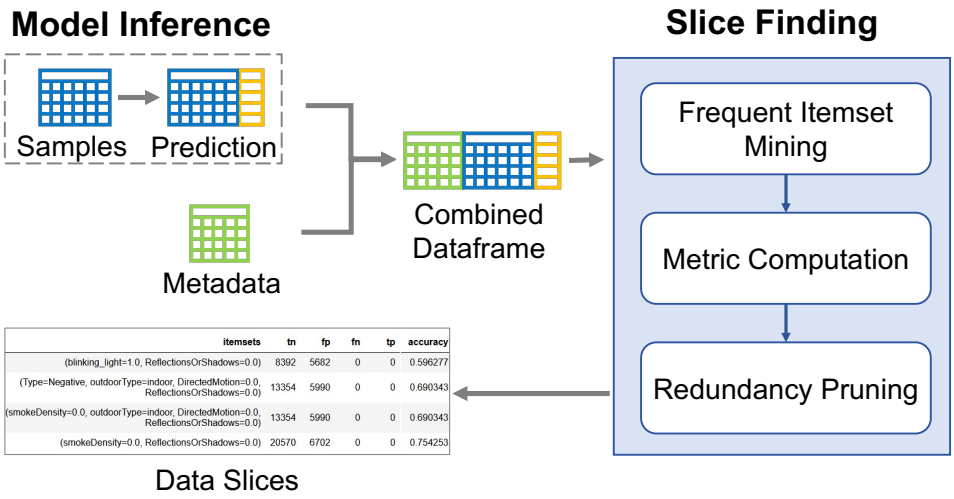


Figure 7.4: Data slicing workflow. It takes as input model predictions combined with interpretable metadata and utilizes frequent itemset mining to automatically identify the most critical slices. It then performs slice merging and redundancy removal to generate concise data slice results.

from the overall model. The user can drill down on the slices in order to explore one or more data slices simultaneously using the Slice Detail View. The System Menu (F) allows users to switch between data slices from the multiple ML models, summarize model slices and perform What-if analyses to estimate the effect of optimizing the model for a particular data slice. These operations are described later in this section.

7.3.4.1 Slice Matrix

The Slice Matrix (4.2(A)) provides an overview of the problematic data slices to the user 1. First, the data slices are identified using DivExplorer [275], described in 7.3.3. After the data slices are found, they are graphically represented using *Slice Matrix*, an adaptation of the UpSet [201] matrix encoding, where sets are represented as rows, and set members, as columns. In the context of data slices, we use a similar encoding where each slice is represented as a row (set), and slice descriptions (items), as columns. Our adaptation also includes data slice metrics on the UpSet visualization encoding.

Model and data metrics are computed and displayed together with their respective slices (2 and 4). For model-agnostic metrics, a bar chart is displayed and, for model-specific metrics, a color-coded 1-D scatter plot is shown. In 4.2(A), the metrics "Support" and "Accuracy" are displayed. We show a truncated scale for "Support", since the support of the entire dataset (slice "All Data") is always equal to 1. Additional metrics can

be defined, including “Precision”, “Recall”, and “F1 Score”. Previous VA systems have enabled users to interactively compare ML models [56, 80, 124, 263, 388, 396]. However, to the best of our knowledge, SliceTeller is the first of its kind to allow a detailed model comparison using automatically computed data slices to guide the analysis process.

7.3.4.2 Slice Detail View

During our interviews, experts expressed their interest in understanding the content of the data slices and valuing how much impact they have in their application 2. We design two types of slice detail view to cater two major data types in our use cases: a *Slice Distribution View* (7.8(D)) for tabular data, and a *MatrixScape View* (4.2(D-E) [150]) for image data.

Slice Distribution View. In this view, users can select the metadata to understand the distribution shifts across slices. We present the distribution of each metadata feature as a sorted histogram and align those for the same feature of different slices to facilitate a more convenient comparison. For example in 7.8(D), two data slices are selected (“All Data” and “Slice 1”) and “Object” metadata distribution is shown.

MatrixScape View. While looking at distributions was useful, the experts working with image data wanted to be able to explore the images themselves. In particular, they wanted to check whether they could identify potential sources for model mistakes in these samples. To allow this task, we used the MatrixScape visualization [150], which can contextualize images with metadata information. In MatrixScape, images can be laid out in a canvas according to different metrics and aggregated at multiple levels of detail. At the coarsest aggregation level, MatrixScape shows a heatmap of a particular data metric (for example, accuracy), grouped by metafeatures chosen by the user (4.2(E). Upon zooming in, users can see individual data samples as well (4.2(D)).

7.3.5 SliceBoosting: Estimating the Effect of Data Slice Optimization

During our interviews with the domain experts, they expressed the need to create multiple models and evaluate trade-offs between them 3 from the perspective of manually created data slices. In their current approach to train multiple models from scratch for comparison, there are mainly three pain points: 1) Since the model training pro-

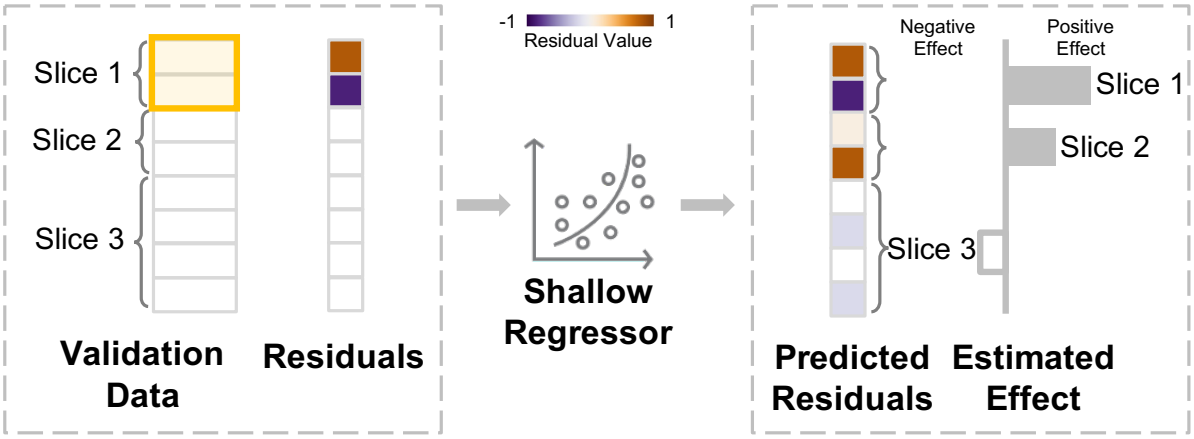


Figure 7.5: Illustration of SliceBoosting algorithm to estimate model optimization effect. Given the selected slice 1, we train a shallow regressor to estimate that, under the ideal scenario where the optimization model correctly fits to slice 1, how will the performance on slice 2 and 3 be affected. We design the prediction target of the regressor as the residuals of the original model predictions to the ground truth validation labels. To focus on the effect estimation of slice 1, we set the residual values only for slice 1, while keeping the residuals of all other slices as 0. The predicted residuals from the regression are in the range of $[-1, 1]$. We then aggregate the sample-level residual predictions to obtain slice-level estimation results: how will the accuracy on these slices increase or decrease?

cess is time-consuming, the experimentation cycle is easily interrupted, and the model iteration is slow. 2) It requires significant efforts from the experts to keeping track of multiple models trained across different data slices. 3) To draw experimentation conclusions and identify the slice trade-offs, they have to switch between development tools multiple times.

Denote the original input model to SliceTeller as f parameterized by θ. Let the training data be $X^{train} \in \mathbb{R}^{N^{train} \times D}$, where N^{train} is the number of samples in training set and D is the feature dimension. Similarly, let the validation data be $X^{val} \in \mathbb{R}^{N^{val} \times D}$. We use S^{val} to denote the slices selected by user as in Section 7.3.4, and S^{train} to denote the training data slices that correspond to the **same description** as S^{val} (e.g., $Weather = Sunny, Object = Wall$). We can utilize the optimization approaches (details in Section 7.3.6) to retrain f on X^{train} to prioritize on S^{train}, in order to obtain the optimized model f'. However, due to the scale of X^{train} and the high complexity of f, the optimization is time-consuming. It is therefore infeasible to try out different slice combinations to obtain the optimal f' that could satisfy the product requirements.

To facilitate fast slice-based experimentation, our objective is to estimate the per-

formance difference between f' and f without explicitly training for f'. We develop a novel SliceBoosting algorithm to perform the estimation. The main idea is that instead of training the full model to evaluate slice trade-offs, we can train a shallow model to approximate the residuals (errors) of the slices, in an approach similar to boosting [175]. We denote the shallow model as h. Due to the shallowness, the training process is significantly faster than training the full model from scratch.

We have two assumptions: 1) The validation set X^{val} has a similar distribution to the training data X^{train} while being significantly smaller. This allows us to train the shallow model on the validation set to approximate the full model behavior on the training set. This assumption is valid in most cross-validation experiment settings. 2) The optimization approach (details in Section 7.3.6) is sufficiently powerful to steer the model to make correct predictions on the selected validation slices. Under these assumptions, we train the shallow model to fit to the selected validation slices S^{val} together with the associated labels. After it is trained, its predictions on other slices will contain the approximation of the full model's behavior with further optimization. Similar approaches have used weak learners to reduce model biases and improve performance, including Multicalibrated Predictor [151] and MultiAccuracy Boosting [183]. However, while these approaches train and evaluate multiple boosted models to improve model accuracy, SliceBoosting uses a simplified approach with a single boosted model to estimate the effect of subgroup optimization and allow quick experimentation.

Since the shallow model is a "weak learner", it is challenging to encode all validation data and labels. Inspired by gradient boosting [175] and surrogate model explanation approaches [71,72], we design the shallow model to instead fit to the residuals (errors) of the original model on the selected slices. Since the original model is powerful (e.g., ResNet-50 Deep Neural Networks), its prediction is close to ground truth labels. Therefore, predicting the residual is a significantly easier task for the shallow model. As shown in Figure 7.5, the residual is calculated as the difference between the ground truth validation labels and the predicted labels, in one-hot-encoded format:

$$\text{residual}_i = y_i^{val} - \hat{y}_i^{val} \tag{7.1}$$

where y_i^{val} denotes the one-hot-encoded ground truth validation label for sample x_i^{val}, and $\hat{y}_i^{val}$ denotes the one-hot-encoded predicted label from the original model f. We illustrate the residual for a certain class in Figure 7.5. There are three possible values in the residuals calculated from Eq. (7.1): 0 denoting the model prediction is correct, 1 denoting the model missed the detection of the class, and -1 denoting the model wrongly predicted the class. Note that since we focus on the selected slices, samples from all other slices have residual of 0.

We then train the shallow regressor h using XGBoost [57] to learn the residuals from S^{val}. To achieve fast response for visual interaction, we use only 3 training iterations and maximum tree height 5 in XGBoost (these parameters can be fine-tuned depending on the problem). To emphasize on the small set of misclassified samples, we increase their weights in the loss function. After h is trained, we infer the residual and prediction label for the full optimized model ($\tilde{y}_j^{val}$) as follows:

$$pred\_residual_j = h(x_j^{val})$$

$$\tilde{y}_j^{val} = pred\_residual_j + \hat{y}_j^{val}$$

Here, x_j^{val} contains data features of the validation set belonging to Slice j. A good estimation is achieved if $\tilde{y}_j^{val}$ is close to the true label y_j. After obtaining all estimated predictions, we measure the new accuracy in each slice and compare it with the original model accuracy to determine final estimated effect. More specifically, let A be the vector containing the accuracy of the original model on all data slices, and A' be the vector containing the accuracy of the boosted model f' on all data slices. The estimated effect E' is given by $E' = A' - A$. As illustrated in Figure 7.5, in the estimation effect, a positive number indicates that the performance on the slice might improve with model optimization. On the other hand, a negative number suggests that the performance on the slice could be reduced.

In order to evaluate SliceBoosting, we can check whether its estimated effects agree with the real optimized model performance (outlined in Section 7.3.6) . We measure this *Agreement Score* using Pearson correlation coefficient [311] for the first two use cases in Section 7.4. More specifically, let A be the original model accuracy on all data

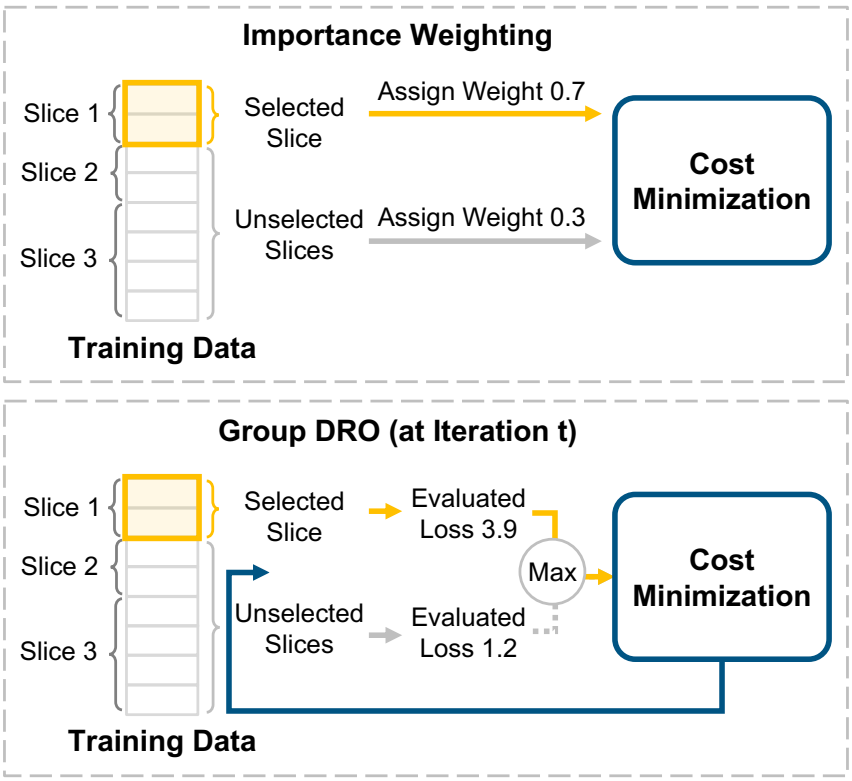


Figure 7.6: Illustration of the model optimization methods considered in our work. During re-training, they prioritize slices in the training data according to user's decision.

slices and A'' be the retrained model accuracy on all data slices. The real performance effect E is given by $E = A'' - A$. We compute the Pearson correlation coefficient between the estimated effect E' and the real effect E as: $Agreement\ Score = corr(E, E')$. We note that in both use cases, the Agreement Score was greater than 0.8, showing high correlation between the estimated slice optimization effects and the real effects. We further validate the SliceBoosting algorithm by evaluating the Agreement Score of ten estimated effects, computed for the top five worst data slices of the two aforementioned use cases (a new estimate and model are computed for each data slice). In Case 1, the estimates for five models optimized on the five worst slices had an Agreement Score of 0.860 ± 0.050. In Case 2, the estimates for five models optimized on the five worst slices had an Agreement Score of 0.776 ± 0.054. A detailed description the SliceBoosting evaluation is available in Appendix A.

7.3.6 Model Optimization

In this section, we utilize state-of-the-art model optimization methods to improve the performance on the selected slices, while minimizing the trade-off for the averaged model performance on the entire dataset 3. These methods adapt the loss function based on identified slice prioritization and subsequently perform additional training to steer the model towards the user requirement. Note that our framework is compatible with data-centric model improvement strategies as well (e.g., additional data collection and data augmentation/synthesis) and we leave the discussion to Section 6.8. Here, we focus on optimization-based model improvements without any change of the dataset.

Figure 7.6 illustrates the model optimization methods in SliceTeller, i.e. importance weighting and group DRO. Note that we merge all unselected slices into a single slice for optimization. In general, importance weighting method changes the loss function by assigning heavier weights to the training samples in the worst-performing slices. On the other hand, group DRO prioritizes the worst-performing slices during the training process. Here we briefly describe the two approaches. Detailed descriptions can be found in [304].

Importance Weighting. Importance weighting modifies the expected loss by emphasizing training samples belonging to the slices S^{train}. Denote the number of samples in S^{train} as n^{train}, the number of samples in the training set as N^{train}, and the total number of slices as M. The weight for slice S is calculated as:

$$w_{S^{train}} = \frac{N^{train}}{M \times n^{train}}$$

Intuitively, the selected slices with lower performance correspond to the minority groups in training set. We can therefore specify the weights of the slices as inverse proportional to the respective slice size. Then, the modified expected loss can be defined as follows [304]:

$$\mathbb{E}_{(x^{train}, y^{train}, S^{train}) \sim P}[w_{S^{train}} l(\theta; (x^{train}, y^{train}))] \tag{7.2}$$

where P is the distribution of training data X^{train} and l is the loss.

Group DRO. Compared to importance weighting that upweights the selected slices by heuristic rule, group DRO adopts a different optimization scheme. Instead of opti-

mizing the averaged loss over entire training data, it optimizes for the worst-case loss over the groups in the training data. Specifically, the expected loss is defined as [304]:

$$\max_{S^{train}} \mathbb{E}_{(x^{train}, y^{train}) \sim P_{S^{train}}} [l(\theta; (x^{train}, y^{train}))] \tag{7.3}$$

During training, the optimization can be conducted by either recording the historical losses of all groups [268], or utilizing gradient ascent [304].

7.4 Use Cases

In this section, we present three use cases to demonstrate the power of SliceTeller on the analysis, validation and improvement of ML models. Case 1 shows how a fictional user would validate an image classification problem for data biases. The next two case studies are taken from our interviews with three MLOps engineers who work on real industry products. Case 2 shows how one engineer working on an ultrasonic object height classification model can use our system to improve their models on critical data slices. Finally, Case 3 shows how two MLOps engineers used our system to explore an image-based fire detection model and identify potential data issues. Fig. 7.1 shows a summary of the three use cases.

Table 7.1: Use Case Summary

	1) Bias Detection	2) Height Classification	3) Fire Detection
Data Type	Image	Tabular	Image
Validation Size	40,520	47,322	126,912
Input Metadata	40 binary	8 numeric / categ.	7 numeric / binary
Target (Binary)	Yes / No	High / Low	Yes / No

7.4.1 Case 1: Bias Detection for AI Fairness in Image Classification Models

To showcase how SliceTeller helps ML practitioners detect bias caused by the imbalanced distribution of image dataset, we describe how Hedy, a PhD student interested in computer vision, profiles ResNet50 performance on the CelebFaces Attributes Dataset (CelebA) [219] and identify a gender-related bias inherited in the model.

The CelebA dataset contains 202,599 face images of 10,177 celebrities, along with 40 binary attribute annotations including gender, skin color, smiling, etc. for each image. It is widely used in the computer vision community for tasks including image classification [176].

Table 7.2: Accuracy of Image Classification Models (Val / Test)

Slice	Description	ResNet-50	Group DRO
0	All Data	**0.98** / **0.98**	0.95 / 0.95
1	Gray_Hair=Yes, Male=No	0.65 / 0.5	**0.91** / **0.81**
2	Gray_Hair=Yes, Double_Chin=No, Wearing_Necktie=No	0.65 / 0.69	**0.90** / **0.93**

Hedy first divides the data into three splits: training (70%), validation (20%) and testing (10%). She fine-tunes a ResNet50 model on the CelebA dataset to classify *gray hair* and obtains an overall validation accuracy of 0.98. She wanted to investigate how different models performed over attributes of $Male$ and $Gray\_Hair$. The system initially identified 22 slices for Hedy to explore (using DivExplorer support=0.02). Upon inspecting the model with the matrix view of SliceTeller (Fig. 4.2(A)), she notices that the model can only achieve a low accuracy of 0.65 on Slice 1, which is defined as **females** ($Male = No$) with **gray hair** ($Gray\_Hair = Yes$) (Fig. 4.2). She checks the distribution of the dataset over the dimension of $Male$ and $Gray\_Hair$, and finds that such a significant performance drop is caused by the highly skewed data distribution in this slice (Fig. 4.2(E) left). She verifies this observation by browsing the sample images in Slice 1 (Fig. 4.2(D)). Interestingly, Hedy observes that Slice 2, defined by ($Wearing\_Necktie = No$, $Double\_Chin = No$, and $Gray\_Hair = Yes$), also only achieves an accuracy of 0.65. Although the attribute $Wearing\_Necktie$ seems related to gender bias as well, it is unclear if this is the true cause of the low performance. Therefore, she decides to take a closer look of these two slices by using the effect estimation functionality in SliceTeller.

Hedy starts by conducting a what-if analysis with SliceTeller to estimate the effect

of optimizing upon Slice 1 and 2 jointly. The estimation result from SliceTeller indicates that the model performance on the worst eight slices will all improve significantly if optimizing Slice 1 and 2 together (Fig. 4.2(B)). The improvement is higher as compared to optimizing over Slice 1 alone. Hedy checked the support size of Slice 2 (Fig. 4.2(A)) and its data distribution, then realized that this slice accounts for more minority attributes than females with gray hair. This explains why optimizing Slices 1 and 2 together can improve the performance of the worst eight slices. Such optimization effect appears promising to Hedy, so she retrains a *GroupDRO* model [304] that dynamically optimizes upon the two slices. Sec. 7.2 shows the validation and test accuracy of the two models.

After the optimization, Hedy adds the performance result of *GroupDRO* into SliceTeller to compare it with the previous ResNet50 model. She immediately observes that the new model's performance on the gray-haired female images improve significantly (Fig. 4.2(C)). As predicted in the effect estimation stage, the optimized model's performance on the worst eight slices are improved, with a small trade-off on overall model performance. To confirm that she has alleviated the model bias towards females on classifying *gray hair*, she places the "All Data" slice of both ResNet50 and *GroupDRO* side by side in the MatrixScape view and compares them from the 2D dimension *Male* and *gray hair* (Figure 4.2(E)). The result confirms that the *GroupDRO*'s performance classifying gray-hair females has significantly improved, with slight performance change on the other blocks.

7.4.2 Case 2: Ultrasonic Object Height Classification for Autonomous Driving

We now consider a real-world application: object height detection for autonomous driving. One of our domain experts worked on a model for object height prediction based on vehicle ultrasonic sensors. The sensors produced tabular data has 157,743 records that consists of 71 numerical features, and the expert's goal was to predict object height as a binary label: 'high' or 'low'. Such predictions would help improve the overall robustness of sensor fusion in the model pipeline, preventing collisions and aiding in the decision-making process of the car.

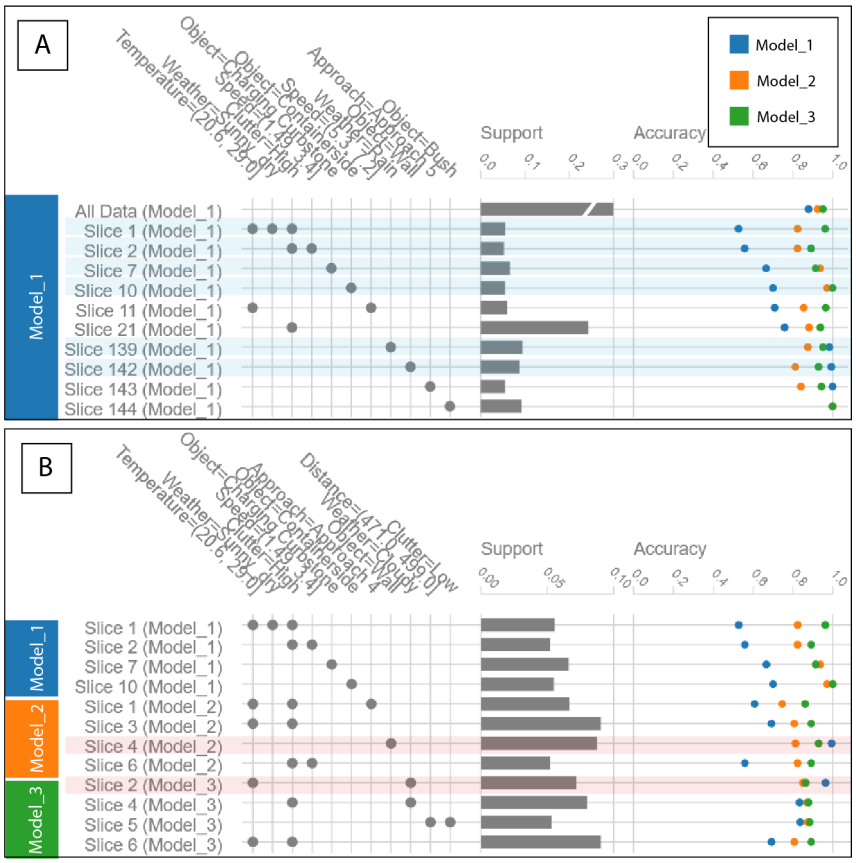


Figure 7.7: A third iteration of the model is added for comparison. (A) Performance of the three models on the worst data slices from *Model_1*. Samples from highlighted slices were more heavily weighted on the training of *Model_3*. *Model_3* has better accuracy than *Model_1* on slices 1, 2, 7, 10, and comparable accuracy on slices 139 and 142. (B) Visualization of the worst data slices from the three models. Slices highlighted in red have *Model_3* performance worse than *Model_1*.

The car's on-board processor has limited compute power and cannot handle very complex models, such as neural networks. Therefore, the expert chose to train an XGBoost [57] model for this problem, which we will call *Model_1*. The expert split the data into three splits: training (60%), validation (30%) and testing (10%). They obtained an overall validation accuracy of 0.88, and were interested in evaluating the model using SliceTeller in order to find failure cases. Every sample in the dataset contained associated metadata about the environmental and sensor conditions. The metadata included 'Object Type', 'Distance', 'Sensor Approach', 'Scene Clutter', 'Direction', 'Speed', 'Temperature' and 'Weather'. In this section, we show an example of such analysis. Because this is a private dataset, the results are anonymized.

SliceTeller identified 145 data slices (using the DivExplorer algorithm with sup-

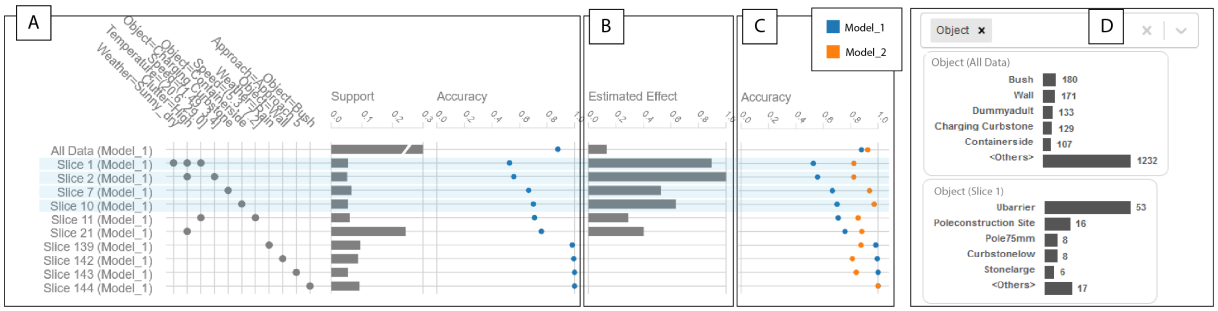


Figure 7.8: Analysis of the original model for object height classification (*Model_1*). (A) The summarized data slices for this model. The highlighted slices were selected by the expert to be improved in further model iterations. (B) SliceBoosting results showing the estimated effect of optimizing the model for the selected slices. (C) The comparison between the original model (*Model_1*) and the model optimized for the selected slices (*Model_2*). (D) The Slice Distribution View, containing the distribution of metadata 'Object' for Slices 'All Data' and 'Slice 1'.

port=0.05). As a first step, the expert used the Slice Summarization tool to reduce the number of data subsets to analyze. They set the Redundancy Threshold to 0.10 and obtained 11 distinct data slices for analysis. Sec. 7.8(A) shows the analysis of the data slices from *Model_1*. The expert noticed that Slices 1 (*sunny weather*, *high clutter* and *high temperature* (in the range (20.6, 29])), 2 (*high clutter* and *low speed*), 7 (*curbstones*) and 10 (*container sides*) performed significantly worse than the overall model.

Before investing into additional data acquisition for the characterized scenario, the expert wanted to check whether optimizing the model for these slices would be possible. Therefore, they used the SliceBoosting algorithm to estimate the effect of training the model with higher weights on these slices. The result of this run is shown in Sec. 7.8(B). Note that the performance of these slices is expected to improve, as well as the performance of slices 11 and 21.

They trained a new XGBoost model using the importance weighting method described in Section 7.3.6, this time adding higher sample weights to the samples belonging to Slices 1, 2, 7 and 10. *Model_2* (Sec. 7.8(C)) contains the result of this optimized model. The performance of *Model_2* is higher on the optimized slices, with an accuracy increase of more than 0.2 for every optimized slice (the *Agreement Score* of the real model with the estimate model is 0.83412.). This improvement was appreciated by the expert. However, they noticed that there was a trade-off with slices 139, 142 and 143. In particular, the expert noted that slices 139 (Weather=Rain) and 142 (Object=Wall)

are critical for the autonomous driving application, and therefore should not have a significant drop in performance.

To fix this issue, they trained a new model, this time weighting the samples by the previously optimized slices, as well as slices 139 and 142. Sec. 7.7(A) shows the results of this optimization. *Model_3* has better performance than *Model_1* on the data slices where the model performs the worst, and comparable results on the slices Rain and Wall. To conclude the analysis, the expert wanted to visualize the worst data slices from all 3 trained models (Sec. 7.7(B)). We see that overall, *Model_3* performs better than the other models on the their worst data slices. We highlighted the slices where *Model_3* performs worse, noting that this difference was not considered significant by the expert. Fig. 7.3 shows the performance of the three models on the data slices of interest for validation and test data.

Table 7.3: Accuracy of Object Height Classification Models (Val / Test)

Slice	Description	Model_1	Model_2	Model_3
0	All Data	0.88 / 0.74	0.92 / **0.84**	**0.95** / 0.84
1	Clutter=High, Weather=Sunny, Temp=(20.6, 29.0]	0.53 / 0.60	0.82 / **0.87**	**0.96** / 0.85
2	Clutter=High, Speed=(1.5, 3.4]	0.56 / 0.67	0.82 / **0.86**	**0.89** / 0.84
7	Object=Charging Curbstone	0.67 / 0.36	0.94 / 0.85	**0.91** / **0.87**
10	Object=Containerside	0.70 / 0.66	0.97 / 0.99	**1.00** / **1.00**
139	Weather=Rain	**0.98** / 0.83	0.88 / 0.88	0.95 / **0.90**
142	Object=Wall	**0.99** / **0.88**	0.81 / 0.80	0.93 / 0.83

7.4.3 Case 3: Image-Based Fire Detection

Image-based fire detection is an important problem in the industrial setting, having the potential to identify fires in its early stages, preventing accidents and losses. In order to address this problem, our partner MLOps product team trained a Convolutional Neural Network on image frames to predict the label 'Fire' / 'No Fire', obtaining a validation accuracy score of 0.94. Transparency with customers is paramount in this

business application. Therefore, the engineering team wanted to identify and convey the model limitations to their customers.

Each video had six interpretable metadata that could be used for exploration with SliceTeller. These metadata were: 'Location' (Outdoor, Indoor), 'Reflections or Shadows' (Yes, No), 'Motion' (Yes, No), 'Approaching Object' (Yes, No), 'Blinking Light' (Yes, No), and 'Smoke Density' (integer between 0 and 4). This metadata was assigned to every video frame, together with a new value: 'Normalized Frame' (real number in the range [0, 1]), describing the frame position in the video. After inserting this data into SliceTeller, the system initially discovered 115 data slices (using the DivExplorer algorithm with support=0.2). They used the slice summarization slider at multiple thresholds in order to inspect the results and select data slices for exploration. Fig. 7.9 shows the most interesting data slices found by them.

The experts first inspected the worst data slice in the model, which was defined by scenes with 'Blinking Lights' and 'no reflections or shadows' (Fig. 7.9(A)). Using the confusion matrix, they saw that this data slice contained solely videos without fire, but it had many false alarms (Accuracy of 0.61). According to them, it was known that blinking lights negatively influenced the classifier prediction. However, they did not expect this large effect, with an accuracy decrease of 0.33.

They also identified data slices with problems in the beginning and ending of the videos. The accuracy of the beginning frames was lower (Fig. 7.9(B)), with the worst slice having an accuracy of 0.86. The problems in the start of the videos were expected, as cameras were moved around at this time and, as a result, frames would get blurry. The problems at the end of the videos (Fig. 7.9(C)), however, were surprising. They investigated the video frames using the Slice Distribution View and formulated the hypothesis that at the end of the videos, the smoke was already dissipated and hard to see. This was a useful insight for the experts, who decided to ignore those frames because their goal was to stop fires when they started.

Another surprising data slice contained samples in the middle of the recording (Fig. 7.9(D)). This slice contained many false alarms (error rate of 0.14). However, the

expert mentioned that "it looks like we should be able to get this case right". Using the Slice Detail View, they attributed these mistakes to mislabeled data. Based on this insight, they decided to double check the sample labels. Furthermore, they were very interested in using SliceTeller to find more cases like this, noting that too many false alarms can annoy customers.

In order to mitigate the problems found in the data slices, their strategy consisted of increasing the training data size, using data collection and data augmentation. To improve particular data slices, they said they would collect more samples in the same conditions of the slices of interest. They would thoroughly inspect the new samples in order to ensure data quality. Another mitigation strategy mentioned is data augmentation. Currently, the MLOps team is in the process of testing different augmentation strategies, such as including frames with added noise and blur to their training data. They were very interested in comparing multiple model versions using our system.

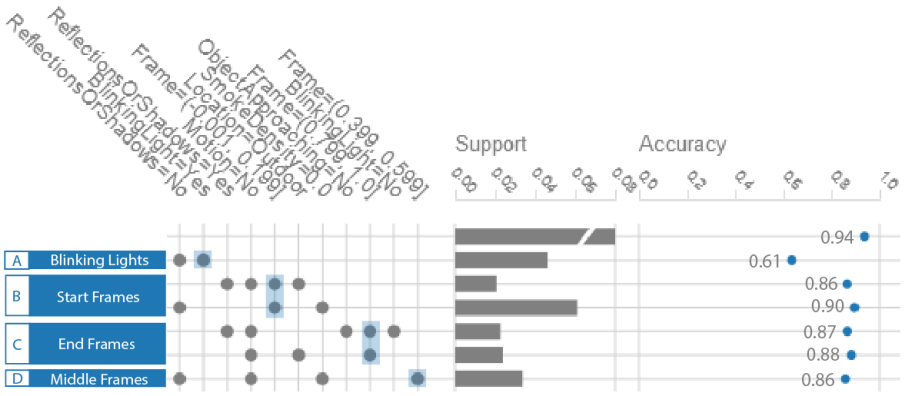


Figure 7.9: Image-Based Fire Detection Use Case: interesting data slices found by the MLOps engineers. (A) Blinking light. (B) Start of video. (C) End of video. (D) Middle of video.

7.5 Expert Interviews

SliceTeller was developed with continuous feedback from our product partners (7.3.1) over the course of six months. In this section, we discuss their analyses, insights and feedback for our system during a final round interview. First, we discuss the feedback from one expert working on the autonomous driving problem, who have used SliceTeller to evaluate and iterate over their object height classification models. Next,

we discuss the feedback from two experts working on the image-based fire detection problem, who derived actionable insights and identify data slices that require more training data using SliceTeller.

Expert User Demographics. We interviewed three ML experts (MLOps engineers) to evaluate SliceTeller. All experts have more than four years of experience in machine learning and have a graduate degree in STEM. They did not report any visual disabilities and were not color blind. The experts are not authors of this work.

7.5.1 Ultrasonic Object Height Classification Experts

In 7.4.2, we have shown an example of the analysis conducted together with one MLOps engineer from the product R&D team working on the ultrasonic object height classification problem. The expert used our system to evaluate their current model, as well as experiment on new models. The expert was interested in finding potential problems in their models. In particular, they wanted to identify misclassified samples close to the cars, as these mistakes can be critical. In their current workflow, they frequently evaluate their models on hand-curated data slices and fine tune them to achieve near-zero errors within a distance threshold. This workflow relies on many handcrafted scripts that have to be executed sequentially to evaluate and iterate over their models, making the process difficult to use. Therefore, they needed a simpler and more efficient alternative to this workflow.

During their analyses, they found a set of problematic slices that they wanted to optimize their model for. They did two model iterations, and were able to produce a solution that maintained a good trade-off across the multiple data slices of interest. They mentioned that the automatic data slicing and the slice matrix component were "very useful", because they could reduce the model testing time and effort significantly. However, they noted that they still needed to guarantee a customer-defined quality on the hand-crafted slices. Thus they were interested in combining SliceTeller in their current evaluation strategy, which would allow them to identify problems in their models that they did not think to look at before. As a feature request, they would like to propose problem-specific metrics for slice evaluation. We are currently working with

the ultrasonic team to learn about their metrics and include them in the next iteration.

7.5.2 Image-Based Fire Detection Experts

We interviewed two MLOps engineers working on the image-based fire detection problem. During their analyses, the experts were able to discover potential model issues, as well as formulate strategies to mitigate these issues. They used the Slice Matrix View and the Slice Summarization Slider the most frequently and expressed that these two views could facilitate the model exploration and help reduce the amount of time they needed to spend looking at data slices. They also liked to use the Slice Detail View: after identifying critical data slices, the experts used the Detail View to explore video frames and formulate hypothesis about the root cause.

Regarding model iteration, these experts did not show too much interest in fine-tuning models. Instead, they wanted to collect quality data to retrain and improve the existing models. They described their own experiments with data augmentation, such as noise and blur, with positive results. They mentioned the model comparison feature would be a powerful tool during model iteration.

They also had feature requests. They were interested in investigating the explanations for the model predictions. In particular, having importance maps displayed together with the images, so they could see where the model was "looking" when making a prediction. Furthermore, they wanted to be able to manually add data slices to SliceTeller, in order to keep track of critical slices for which 100% accuracy was required. These features will be implemented in the future.

SliceTeller provided our experts with new insights about their model, as well as validated hypothesis previously held by them. The experts mentioned that the system could be very useful, especially because they wanted to identify "interpretable, quantifiable boundaries for our system". These quality boundaries could be shared with customers, who can make an informed decision about their models.

7.6 Discussion

Other Mitigation Solutions for Model Improvements. As mentioned in Section 7.3.6, we focused on optimization-based model improvement approaches for SliceTeller without changing the training data. In practice, data-centric model improvement [249] is a promising direction to tackle real-world challenges. For example, after SliceTeller identifies particular weakness scenarios (e.g., snowy condition for autonomous driving) as indicated by the most critical slices, additional data can be collected corresponding to the scenarios. Recent work in data-centric AI focus on weakly-supervised learning [58, 288] and self-supervised learning [134, 243, 395] to reduce the annotation cost and facilitate fast model iteration. Another approach is to use image processing and deep learning-based image synthesis techniques to perform data augmentation. For example, adversarial objects can be placed on top of the images in order to improve the robustness of a model [150].

Limitations of SliceBoosting. The SliceBoosting algorithm in Section 7.3.5 was developed based on the assumption that, with a powerful optimization method, the model is able to make correct predictions on the selected slices in the validation data. The hardness of this task is determined by the generalization gap between training and validation slices. As shown in [304], strong regularization strategies in DRO training help to significantly close the gap and we utilized them in the implementation of *GroupDRO*. Robust optimization [202, 405] and domain generalization [406] are active research topics in the ML community. Although our empirical results in Section 7.4 demonstrated strong correlation between SliceBoosting estimation of the model improvement and the real improvement from importance weighting and *GroupDRO*, we plan to evaluate the effect with additional model optimization techniques as well as more public datasets.

Application Domains. In this chapter, I have shown how SliceTeller can be used for the analysis of classification models for image and tabular data. However, our system design is not specific to these domains, and can be applied to other data types, e.g., text data. The Slice Matrix abstracts the data features by using interpretable metadata to

describe the data slices. Therefore, it does not need to be adapted to other data types. The Slice Detail View, however, is dependent on the domain, and needs to be adapted to display a summary of the selected slices. For example, in the case of text domains, we foresee the use of text visualizations to convey the data slice's content (e.g., word clouds). Other data types would require custom visualization implementations for the inspection of data samples. An extension of SliceTeller is also possible for regression models: in this case, the slice finding algorithm needs to be adapted to have a measure of how correct a prediction is, and when it should be considered a defect.

Requirement on Metadata. One limitation of our system is the need for interpretable metadata to slice the models. This metadata usually requires intensive manual annotation, and is therefore expensive to create. Therefore, as future work, we would like to investigate automatic methods to generate such information. For example, for images, one possible research direction is to use self-supervised learning [134, 395] to automatically identify interpretable visual concepts.

7.7 Summary

In this chapter, I present SliceTeller, a novel VA system for data slice-driven validation of ML models. SliceTeller allows users to quickly identify problematic data slices, investigate the failure cases, understand the potential optimization trade-offs, and eventually iterate on new model solutions.

I demonstrated the power of this tool with three use cases that show how SliceTeller can be used to analyze, validate, and improve ML models in diverse application areas. SliceTeller was developed and improved in close collaboration with industry ML Ops engineers and domain experts working on product R&D. With the positive feedback, my industrial collaborators are currently working on incorporating SliceTeller into their ML development workflows to facilitate fast product iteration and model release.

Chapter 8
Conclusion

Language and text have served as the oldest and most instinctive mediums for preserving and disseminating knowledge for millennia. Their inherent capacity to encapsulate information and convey meaning, coupled with the exponential growth of textual data in our modern era, has significantly contributed to the emergence and advancement of large language models (LLMs) in recent times. My dissertation work offers a more reliable and transparent approach to harnessing the potential of LLMs. Specifically, I adopt the human-in-the-loop methodology and human-centered design principles in the development of intelligent visual analytics systems. Through my projects, I highlight the irreplaceable role of human intelligence and demonstrate how interactive visualization tools can enable greater human participation in the decision-making process of advanced AI models. By converting human interpretation of data into actionable insights, my research provides a safer and more trustworthy method for leveraging the knowledge and power of machine learning.

This dissertation presents several techniques that integrate interactive visual analytics (VA) with state-of-the-art machine learning, specifically natural language processing techniques, to facilitate knowledge exploration, presentation, and exploitation for domain experts. In Chapter 2, I demonstrate how these techniques can aid in knowledge exploration for manually or automatically generated technical logs of machine maintenance, enabling domain experts to participate in the analysis process for issue diagnosis and alleviation. After that, I introduce two knowledge presentation solutions for two distinct data types. In Chapter 3, I describe Text2Viz, a tool for generating infographics from natural language statements on numerical facts. In Chapter 4, I present ConceptScope, a tool for visualizing knowledge formalized as scientific documents based

on ontology. Finally, as the utilization and extension of insights are highly application-oriented, I showcase three visually-assisted knowledge exploitation projects driven by industry requirements. Chapter 5 presents ConceptEVA, a collaborative work with researchers from TUDelft and Databricks to exploit knowledge from domain ontology to support document summary customization for academic reading. Chapter 6 presents LabelVizier, a collaborative work with data analysts from the National Institute of Standard Technology (NIST) to facilitate the validation and relabeling of technical text annotations by creating computational notebooks for domain experts. Finally, Chapter 7 presents SliceTeller, my internship project with Bosch Research to utilize human knowledge for machine learning model validation and optimization in autonomous driving and intelligent fire detection.

My future plans involve integrating all tasks of the visual knowledge discovery loop (Chapter 1) to enable more advanced and real-world applications that serve for social good, particularly in the domains of education and smart manufacturing. I aspire for my work to offer solutions to the potential challenges arising from increasingly intelligent AI systems, and to pave the way for more human-centered research, regardless of how advanced machine learning techniques may become. Ultimately, I hope to spark philosophical and social discourse surrounding the roles of humans, machines, and robots within the ever-expanding field of artificial intelligence.

Appendix A
Appendix for Chapter 7

In this Appendix, I provide the supplementary materials for the SliceTeller paper. In Section A.1, we include a detailed evaluation of the SliceBoosting algorithm, using models and datasets from the first two use cases of the paper. In Section A.2 we provide experiment details for the three use cases, including model hyperparameters and a complete list of data slices.

A.1 SliceBoosting Evaluation

In order to evaluate the SliceBoosting algorithm, we have taken the original models from Case 1 (Bias Detection for AI Fairness) and Case 2 (Ultrasonic Object Height Classification for Autonomous Driving), computed the data slices for these models (Paper: Section 4.1 and 4.2), and validated the estimated effect of optimizing these models for their worst data slices.

For each use case, we selected the top five data slices and retrained five new models with an emphasis on each of these slices (i.e. each model is optimized for a single data slice). Next, we estimate the performance effect of optimizing a model for each of its worst five slices using SliceBoosting. Finally, we compute the Agreement Score between the estimated effects and the real performance effects of the optimization. The results for the Agreement Scores of the slices for Case 1 are presented in Table A.1 and the Agreement Scores of the slices for Case 2, in Table A.2.

Table A.1: *SliceBoosting* Evaluation - Case 1: Agreement Score of hair color classification models retrained on CelebA Dataset with emphasis on the top 5 worst data slices.

Model	Slice Desc	Agreement
Slice 1	Gray_Hair=Yes,Male=No	0.774557
Slice 2	Wearing_Necktie=No,Gray_Hair=Yes,Double_Chin=No	0.895770
Slice 3	Wearing_Necktie=No,Gray_Hair=Yes	0.890753
Slice 4	Gray_Hair=Yes,Double_Chin=No	0.860036
Slice 5	Gray_Hair=Yes	0.878247

Table A.2: *SliceBoosting* Evaluation - Case 2: Agreement Score of Models retrained on Ultrasonic Object Height Classification Dataset with emphasis on the top 5 worst data slices.

Slice	Slice Desc	Agreement
Slice 1	Clutter=High,Temperature=(20.6, 29.0],Weather=Sunny_dry	0.837487
Slice 2	Clutter=High,Speed=(1.49, 3.4]	0.733722
Slice 3	Approach=Approach 4,Clutter=High,Temperature=(20.6, 29.0]	0.727766
Slice 4	Approach=Approach 4,Clutter=High,Weather=Sunny_dry	0.832735
Slice 5	Approach=Approach 4,Distance=(399.999, 441.0],Clutter=High	0.748457

A.2 Use Cases

A.2.1 Case 1: Bias Detection for AI Fairness in Image Classification Models

For this study, a pretrained ResNet50 model (PyTorch implementation[1]) was fine tuned to classify "Gray Hair" on the CelebA dataset Images. The last fully connected layer was trained for 50 epochs using the Adam optimizer with learning rate $lr = 0.0001$, obtaining an overall accuracy of 0.98. A complete list of data slices found for the ResNet50 model and their validation accuracy are shown in Table A.3.

After the users selected Slices 1 and 2 to optimize the model for, a new model was trained using the same architecture (ResNet50), but with the *GroupDRO* training strategy (a loss function that focuses on a subset of the data, i.e. selected slices). The GroupDRO parameters are as follows: *Generalization Adjustment*=0.1, *L2 Regularization*=0.1,

[1] `https://pytorch.org/vision/0.13/models.html`

Learning Rate=0.0001, *Number Epochs*=25. The new model obtained an overall accuracy of 0.95, with improved performance on the optimized slices. The performances of the *GroupDRO* model on the validation data slices are shown in Table A.3.

Table A.3: Accuracy of Data Slices for Use Case 1 (AI Fairness)

Slice	Description	Support	Acc. ResNet50	Acc. GroupDRO
Slice 0	All Data	1.000000	0.980269	0.952836
Slice 1	Male=No, Gray_Hair=Yes	0.010470	0.649038	0.908654
Slice 2	Gray_Hair=Yes, Double_Chin=No, Wearing_Necktie=No	0.023406	0.651613	0.903226
Slice 3	Gray_Hair=Yes, Wearing_Necktie=No	0.030302	0.682724	0.915282
Slice 4	Gray_Hair=Yes, Double_Chin=No	0.033573	0.683658	0.911544
Slice 5	Gray_Hair=Yes	0.048674	0.719752	0.921406
Slice 6	Gray_Hair=Yes, Chubby=Yes	0.014849	0.755932	0.942373
Slice 7	Wearing_Necktie=Yes, Gray_Hair=Yes	0.018372	0.780822	0.931507
Slice 8	Double_Chin=Yes, Gray_Hair=Yes	0.015100	0.800000	0.943333
Slice 9	Wearing_Necktie=Yes, Double_Chin=Yes	0.016862	0.862687	0.826866
Slice 10	Wearing_Necktie=Yes, Chubby=Yes	0.017567	0.865330	0.836676
Slice 11	Young=No, Wearing_Necktie=Yes	0.046459	0.874323	0.834236
Slice 12	Young=No, Bald=Yes	0.015352	0.878689	0.685246
Slice 13	Double_Chin=Yes	0.049076	0.902564	0.838974
Slice 14	Bald=Yes	0.020688	0.909976	0.754258

Continued on next page

Table A.3: Accuracy of Data Slices for Use Case 1 (AI Fairness)

Slice	Description	Support	Acc. ResNet50	Acc. GroupDRO
Slice 15	Chubby=Yes	0.061207	0.914474	0.865954
Slice 16	Wearing_Necktie=Yes	0.072633	0.918919	0.893278
Slice 17	Young=No	0.253435	0.927110	0.857398
Slice 18	Young=No, Gray_Hair=No, Wearing_Necktie=Yes	0.028137	0.933810	0.771020
Slice 19	Gray_Hair=No, Double_Chin=Yes	0.033976	0.948148	0.792593
Slice 20	Pointy_Nose=Yes, Wearing_Necktie=Yes	0.013842	0.949091	0.901818
Slice 21	Gray_Hair=No, Bald=Yes	0.015151	0.960133	0.677741

A.2.2 Case 2: Ultrasonic Object Height Classification for Autonomous Driving

For this study, an XGBoost model trained by MLOps engineers was investigated. The model performed binary classification of object heights (High / Low) based on ultrassonic sensor features (tabular, 71 numerical features). Because the model is proprietary, we do not share its hyperparameters in this document.

The users inspected the data slices for this model (Model_1, shown in Table A.4) using SliceTeller, and trained two new models with the same hyperparameters, this time using higher sample weights on the selected data slices (as described in Section 3.6). Model_2 was optimized based on Slices 1, 2, 7 and 10. Model_3 was optimized based on Slices 1, 2, 7, 10, 139, 142. The validation accuracy of all models are shown in Table A.4.

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 0	All Data	1.000000	0.879098	0.925205	0.951332
Slice 1	Temperature=(20.6, 29.0], Weather=Sunny_dry, Clutter=High	0.055328	0.527778	0.824074	0.962963
Slice 2	Speed=(1.49, 3.4], Clutter=High	0.052254	0.558824	0.823529	0.892157
Slice 3	Temperature=(20.6, 29.0], Approach=Approach 4, Clutter=High	0.066598	0.607692	0.746154	0.861538
Slice 4	Weather=Sunny_dry, Approach=Approach 4, Clutter=High	0.117316	0.646288	0.877729	0.960699
Slice 5	Approach=Approach 4, Distance=(399.999, 441.0], Clutter=High	0.057377	0.651786	0.821429	0.946429
Slice 6	Weather=Sunny_dry, Distance=(399.999, 441.0], Clutter=High	0.060963	0.655462	0.848739	0.983193
Slice 7	Object=Charging Curbstone	0.066086	0.666667	0.937984	0.914729
Slice 8	Approach=Approach 4, Object=Containerside	0.050205	0.673469	0.969388	1.000000
Slice 9	Temperature=(20.6, 29.0], Clutter=High	0.090164	0.693182	0.806818	0.892045
Slice 10	Object=Containerside	0.054816	0.700935	0.971963	1.000000

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 11	Temperature=(20.6, 29.0], Speed=(5.3, 7.2]	0.059939	0.709402	0.854701	0.965812
Slice 12	Distance=(399.999, 441.0], Clutter=High	0.089139	0.712644	0.873563	0.965517
Slice 13	Weather=Sunny_dry, Clutter=High	0.157787	0.714286	0.883117	0.967532
Slice 14	Temperature=(12.2, 20.6], Weather=Sunny_dry, Speed=(1.49, 3.4]	0.054303	0.726415	0.933962	0.971698
Slice 15	Approach=Approach 4, Clutter=High	0.158299	0.728155	0.860841	0.919094
Slice 16	Weather=Sunny_dry, Speed=(1.49, 3.4], Approach=Approach 4	0.099385	0.742268	0.932990	0.963918
Slice 17	Clutter=Medium, Object=Charging Curbstone	0.052254	0.745098	0.960784	0.911765
Slice 18	Distance=(441.0, 471.0], Clutter=High	0.080943	0.746835	0.854430	0.905063
Slice 19	Temperature=(20.6, 29.0], Weather=Sunny_dry, Approach=Approach 4, Distance=(399.999, 441.0]	0.062500	0.754098	0.885246	0.967213
Slice 20	Temperature=(12.2, 20.6], Distance=(399.999, 441.0], Clutter=High	0.050717	0.757576	0.949495	0.989899

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 21	Clutter=High	0.242828	0.759494	0.881857	0.938819
Slice 22	Approach=Approach 1, Weather=Cloudy	0.088115	0.767442	0.953488	0.936047
Slice 23	Temperature=(12.2, 20.6], Weather=Sunny_dry, Approach=Approach 4, Clutter=High	0.070184	0.773723	0.934307	0.963504
Slice 24	Approach=Approach 4, Speed=(1.49, 3.4], Distance=(399.999, 441.0]	0.053279	0.778846	0.932692	0.971154
Slice 25	Temperature=(20.6, 29.0], Weather=Sunny_dry, Approach=Approach 4	0.174693	0.780059	0.923754	0.973607
Slice 26	Temperature=(20.6, 29.0], Weather=Sunny_dry, Distance=(399.999, 441.0]	0.086066	0.785714	0.904762	0.976190
Slice 27	Temperature=(20.6, 29.0], Approach=Approach 4, Distance=(399.999, 441.0]	0.074795	0.787671	0.876712	0.945205
Slice 28	Temperature=(12.2, 20.6], Approach=Approach 4, Speed=(1.49, 3.4]	0.094262	0.793478	0.929348	0.967391
Slice 29	Temperature=(12.2, 20.6], Clutter=High	0.138832	0.800738	0.948339	0.966790

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 30	Temperature=(20.6, 29.0], Weather=Sunny_dry, Distance=(471.0, 499.0], Approach=Approach 4	0.059426	0.801724	0.956897	0.974138
Slice 31	Approach=Approach 1	0.140369	0.802920	0.956204	0.941606
Slice 32	Temperature=(20.6, 29.0], Weather=Sunny_dry	0.245389	0.805846	0.933194	0.968685
Slice 33	Weather=Sunny_dry, Speed=(1.49, 3.4]	0.147541	0.809028	0.944444	0.972222
Slice 34	Approach=Approach 4, Speed=(1.49, 3.4]	0.165471	0.814241	0.922601	0.950464
Slice 35	Temperature=(12.2, 20.6], Approach=Approach 4, Clutter=High	0.091189	0.814607	0.943820	0.960674
Slice 36	Temperature=(12.2, 20.6], Weather=Sunny_dry, Distance=(441.0, 471.0], Approach=Approach 4	0.052766	0.815534	0.990291	0.990291
Slice 37	Temperature=(12.2, 20.6], Approach=Approach 1	0.111168	0.815668	0.963134	0.953917
Slice 38	Temperature=(20.6, 29.0], Clutter=Low	0.067623	0.818182	0.962121	0.977273
Slice 39	Weather=Sunny_dry, Speed=(5.3, 7.2]	0.119365	0.819742	0.905579	0.969957

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 40	Temperature=(20.6, 29.0], Weather=Sunny_dry, Distance=(441.0, 471.0]	0.071209	0.820144	0.942446	0.978417
Slice 41	Temperature=(20.6, 29.0], Approach=Approach 4	0.222336	0.820276	0.894009	0.940092
Slice 42	Speed=(1.49, 3.4], Distance=(399.999, 441.0]	0.078381	0.823529	0.862745	0.941176
Slice 43	Temperature=(20.6, 29.0], Weather=Sunny_dry, Speed=(1.49, 3.4]	0.067111	0.824427	0.954198	0.969466
Slice 44	Weather=Sunny_dry, Distance=(441.0, 471.0], Speed=(1.49, 3.4]	0.052766	0.825243	0.951456	0.970874
Slice 45	Weather=Sunny_dry, Distance=(441.0, 471.0], Approach=Approach 4	0.130123	0.826772	0.964567	0.988189
Slice 46	Approach=Approach 4, Distance=(441.0, 471.0], Speed=(1.49, 3.4]	0.059939	0.829060	0.940171	0.965812
Slice 47	Temperature=(20.6, 29.0], Distance=(399.999, 441.0]	0.111680	0.830275	0.903670	0.958716
Slice 48	Temperature=(12.2, 20.6], Weather=Sunny_dry, Clutter=High	0.093750	0.830601	0.950820	0.972678

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 49	Distance=(471.0, 499.0], Clutter=High	0.072746	0.830986	0.922535	0.943662
Slice 50	Distance=(471.0, 499.0], Approach=Approach 4, Speed=(1.49, 3.4]	0.052254	0.833333	0.892157	0.911765
Slice 51	Clutter=High, Weather=Cloudy	0.079918	0.833333	0.871795	0.878205
Slice 52	Speed=(5.3, 7.2], Distance=(399.999, 441.0]	0.077869	0.835526	0.907895	0.980263
Slice 53	Distance=(471.0, 499.0], Clutter=Low	0.053279	0.836538	0.875000	0.884615
Slice 54	Weather=Sunny_dry, Approach=Approach 4	0.404201	0.837769	0.949303	0.980989
Slice 55	Speed=(7.2, 9.1], Clutter=High	0.060451	0.838983	0.881356	0.923729
Slice 56	Weather=Sunny_dry, Distance=(399.999, 441.0]	0.183914	0.844011	0.930362	0.980501
Slice 57	Speed=(7.2, 9.1], Temperature=(12.2, 20.6], Weather=Cloudy	0.055840	0.844037	0.889908	0.899083
Slice 58	Clutter=Medium, Approach=Approach 1	0.062500	0.844262	0.967213	0.934426
Slice 59	Temperature=(20.6, 29.0]	0.323258	0.844691	0.912837	0.942948
Slice 60	Distance=(441.0, 471.0], Speed=(5.3, 7.2]	0.057377	0.848214	0.955357	0.973214

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 61	Clutter=Medium, Speed=(3.4, 5.3], Weather=Cloudy	0.054816	0.850467	0.943925	0.943925
Slice 62	Speed=(7.2, 9.1], Distance=(471.0, 499.0]	0.061988	0.851240	0.876033	0.925620
Slice 63	Clutter=Medium, Temperature=(12.2, 20.6], Weather=Cloudy	0.106557	0.855769	0.956731	0.937500
Slice 64	Temperature=(12.2, 20.6], Distance=(399.999, 441.0], Weather=Cloudy	0.085553	0.856287	0.976048	0.976048
Slice 65	Clutter=Very low, Temperature=(12.2, 20.6], Distance=(441.0, 471.0]	0.050205	0.857143	0.948980	0.979592
Slice 66	Speed=(3.4, 5.3], Clutter=High	0.064549	0.857143	0.936508	0.960317
Slice 67	Weather=Sunny_dry	0.563525	0.857273	0.939091	0.974545
Slice 68	Distance=(471.0, 499.0], Weather=Sunny_dry, Approach=Approach 4	0.134221	0.858779	0.958015	0.977099
Slice 69	Temperature=(12.2, 20.6], Weather=Sunny_dry, Approach=Approach 4	0.174693	0.859238	0.964809	0.985337
Slice 70	Speed=(5.3, 7.2]	0.204918	0.860000	0.930000	0.967500

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 71	Speed=(7.2, 9.1], Temperature=(20.6, 29.0]	0.078381	0.862745	0.921569	0.960784
Slice 72	Speed=(1.49, 3.4]	0.268443	0.864504	0.908397	0.950382
Slice 73	Approach=Approach 4	0.631660	0.868613	0.932685	0.954582
Slice 74	Temperature=(12.2, 20.6], Weather=Cloudy	0.243340	0.869474	0.957895	0.953684
Slice 75	Clutter=Very low, Approach=Approach 4, Speed=(1.49, 3.4]	0.059426	0.870690	0.965517	0.974138
Slice 76	Speed=(5.3, 7.2], Weather=Cloudy	0.055840	0.871560	0.954128	0.954128
Slice 77	Clutter=Medium, Weather=Cloudy	0.138320	0.877778	0.929630	0.929630
Slice 78	Clutter=Very low, Speed=(5.3, 7.2]	0.067111	0.877863	0.969466	0.977099
Slice 79	Speed=(3.4, 5.3], Weather=Cloudy	0.093238	0.879121	0.939560	0.923077
Slice 80	Speed=(3.4, 5.3], Temperature=(12.2, 20.6], Distance=(441.0, 471.0]	0.059426	0.879310	0.965517	0.974138
Slice 81	Clutter=Very low, Weather=Sunny_dry, Approach=Approach 4	0.102971	0.885572	0.965174	0.985075

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 82	Clutter=Very low, Weather=Sunny_dry, Distance=(441.0, 471.0]	0.058402	0.885965	0.991228	1.000000
Slice 83	Speed=(3.4, 5.3], Approach=Approach 1	0.050205	0.887755	0.989796	0.928571
Slice 84	Weather=Cloudy	0.323770	0.890823	0.927215	0.925633
Slice 85	Clutter=Low	0.158299	0.893204	0.915858	0.935275
Slice 86	Clutter=Medium, Temperature=(12.2, 20.6], Speed=(3.4, 5.3]	0.069160	0.896296	0.992593	0.962963
Slice 87	Distance=(471.0, 499.0], Speed=(5.3, 7.2]	0.069672	0.897059	0.933824	0.948529
Slice 88	Temperature=(12.2, 20.6], Approach=Approach 4, Distance=(399.999, 441.0]	0.114754	0.897321	0.968750	0.986607
Slice 89	Clutter=Very low, Distance=(441.0, 471.0]	0.090676	0.898305	0.966102	0.988701
Slice 90	Clutter=Very low, Weather=Sunny_dry	0.164959	0.900621	0.953416	0.972050
Slice 91	Clutter=Very low, Distance=(471.0, 499.0]	0.098873	0.901554	0.922280	0.943005
Slice 92	Clutter=Medium, Approach=Approach 4, Weather=Cloudy	0.062500	0.901639	0.885246	0.918033

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 93	Approach=Approach 4, Speed=(3.4, 5.3], Weather=Cloudy	0.052254	0.901961	0.892157	0.901961
Slice 94	Temperature=(20.6, 29.0], Approach=Approach 4, Speed=(3.4, 5.3]	0.063012	0.902439	0.926829	0.943089
Slice 95	Clutter=Very low, Speed=(1.49, 3.4]	0.080430	0.904459	0.968153	0.980892
Slice 96	Clutter=Very low, Approach=Approach 4	0.196209	0.906005	0.958225	0.971279
Slice 97	Speed=(3.4, 5.3], Distance=(441.0, 471.0]	0.098873	0.906736	0.922280	0.958549
Slice 98	Clutter=Very low, Temperature=(20.6, 29.0]	0.067111	0.908397	0.946565	0.931298
Slice 99	Speed=(7.2, 9.1], Distance=(399.999, 441.0]	0.067111	0.908397	0.938931	0.931298
Slice 100	Clutter=Medium, Distance=(399.999, 441.0]	0.100922	0.908629	0.918782	0.934010
Slice 101	Temperature=(12.2, 20.6], Weather=Sunny_dry, Distance=(471.0, 499.0]	0.090164	0.909091	0.937500	0.977273
Slice 102	Clutter=Medium, Temperature=(12.2, 20.6]	0.201332	0.910941	0.944020	0.951654

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 103	Clutter=Medium, Distance=(441.0, 471.0], Weather=Cloudy	0.054816	0.915888	0.943925	0.962617
Slice 104	Clutter=Medium, Speed=(5.3, 7.2]	0.061475	0.916667	0.966667	0.966667
Slice 105	Speed=(3.4, 5.3]	0.290984	0.917254	0.952465	0.957746
Slice 106	Temperature=(12.2, 20.6], Weather=Sunny_dry, Speed=(3.4, 5.3]	0.081455	0.918239	0.974843	0.987421
Slice 107	Clutter=Very low	0.284836	0.919065	0.951439	0.967626
Slice 108	Approach=Approach 4, Distance=(441.0, 471.0], Weather=Cloudy	0.065061	0.921260	0.858268	0.897638
Slice 109	Distance=(441.0, 471.0], Clutter=Low	0.055840	0.926606	0.926606	0.954128
Slice 110	Temperature=(12.2, 20.6], Speed=(5.3, 7.2]	0.118852	0.926724	0.978448	0.978448
Slice 111	Clutter=Medium	0.314037	0.928222	0.939641	0.954323
Slice 112	Weather=Sunny_dry, Speed=(3.4, 5.3]	0.165984	0.929012	0.969136	0.981481
Slice 113	Temperature=(3.8, 12.2]	0.095287	0.930108	0.887097	0.940860
Slice 114	Clutter=Medium, Approach=Approach 4, Distance=(399.999, 441.0]	0.066086	0.930233	0.976744	0.976744
Slice 115	Object=Pole25mm	0.052766	0.932039	0.961165	0.980583

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 116	Approach=Approach 4, Weather=Cloudy	0.190061	0.932615	0.902965	0.911051
Slice 117	Clutter=Medium, Distance=(471.0, 499.0], Approach=Approach 4	0.069672	0.933824	0.955882	0.948529
Slice 118	Clutter=Medium, Temperature=(12.2, 20.6], Distance=(441.0, 471.0]	0.072234	0.936170	0.985816	0.978723
Slice 119	Clutter=Very low, Weather=Cloudy	0.073770	0.937500	0.965278	0.965278
Slice 120	Temperature=(12.2, 20.6], Clutter=Low	0.065574	0.937500	0.875000	0.898438
Slice 121	Speed=(1.49, 3.4], Weather=Cloudy	0.078381	0.941176	0.934641	0.954248
Slice 122	Temperature=(3.8, 12.2], Approach=Approach 4	0.063525	0.943548	0.943548	0.943548
Slice 123	Clutter=Medium, Approach=Approach 4	0.202357	0.944304	0.959494	0.972152
Slice 124	Clutter=Medium, Temperature=(12.2, 20.6], Speed=(1.49, 3.4]	0.065574	0.945312	0.875000	0.953125
Slice 125	Clutter=Medium, Distance=(471.0, 499.0], Weather=Sunny_dry	0.052766	0.951456	0.990291	1.000000

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 126	Clutter=Medium, Distance=(441.0, 471.0]	0.109119	0.953052	0.957746	0.976526
Slice 127	Approach=Approach 2	0.066598	0.953846	0.976923	0.976923
Slice 128	Clutter=Very low, Distance=(399.999, 441.0]	0.095287	0.956989	0.967742	0.973118
Slice 129	Clutter=Medium, Temperature=(20.6, 29.0]	0.098361	0.958333	0.953125	0.973958
Slice 130	Approach=Approach 4, Speed=(1.49, 3.4], Weather=Cloudy	0.051742	0.960396	0.920792	0.940594
Slice 131	Temperature=(12.2, 20.6], Approach=Approach 4, Speed=(5.3, 7.2]	0.078893	0.961039	0.987013	0.993506
Slice 132	Clutter=Medium, Speed=(1.49, 3.4]	0.093750	0.961749	0.890710	0.950820
Slice 133	Object=Dummyadult	0.068135	0.962406	0.992481	1.000000
Slice 134	Temperature=(20.6, 29.0], Weather=Cloudy	0.071721	0.964286	0.850000	0.864286
Slice 135	Clutter=Medium, Approach=Approach 4, Distance=(441.0, 471.0]	0.066598	0.969231	0.946154	0.992308
Slice 136	Temperature=(12.2, 20.6], Weather=Sunny_dry, Speed=(5.3, 7.2]	0.052766	0.970874	0.990291	0.990291

Continued on next page

Table A.4: Accuracy of Data Slices for Use Case 2 (Ultrasonic Object Height Classification)

Slice	Description	Support	Acc. Model_1	Acc. Model_2	Acc. Model_3
Slice 137	Clutter=Medium, Weather=Sunny_dry	0.142930	0.971326	0.989247	0.996416
Slice 138	Clutter=Very low, Temperature=(12.2, 20.6], Distance=(399.999, 441.0]	0.055328	0.981481	0.990741	1.000000
Slice 139	Weather=Rain	0.094775	0.983784	0.875676	0.951351
Slice 140	Clutter=Medium, Temperature=(12.2, 20.6], Weather=Sunny_dry	0.068648	0.985075	1.000000	1.000000
Slice 141	Clutter=Very low, Speed=(3.4, 5.3]	0.085041	0.987952	0.981928	0.969880
Slice 142	Object=Wall	0.087602	0.994152	0.812865	0.929825
Slice 143	Approach=Approach 5	0.054816	1.000000	0.841121	0.943925
Slice 144	Object=Bush	0.092213	1.000000	1.000000	1.000000

A.2.3 Case 3: Image-Based Fire Detection

In this case study, MLOps engineers investigated a Convolutional Neural Network model trained for classifying the presence of fire in images. Because the model is proprietary, we do not share its hyperparameters in this document. The experts used SliceTeller to investigate their model in detail, and found data and model problems based on their inspection (Section 4.3). A complete list of data slices for the fire detection model is shown in Table A.5.

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 0	All Data	1.000000	0.935606
Slice 1	BlinkingLight=Yes, ReflectionsOrShadows=No	0.047222	0.614973
Slice 2	Motion=No, Frame=(0.199, 0.399], Location=Indoo...	0.020707	0.670732
Slice 3	BlinkingLight=Yes, Frame=(0.199, 0.399]	0.020960	0.674699
Slice 4	Location=Indoor, Motion=No, BlinkingLight=Yes	0.072980	0.737024
Slice 5	ObjectApproaching=No, Motion=No, BlinkingLight=Yes	0.072980	0.737024
Slice 6	Motion=No, ObjectApproaching=No, Frame=(0.199, ...	0.026515	0.742857
Slice 7	ObjectApproaching=No, BlinkingLight=Yes	0.076010	0.747508
Slice 8	Motion=No, BlinkingLight=Yes	0.080556	0.761755
Slice 9	Motion=No, Frame=(0.199, 0.399], Location=Indoo...	0.030808	0.770492
Slice 10	Location=Indoor, BlinkingLight=Yes	0.084596	0.773134
Slice 11	Motion=No, Location=Indoor, SmokeDensity=0.0, R...	0.089899	0.786517
Slice 12	BlinkingLight=Yes	0.092172	0.791781
Slice 13	SmokeDensity=0.0, Motion=No, Frame=(0.199, 0.39...	0.036616	0.806897
Slice 14	Motion=No, Frame=(0.199, 0.399], Location=Indoo...	0.038636	0.810458

Continued on next page

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 15	Motion=No, Frame=(-0.001, 0.199], Location=Indo...	0.029798	0.813559
Slice 16	Motion=No, ObjectApproaching=No, Frame=(-0.001,...	0.026263	0.817308
Slice 17	Location=Indoor, Frame=(0.599, 0.799], Blinking...	0.020960	0.819277
Slice 18	Frame=(0.599, 0.799], BlinkingLight=Yes	0.022980	0.835165
Slice 19	Frame=(0.199, 0.399], Location=Indoor, SmokeDen...	0.046465	0.836957
Slice 20	Location=Indoor, SmokeDensity=0.0, Motion=No, R...	0.142929	0.839223
Slice 21	SmokeDensity=0.0, Motion=No, ObjectApproaching=...	0.120202	0.840336
Slice 22	SmokeDensity=3.0, Frame=(0.799, 1.0]	0.023990	0.842105
Slice 23	SmokeDensity=0.0, Motion=No, Frame=(-0.001, 0.1...	0.036111	0.846154
Slice 24	Frame=(-0.001, 0.199], Location=Indoor, SmokeDe...	0.040404	0.850000
Slice 25	Motion=No, Frame=(-0.001, 0.199], Location=Indo...	0.034091	0.851852
Slice 26	SmokeDensity=0.0, ObjectApproaching=No, Frame=(...	0.052273	0.855072
Slice 27	Location=Indoor, SmokeDensity=0.0, Motion=No, F...	0.054040	0.859813

Continued on next page

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 28	Location=Indoor, SmokeDensity=0.0, Frame=(0.199...	0.056566	0.861607
Slice 29	Motion=No, Frame=(0.599, 0.799], ObjectApproach...	0.023737	0.861702
Slice 30	SmokeDensity=0.0, Motion=No, Frame=(0.199, 0.39...	0.055051	0.862385
Slice 31	Location=Indoor, SmokeDensity=0.0, Frame=(-0.00...	0.050253	0.864322
Slice 32	Location=Indoor, SmokeDensity=0.0, Motion=No, F...	0.048990	0.865979
Slice 33	SmokeDensity=0.0, Motion=No, ReflectionsOrShado...	0.173232	0.867347
Slice 34	Motion=No, Frame=(0.599, 0.799], Location=Indoo...	0.026768	0.867925
Slice 35	Motion=No, Frame=(0.399, 0.599], Location=Indoo...	0.027525	0.871560
Slice 36	SmokeDensity=0.0, Frame=(0.199, 0.399], Reflect...	0.062374	0.874494
Slice 37	Location=Indoor, SmokeDensity=0.0, Motion=No, O...	0.176768	0.875714
Slice 38	ObjectApproaching=No, Motion=No, Frame=(0.199, ...	0.078030	0.877023
Slice 39	Location=Indoor, SmokeDensity=0.0, ObjectApproa...	0.208081	0.877427

Continued on next page

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 40	ObjectApproaching=No, Motion=No, Frame=(0.799, ...	0.024747	0.877551
Slice 41	SmokeDensity=0.0, Frame=(-0.001, 0.199], Reflec...	0.056818	0.880000
Slice 42	Location=Indoor, SmokeDensity=0.0, Frame=(0.199...	0.090404	0.882682
Slice 43	Motion=No, Frame=(0.799, 1.0], Location=Indoor,...	0.028030	0.882883
Slice 44	ObjectApproaching=No, Location=Outdoor, Motion=...	0.048232	0.884817
Slice 45	Location=Indoor, SmokeDensity=0.0, ReflectionsO...	0.261111	0.887814
Slice 46	Location=Indoor, ObjectApproaching=No, Motion=N...	0.090152	0.887955
Slice 47	Motion=No, Frame=(0.199, 0.399], ReflectionsOrS...	0.088131	0.888252
Slice 48	SmokeDensity=0.0, Frame=(0.599, 0.799], Motion=...	0.032576	0.891473
Slice 49	SmokeDensity=0.0, Motion=No, Frame=(-0.001, 0.199]	0.074747	0.891892
Slice 50	SmokeDensity=0.0, Motion=No, ObjectApproaching=No	0.255303	0.892186
Slice 51	Location=Indoor, SmokeDensity=0.0, Motion=No	0.260101	0.893204
Slice 52	SmokeDensity=3.0, Frame=(-0.001, 0.199]	0.024242	0.895833

Continued on next page

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 53	Frame=(0.399, 0.599], SmokeDensity=0.0, Motion=...	0.034091	0.896296
Slice 54	Frame=(0.599, 0.799], Location=Indoor, SmokeDen...	0.044192	0.897143
Slice 55	Motion=No, Frame=(0.599, 0.799], Location=Indoo...	0.037121	0.897959
Slice 56	SmokeDensity=0.0, ReflectionsOrShadows=No	0.291667	0.899567
Slice 57	Location=Indoor, SmokeDensity=0.0, Frame=(-0.00...	0.098232	0.899743
Slice 58	SmokeDensity=0.0, Frame=(0.199, 0.399], ObjectA...	0.108838	0.900232
Slice 59	ObjectApproaching=No, Location=Outdoor, Reflect...	0.055808	0.900452
Slice 60	Location=Indoor, SmokeDensity=0.0, Frame=(0.199...	0.109848	0.901149
Slice 61	Location=Indoor, Motion=No, Frame=(0.199, 0.399]	0.105556	0.901914
Slice 62	ObjectApproaching=No, Motion=No, Frame=(0.199, ...	0.106566	0.902844
Slice 63	Frame=(0.199, 0.399], ReflectionsOrShadows=No	0.113889	0.906874
Slice 64	Location=Indoor, SmokeDensity=0.0, Frame=(0.399...	0.052020	0.907767

Continued on next page

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 65	Frame=(0.799, 1.0], ReflectionsOrShadows=No	0.102525	0.908867
Slice 66	ObjectApproaching=No, Frame=(0.799, 1.0]	0.141667	0.909091
Slice 67	Motion=No, Frame=(0.799, 1.0]	0.122980	0.909651
Slice 68	Motion=Yes, ObjectApproaching=No, Frame=(0.199,...	0.028030	0.909910
Slice 69	SmokeDensity=0.0, Frame=(0.599, 0.799], Motion=...	0.053283	0.909953
Slice 70	Frame=(-0.001, 0.199], ReflectionsOrShadows=No	0.113636	0.911111
Slice 71	Motion=No, Frame=(0.399, 0.599], ReflectionsOrS...	0.022980	0.912088
Slice 72	SmokeDensity=0.0, Motion=No	0.391414	0.912903
Slice 73	Motion=No, Frame=(-0.001, 0.199]	0.131566	0.913628
Slice 74	Location=Indoor, SmokeDensity=0.0, Frame=(0.599...	0.053030	0.914286
Slice 75	ObjectApproaching=No, Frame=(0.199, 0.399]	0.160354	0.914961
Slice 76	Frame=(0.399, 0.599], SmokeDensity=0.0, Motion=...	0.050505	0.915000
Slice 77	SmokeDensity=0.0, ObjectApproaching=No	0.505556	0.915085
Slice 78	Location=Indoor, Frame=(0.199, 0.399]	0.161364	0.915493

Continued on next page

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 79	Location=Indoor, Motion=Yes, Frame=(0.199, 0.39...	0.030051	0.915966
Slice 80	Location=Indoor, SmokeDensity=0.0	0.520202	0.916505
Slice 81	SmokeDensity=2.0, Frame=(0.199, 0.399]	0.024495	0.917526
Slice 82	Frame=(0.399, 0.599], SmokeDensity=0.0, Reflect...	0.058586	0.918103
Slice 83	SmokeDensity=0.0, Frame=(0.599, 0.799], Reflect...	0.058838	0.918455
Slice 84	ReflectionsOrShadows=No	0.551768	0.921739
Slice 85	Frame=(0.799, 1.0]	0.190909	0.921958
Slice 86	BlinkingLight=No, Motion=No, Frame=(-0.001, 0.1...	0.032576	0.922481
Slice 87	Location=Indoor, Frame=(0.799, 1.0], Motion=Yes	0.049495	0.923469
Slice 88	ObjectApproaching=No, Frame=(0.799, 1.0], Motio...	0.047222	0.925134
Slice 89	Frame=(-0.001, 0.199]	0.199242	0.925222
Slice 90	Frame=(0.399, 0.599], SmokeDensity=0.0, Motion=No	0.077778	0.925325
Slice 91	Location=Indoor, ObjectApproaching=Yes, Frame=(...	0.021212	0.928571
Slice 92	Location=Indoor, ObjectApproaching=No, Frame=(0...	0.043939	0.931034
Slice 93	Frame=(0.799, 1.0], ReflectionsOrShadows=Yes	0.088384	0.937143

Continued on next page

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 94	Location=Outdoor, Motion=No, Reflection-sOrShado...	0.101010	0.937500
Slice 95	ObjectApproaching=Yes, Motion=No, Frame=(0.799,...	0.028535	0.938053
Slice 96	ObjectApproaching=No, Frame=(0.799, 1.0], Refle...	0.036869	0.938356
Slice 97	ObjectApproaching=No, Motion=Yes	0.250253	0.938446
Slice 98	Location=Indoor, Motion=Yes	0.260101	0.939806
Slice 99	Location=Indoor, ObjectApproaching=Yes	0.101010	0.940000
Slice 100	Location=Outdoor, Motion=No, Frame=(-0.001, 0.199]	0.025758	0.941176
Slice 101	ObjectApproaching=Yes, Frame=(-0.001, 0.199]	0.045455	0.944444
Slice 102	BlinkingLight=No	0.907828	0.950209
Slice 103	ObjectApproaching=Yes, Motion=No	0.136111	0.951763
Slice 104	Motion=No, Frame=(0.799, 1.0], Location=Indoor,...	0.020960	0.951807
Slice 105	Location=Outdoor, Motion=No	0.131313	0.951923
Slice 106	ReflectionsOrShadows=Yes	0.448232	0.952676
Slice 107	Motion=Yes	0.348485	0.952899
Slice 108	Location=Outdoor, Frame=(0.799, 1.0]	0.043434	0.953488
Slice 109	Frame=(0.399, 0.599]	0.198990	0.954315
Slice 110	Frame=(0.599, 0.799], SmokeDensity=2.0	0.023232	0.956522
Slice 111	ObjectApproaching=No, Motion=No, Frame=(-0.001,...	0.023232	0.956522

Continued on next page

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 112	BlinkingLight=No, Frame=(0.199, 0.399]	0.185101	0.961801
Slice 113	Frame=(0.199, 0.399], ReflectionsOrShadows=Yes	0.092172	0.964384
Slice 114	ObjectApproaching=No, BlinkingLight=Yes, Reflec...	0.028788	0.964912
Slice 115	ObjectApproaching=Yes, Motion=No, ReflectionsOr...	0.083081	0.966565
Slice 116	SmokeDensity=2.0	0.121717	0.966805
Slice 117	ObjectApproaching=Yes	0.234343	0.967672
Slice 118	Frame=(0.399, 0.599], Motion=Yes	0.070455	0.967742
Slice 119	Location=Outdoor	0.219697	0.967816
Slice 120	Motion=Yes, Frame=(0.199, 0.399], ReflectionsOr...	0.025758	0.970588
Slice 121	Motion=No, Location=Indoor, SmokeDensity=0.0, O...	0.103788	0.973236
Slice 122	ObjectApproaching=No, Motion=No, Frame=(0.199, ...	0.028535	0.973451
Slice 123	BlinkingLight=Yes, ReflectionsOrShadows=Yes	0.044949	0.977528
Slice 124	Frame=(0.399, 0.599], Motion=Yes, ReflectionsOr...	0.045960	0.978022
Slice 125	Motion=No, Frame=(0.399, 0.599], Location=Indoo...	0.071717	0.978873
Slice 126	ObjectApproaching=Yes, ReflectionsOrShadows=Yes	0.181313	0.979109

Continued on next page

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 127	SmokeDensity=0.0, BlinkingLight=No, Frame=(0.19...	0.049242	0.979487
Slice 128	SmokeDensity=3.0, Frame=(0.199, 0.399]	0.025000	0.979798
Slice 129	Motion=No, Frame=(0.199, 0.399], ReflectionsOrS...	0.045960	0.983516
Slice 130	SmokeDensity=0.0, BlinkingLight=No, Motion=No, ...	0.063889	0.984190
Slice 131	Motion=No, ObjectApproaching=No, ReflectionsOrS...	0.072980	0.986159
Slice 132	Motion=No, Frame=(0.599, 0.799], ReflectionsOrS...	0.021970	0.988506
Slice 133	ObjectApproaching=Yes, Location=Outdoor	0.133333	0.988636
Slice 134	ObjectApproaching=Yes, Motion=Yes	0.098232	0.989717
Slice 135	SmokeDensity=3.0, Frame=(0.599, 0.799]	0.026768	0.990566
Slice 136	Location=Outdoor, Motion=Yes	0.088384	0.991429
Slice 137	SmokeDensity=2.0, Frame=(-0.001, 0.199]	0.030051	0.991597
Slice 138	Frame=(0.399, 0.599], ObjectApproaching=Yes, Re...	0.033081	0.992366
Slice 139	Location=Outdoor, Frame=(0.199, 0.399]	0.044697	0.994350
Slice 140	ObjectApproaching=Yes, Frame=(0.199, 0.399]	0.045707	0.994475
Slice 141	SmokeDensity=2.0, Frame=(0.799, 1.0]	0.021212	1.000000
Slice 142	Frame=(0.399, 0.599], Location=Outdoor, ObjectA...	0.024495	1.000000

Continued on next page

Table A.5: Accuracy of Data Slices for Use Case 3 (Image-Based Fire Detection)

Slice	Description	Support	Acc. Model
Slice 143	Frame=(0.399, 0.599], SmokeDensity=3.0	0.026263	1.000000
Slice 144	ObjectApproaching=Yes, Location=Outdoor, Frame=...	0.026515	1.000000
Slice 145	Location=Outdoor, ReflectionsOrShadows=No	0.030556	1.000000

References

[1] ABZcreativePartners. The power of infographics. https://visual.ly/community /infographic /business/power-infographics-infographic, 2014.

[2] N. Achich, B. Bouaziz, A. Algergawy, and F. Gargouri. Ontology visualization: An overview. In *International Conference on Intelligent Systems Design and Applications*, pp. 880–891, 2017.

[3] S. Ackerman, O. Raz, and M. Zalmanovici. FreaAI: Automated extraction of data slices to test machine learning models. In *International Workshop on Engineering Dependable and Secure Machine Learning Systems*, pp. 67–83. Springer, 2020.

[4] Adobe. Adobe color cc. https://color.adobe.com.

[5] M. Agarwal, A. Srinivasan, and J. Stasko. Viswall: Visual data exploration using direct combination on large touch displays. In *2019 IEEE Visualization Conference (VIS)*, pp. 26–30. IEEE, 2019.

[6] E. Alexander, J. Kohlmann, R. Valenza, M. Witmore, and M. Gleicher. Serendip: Topic model-driven visual exploration of text corpora. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pp. 173–182, 2014.

[7] P. A. Alexander. Domain knowledge: Evolving themes and emerging concerns. *Educational psychologist*, 27(1):33–51, 1992.

[8] B. Alsallakh and L. Ren. Powerset: A comprehensive visualization of set intersections. *IEEE Transactions on visualization and computer graphics*, 23(1):361–370, 2016.

[9] S. Alspaugh, N. Zokaei, A. Liu, C. Jin, and M. A. Hearst. Futzing and moseying: Interviews with professional data analysts on exploration practices. *IEEE transactions on visualization and computer graphics*, 25(1):22–31, 2018.

[10] F. Amini, N. H. Riche, B. Lee, A. Monroy-Hernandez, and P. Irani. Authoring data-driven videos with dataclips. *IEEE transactions on visualization and computer graphics*, 23(1):501–510, 2017.

[11] P. Angelelli, S. Oeltze, J. Haász, C. Turkay, E. Hodneland, A. Lundervold, A. J. Lundervold, B. Preim, and H. Hauser. Interactive visual analysis of heterogeneous cohort-study data. *IEEE computer graphics and applications*, 34(5):70–82, 2014.

[12] D. Archambault and H. C. Purchase. Can animation support the visualisation of dynamic graphs? *Information Sciences*, 330:495–509, 2016.

[13] A. Arleo, C. Tsigkanos, C. Jia, R. A. Leite, I. Murturi, M. Klaffenböck, S. Dustdar, M. Wimmer, S. Miksch, and J. Sorger. Sabrina: Modeling and visualization of financial data over time with incremental domain knowledge. In *2019 IEEE Visualization Conference (VIS)*, pp. 51–55. IEEE, 2019.

[14] M. Ascari. The dangers of distant reading: Reassessing moretti's approach to literary genres. *Genre: Forms of Discourse and Culture*, 47(1):1–19, 2014. doi: 10.1215/00166928-2392348

[15] Z. Ashktorab, B. Hoover, M. Agarwal, C. Dugan, W. Geyer, H. B. Yang, and M. Yurochkin. Fairness evaluation in text classification: Machine learning practitioner perspectives of individual and group fairness. *arXiv preprint arXiv:2303.00673*, 2023.

[16] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pp. 722–735. Springer, Berlin, Heidelberg, 2007.

[17] B. Bach, Z. Wang, M. Farinella, D. Murray-Rust, and N. Henry Riche. Design patterns for data comics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 38. ACM, 2018.

[18] J. Bae, T. Helldin, M. Riveiro, S. Nowaczyk, M.-R. Bouguelia, and G. Falkman. Interactive clustering: A comprehensive review. *ACM Computing Surveys*, 53(1):1–39, 2020.

[19] G. Barash, E. Farchi, I. Jayaraman, O. Raz, R. Tzoref-Brill, and M. Zalmanovici. Bridging the gap between ML solutions and their business requirements using feature interactions. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 1048–1058, 2019.

[20] T. Barlow and P. Neville. A comparison of 2-d visualizations of hierarchies. In *IEEE Symposium on Information Visualization*, pp. 131–138, 2001.

[21] R. Baskerville and A. Dulipovici. The theoretical foundations of knowledge management. *Knowledge Management Research & Practice*, 4(2):83–105, 2006.

[22] A. Bäuerle, H. Neumann, and T. Ropinski. Classifier-guided visual correction of noisy labels for image classification tasks. In *Computer Graphics Forum*, vol. 39, pp. 195–205. Wiley misc Library, 2020.

[23] B. B. Bederson and A. Boltman. Does animation help users build mental maps of spatial information? In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*, pp. 28–35. IEEE, 1999.

[24] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020.

[25] M. Berger, K. McDonough, and L. M. Seversky. cite2vec: Citation-driven document exploration via word embeddings. *IEEE transactions on visualization and computer graphics*, 23(1):691–700, 2016. doi: 10.1109/TVCG.2016.2598667

[26] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009.

[27] T. Blascheck, L. M. Vermeulen, J. Vermeulen, C. Perin, W. Willett, T. Ertl, and S. Carpendale. Exploration strategies for discovery of interactivity in visualizations. *IEEE transactions on visualization and computer graphics*, 25(2):1407–1420, 2018.

[28] M. Blumenschein, M. Behrisch, S. Schmid, S. Butscher, D. R. Wahl, K. Villinger, B. Renner, H. Reiterer, and D. A. Keim. Smartexplore: Simplifying high-dimensional data analysis through a table-based visual analytics approach. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 36–47. IEEE, 2018.

[29] A. Bock, E. Axelsson, C. Emmart, M. Kuznetsova, C. Hansen, and A. Ynnerman. Openspace: Changing the narrative of public dissemination in astronomical visualization from what to how. *IEEE computer graphics and applications*, 38(3):44–57, 2018.

[30] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pp. 1247–1250. ACM, New York, NY, 2008. doi: 10.1145/1376616.1376746

[31] R. Bommasani and C. Cardie. Intrinsic evaluation of summarization datasets. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing*, pp. 8075–8096. Association for Computational Linguistics, misc, 2020. doi: 10.18653/v1/2020.emnlp-main.649

[32] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark, D. De Las Casas, A. Guy, J. Menick, R. Ring, T. Hennigan, S. Huang, L. Maggiore, C. Jones, A. Cassirer, A. Brock, M. Paganini, G. Irving, O. Vinyals, S. Osindero, K. Simonyan, J. Rae, E. Elsen, and L. Sifre. Improving language models by retrieving from trillions of tokens. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds., *Proceedings of the International Conference on Machine Learning*, vol. 162 of *Proceedings of Machine Learning Research*, pp. 2206–2240. PMLR, Baltimore, Maryland, 17–23 Jul 2022.

[33] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2306–2315, 2013.

[34] M. Bostock, V. Ogievetsky, and J. Heer. D^3: Data-driven documents. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2301–2309, 2011. doi: 10.1109/tvcg.2011.185

[35] N. Bostrom. What happens when our computer gets smarter than we are? https://www.ted.com/talks/nick_bostrom_what_happens_when_our_computers_get_smarter_than_we_are?language=en, 2015. Accessed December 3, 2019.

[36] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642. Association for Computational Linguistics, Lisbon, Portugal, 2015.

[37] D. Braun, A. H. Mendez, F. Matthes, and M. Langen. Evaluating natural language understanding services for conversational question answering systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 174–185, 2017.

[38] M. Brehmer, J. Ng, K. Tate, and T. Munzner. Matches, mismatches, and methods: Multiple-view workflows for energy portfolio analysis. *IEEE transactions on visualization and computer graphics*, 22(1):449–458, 2015.

[39] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner. Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pp. 1–8, 2014.

[40] BrittSE. Get up and get moving! the health benefits of exercise. https://visual.ly/community/infographic/health/get-and-get-moving-health-benefits-exercise, 2013.

[41] B. Broeksema, A. C. Telea, and T. Baudel. Visual analysis of multi-dimensional categorical data sets. *Computer Graphics Forum*, 32(8):158–169, 2013. doi: 10.1111/cgf.12194

[42] M. P. Brundage, K. Morris, T. Sexton, S. Moccozet, and M. Hoffman. Developing maintenance key performance indicators from maintenance work order data. In *Proc. MSEC*. ASME, 2018. doi: 10.1115/msec2018-6492

[43] M. P. Brundage, T. Sexton, M. Hodkiewicz, A. Dima, and S. Lukens. Technical language processing: Unlocking maintenance knowledge. *Manufacturing Letters*, 27:42–46, 2021.

[44] M. P. Brundage, T. Sexton, M. Hodkiewicz, K. C. Morris, J. Arinez, F. Ameri, J. Ni, and G. Xiao. Where do we start? guidance for technology implementation in maintenance management for manufacturing. *J. of Manufacturing Science and Engineering*, 141(9):091005, 2019. doi: 10.1115/1.4044105

[45] Á. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau. FairVis: Visual analytics for discovering intersectional bias in machine

learning. In *Proceedings of 2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 46–56. IEEE, 2019.

[46] A. Camisetty, C. Chandurkar, M. Sun, and D. Koop. Enhancing web-based analytics applications through provenance. *IEEE transactions on visualization and computer graphics*, 25(1):131–141, 2018.

[47] M. Cammarano, X. Dong, B. Chan, J. Klingner, J. Talbot, A. Halevey, and P. Hanrahan. Visualization of heterogeneous data. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1200–1207, 2007. doi: 10.1109/tvcg.2007.70617

[48] D. B. Carr, R. J. Littlefield, W. Nicholson, and J. Littlefield. Scatterplot matrix techniques for large N. *J. of the American Statistical Association*, 82(398):424–436, 1987. doi: 10.1080/01621459.1987.10478445

[49] T. P. Carvalho, F. A. Soares, R. Vita, R. d. P. Francisco, J. P. Basto, and S. G. Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137:106024, 2019.

[50] M. Cavallo and Ç. Demiralp. Clustrophile 2: Guided visual clustering analysis. *arXiv preprint arXiv:1804.03048*, 2018.

[51] G. Y.-Y. Chan, P. Xu, Z. Dai, and L. Ren. ViBR: Visualizing bipartite relations at scale with the minimum description length principle. *IEEE Trans. on Visualization and Computer Graphics*, 25(1):321–330, 2019. doi: 10.1109/tvcg.2018.2864826

[52] S. Chandrasegaran, S. K. Badam, L. Kisselburgh, K. Peppler, N. Elmqvist, and K. Ramani. VizScribe: A visual analytics approach to understand designer behavior. *International J. of Human-Computer Studies*, 100:66–80, 2017. doi: 10.1016/j.ijhcs.2016.12.007

[53] S. Chandrasegaran, C. Bryan, H. Shidara, T.-Y. Chuan, and K.-L. Ma. TalkTraces: Real-time capture and visualization of verbal content in meetings. *The ACM CHI Conference on Human Factors in Computing Systems*, 2019 *(conditionally accepted)*.

[54] S. Chandrasegaran, X. Zhang, and K.-L. Ma. Using text visualization to aid analysis of machine maintenance logs. In *Proceedings of the Model-Based Enterprise Summit*, pp. 76–85, 2020.

[55] S. K. Chandrasegaran, K. Ramani, R. D. Sriram, I. Horváth, A. Bernard, R. F. Harik, and W. Gao. The evolution, challenges, and future of knowledge representation in product design systems. *Computer-aided design*, 45(2):204–228, 2013.

[56] A. Chatzimparmpas, R. M. Martins, K. Kucher, and A. Kerren. Stackgenvis: Alignment of data, algorithms, and models for stacking ensemble learning using performance metrics. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1547–1557, 2020.

[57] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[58] V. Chen, S. Wu, A. J. Ratner, J. Weng, and C. Ré. Slice-based learning: A programming model for residual learning in critical data slices. *Advances in neural information processing systems*, 32, 2019.

[59] Y. Chen, P. Xu, and L. Ren. Sequence synopsis: Optimize visual summary of temporal event data. *IEEE transactions on visualization and computer graphics*, 24(1):45–55, 2017.

[60] Z. Chen, Y. Wang, Q. Wang, Y. Wang, and H. Qu. Towards automated infographic design: Deep learning-based auto-extraction of extensible timeline. *IEEE transactions on visualization and computer graphics*, 26(1):917–926, 2019.

[61] S. Cheng, K. Mueller, and W. Xu. A framework to visualize temporal behavioral relationships in streaming multivariate data. In *Proceedings of New York Scientific Data Summit*, pp. 1–10. IEEE, 2016.

[62] N. Chibana. 50 gorgeous color schemes from award-winning websites. http://blog.visme.co/website-color-schemes/, 2016.

[63] I. Cho, W. Dou, D. X. Wang, E. Sauda, and W. Ribarsky. Vairoma: A visual analytics system for making sense of places, times, and events in roman history. *IEEE transactions on visualization and computer graphics*, 22(1):210–219, 2015.

[64] M. Choi, C. Park, S. Yang, Y. Kim, J. Choo, and S. R. Hong. Aila: Attentive interactive labeling assistant for document classification through attention-based deep neural networks. In *Proc. CHI*, pp. 1–12, 2019. doi: 10.1145/3290605.3300460

[65] A. K. Choudhary, J. A. Harding, and M. K. Tiwari. Data mining in manufacturing: a review based on the kind of knowledge. *Journal of Intelligent Manufacturing*, 20(5):501, 2009.

[66] D. H. Chung, M. L. Parry, I. W. Griffiths, R. S. Laramee, R. Bown, P. A. Legg, and M. Chen. Knowledge-assisted ranking: A visual analytic application for sports event data. *IEEE Computer Graphics and Applications*, 36(3):72–82, 2015.

[67] Y. Chung, T. Kraska, N. Polyzotis, K. H. Tae, and S. E. Whang. Automated data slicing for model validation: A big data-ai integration approach. *IEEE Transactions on Knowledge and Data Engineering*, 32(12):2284–2296, 2019.

[68] Y. Chung, T. Kraska, N. Polyzotis, K. H. Tae, and S. E. Whang. Slice finder: Automated data slicing for model validation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1550–1553. IEEE, 2019.

[69] C. B. Clement, M. Bierbaum, K. P. O'Keeffe, and A. A. Alemi. On the use of arxiv as a dataset. *CoRR*, abs/1905.00075, 2019.

[70] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld. SPECTER: document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2270–2282. Association for Computational Linguistics, misc, 2020. doi: 10.18653/v1/2020.acl-main.207

[71] D. Collaris and J. Van Wijk. Strategyatlas: Strategy analysis for machine learning interpretability. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[72] D. Collaris and J. J. van Wijk. Explainexplore: Visual exploration of machine learning explanations. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 26–35. IEEE, 2020.

[73] C. Collins, S. Carpendale, and G. Penn. DocuBurst: Visualizing document content using language structure. *Computer Graphics Forum*, 28(3):1039–1046, 2009.

[74] C. Collins, F. B. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *IEEE Symposium on Visual Analytics Science and Technology*, pp. 91–98, 2009.

[75] K. Cook, N. Cramer, D. Israel, M. Wolverton, J. Bruce, R. Burtner, and A. Endert. Mixed-initiative visual analytics using task-driven recommendations. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 9–16. IEEE, 2015.

[76] K. A. Cook and J. J. Thomas. Illuminating the path: The research and development agenda for visual analytics. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2005.

[77] Coolors. coolors. https://coolors.co/browser, 2018.

[78] A. Cozad, N. V. Sahinidis, and D. C. Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, 2014.

[79] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, and D. Zhang. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE transactions on visualization and computer graphics*, 26(1):906–916, 2019.

[80] S. Das, D. Cashman, R. Chang, and A. Endert. Beames: Interactive multimodel steering, selection, and inspection for regression tasks. *IEEE computer graphics and applications*, 39(5):20–32, 2019.

[81] A. Dasgupta, J. Poco, Y. Wei, R. Cook, E. Bertini, and C. T. Silva. Bridging theory with practice: An exploratory study of visualization use and design for climate model comparison. *IEEE transactions on visualization and computer graphics*, 21(9):996–1014, 2015.

[82] D. Datta, N. Self, J. Simeone23, A. Meadows, W. Outhwaite, L. Walker, N. Elmqvist, and N. Ramkrishnan. Timbersleuth: Visual anomaly detection with human feedback for mitigating the illegal timber trade. *Information Visualization*, 2022.

[83] M. F. De Oliveira and H. Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE transactions on visualization and computer graphics*, 9(3):378–394, 2003.

[84] M. Desmond, M. Muller, Z. Ashktorab, C. Dugan, E. Duesterwald, K. Brimijoin, C. Finegan-Dollak, M. Brachman, A. Sharma, N. N. Joshi, et al. Increasing the speed and accuracy of data labeling through an ai assisted interface. In *26th International Conference on Intelligent User Interfaces*, pp. 392–401, 2021.

[85] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding, 2019. doi: 10.18653/v1/N19-1423

[86] S. Di, R. Gupta, M. Snir, E. Pershey, and F. Cappello. Logaider: A tool for mining potential correlations of hpc log events. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pp. 442–451. IEEE, 2017.

[87] A. Dima, S. Lukens, M. Hodkiewicz, T. Sexton, and M. P. Brundage. Adapting natural language processing for technical text. *Applied AI Letters*, 2(2):e33, June 2021.

[88] A. Dima and A. Massey. Keyphrase extraction for technical language processing. *Journal of Research of National Institute of Standards and Technology*, 126(126053), Mar. 2022.

[89] D. Dingen, M. van't Veer, P. Houthuizen, E. H. Mestrom, E. H. Korsten, A. R. Bouwman, and J. Van Wijk. Regressionexplorer: Interactive exploration of logistic regression models with subgroup analysis. *IEEE transactions on visualization and computer graphics*, 25(1):246–255, 2018.

[90] D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. In *Proc. AMS Conf. on Math Challenges of the 21st Century*, 2000.

[91] B. Dorr, C. Monz, S. President, R. Schwartz, and D. Zajic. A methodology for extrinsic evaluation of text summarization: does rouge correlate? In *Proceedings*

of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 1–8. Association for Computational Linguistics, Ann Arbor, Michigan, 2005.

[92] M. Dowling, J. Wenskovitch, J. Fry, L. House, and C. North. Sirius: Dual, symmetric, interactive dimension reductions. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):172–182, 2018.

[93] M. Dudáš, S. Lohmann, V. Svátek, and D. Pavlov. Ontology visualization methods and tools: a survey of the state of the art. *The Knowledge Engineering Review*, 33, 2018.

[94] A. Dumitrache, L. Aroyo, and C. Welty. Achieving expert-level annotation quality with crowdtruth. In *Proc. of BDM2I Workshop, ISWC*, 2015.

[95] M. El-Assady, V. Gold, C. Acevedo, C. Collins, and D. Keim. ConToVi: multi-party conversation exploration using topic-space views. In *Computer Graphics Forum*, vol. 35, pp. 431–440. Wiley misc Library, 2016.

[96] M. El-Assady, R. Sevastjanova, B. Gipp, D. Keim, and C. Collins. Nerex: Named-entity relationship exploration in multi-party conversations. *Computer Graphics Forum*, 36(3):213–225, 2017.

[97] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679, 2021. doi: 10.1016/j.eswa.2020.113679

[98] N. Elmqvist and J. S. Yi. Patterns for visualization evaluation. *Information Visualization*, 14(3):250–269, 2015.

[99] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of KDD*, pp. 226–231, 1996.

[100] S. Evensen, C. Ge, and C. Demiralp. Ruler: Data programming by demonstration for document labeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1996–2005, 2020.

[101] R. Faust, D. Glickenstein, and C. Scheidegger. DimReader: Axis lines that explain non-linear projections. *IEEE Trans. on Visualization and Computer Graphics*, 25(1):481–490, 2018.

[102] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.

[103] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.

[104] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, et al. Knowledge discovery and data mining: Towards a unifying framework. In *Proceedings of KDD*, vol. 96, pp. 82–88, 1996.

[105] P. Federico, M. Wagner, A. Rind, A. Amor-Amorós, S. Miksch, and W. Aigner. The role of explicit knowledge: A conceptual model of knowledge-assisted visual analytics. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 92–103. IEEE, 2017.

[106] C. Felix, A. Dasgupta, and E. Bertini. The exploratory labeling assistant: Mixed-initiative label curation with large document collections. In *Proc. UIST*, pp. 153–164, 2018. doi: 10.1145/3242587.3242596

[107] C. Fellbaum, ed. *WordNet: An electronic lexical database*. MIT press, 1998.

[108] U. Feyyad. Data mining and knowledge discovery: Making sense out of data. *IEEE expert*, 11(5):20–25, 1996.

[109] J. Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, pp. 10–32, 1957.

[110] A. for the Advancement of Artificial Intelligence. The international conference on knowledge discovery and data mining. https://aaai.org/Conferences/KDD/kdd.php, 1995.

[111] K. Fort. *Collaborative Annotation for Reliable Natural Language Processing: Technical and Sociological Aspects*. John Wiley & Sons, 2016.

[112] A. Fouse, N. Weibel, E. Hutchins, and J. D. Hollan. ChronoViz: a system for supporting navigation of time-coded data. In *Proc. ACM Extended Abstracts on Human Factors in Computing Systems*, pp. 299–304, 2011. doi: 10.1145/1979742.1979706

[113] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus. Knowledge discovery in databases: An overview. *AI magazine*, 13(3):57–57, 1992.

[114] A. A. Freitas. A survey of evolutionary algorithms for data mining and knowledge discovery. In *Advances in evolutionary computing*, pp. 819–845. Springer, 2003.

[115] D. Fu, M. Chen, F. Sala, S. Hooper, K. Fatahalian, and C. Ré. Fast and three-rious: Speeding up weak supervision with triplet methods. In *International Conference on Machine Learning*, pp. 3280–3291. PMLR, 2020.

[116] T. Fujiwara, O.-H. Kwon, and K.-L. Ma. Supporting analysis of dimensionality reduction results with contrastive learning. *IEEE Trans. on Visualization and Computer Graphics*, 26(1):45–55, 2020. doi: 10.1109/tvcg.2019.2934251

[117] T. Fujiwara and T.-P. Liu. Contrastive multiple correspondence analysis (cMCA): Applying the contrastive learning method to identify political subgroups. *arXiv:2007.04540*, 2020.

[118] T. Fujiwara, N. Sakamoto, J. Nonaka, K. Yamamoto, K.-L. Ma, et al. A visual analytics framework for reviewing multivariate time-series data with dimensionality reduction. *arXiv:2008.01645*, 2020.

[119] T. Fujiwara, X. Wei, J. Zhao, and K.-L. Ma. Interactive dimensionality reduction for comparative analysis. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):758–768, 2021.

[120] K. Furmanová, A. Jurčík, B. Kozlíková, H. Hauser, and J. Byška. Multiscale visual drilldown for the analysis of large ensembles of multi-body protein complexes. *IEEE transactions on visualization and computer graphics*, 26(1):843–852, 2019.

[121] S. Gad, W. Javed, S. Ghani, N. Elmqvist, T. Ewing, K. N. Hampton, and N. Ramakrishnan. Themedelta: dynamic segmentations over temporal topic models. *IEEE Transactions on Visualization and Computer Graphics*, 21(5):672–685, 2015.

[122] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

[123] A. Ge, H. Jang, G. Carenini, K. Ho, and Y. J. Lee. Octvis: ontology-based comparison of topic models. In *2019 IEEE Visualization Conference (VIS)*, pp. 66–70. IEEE, 2019.

[124] M. Gleicher, A. Barve, X. Yu, and F. Heimerl. Boxer: Interactive comparison of classifier results. In *Computer Graphics Forum*, vol. 39, pp. 181–193. Wiley misc Library, 2020.

[125] M. Glueck, A. Gvozdik, F. Chevalier, A. Khan, M. Brudno, and D. Wigdor. Phenostacks: cross-sectional cohort phenotype comparison visualizations. *IEEE transactions on visualization and computer graphics*, 23(1):191–200, 2016.

[126] M. Glueck, P. Hamilton, F. Chevalier, S. Breslav, A. Khan, D. Wigdor, and M. Brudno. Phenoblocks: phenotype comparison visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):101–110, 2015.

[127] M. Glueck, M. P. Naeini, F. Doshi-Velez, F. Chevalier, A. Khan, D. Wigdor, and M. Brudno. Phenolines: phenotype comparison visualizations for disease subtyping via topic models. *IEEE transactions on visualization and computer graphics*, 24(1):371–381, 2017.

[128] K. Goel, N. Rajani, J. Vig, S. Tan, J. Wu, S. Zheng, C. Xiong, M. Bansal, and C. Ré. Robustness gym: Unifying the nlp evaluation landscape. *arXiv preprint arXiv:2101.04840*, 2021.

[129] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. In *Linear algebra*, pp. 134–151. Springer, 1971.

[130] J. R. Goodall, E. D. Ragan, C. A. Steed, J. W. Reed, G. D. Richardson, K. M. Huffer, R. A. Bridges, and J. A. Laska. Situ: Identifying and explaining suspicious behavior in networks. *IEEE transactions on visualization and computer graphics*, 25(1):204–214, 2018.

[131] J. Görtler, C. Schulz, D. Weiskopf, and O. Deussen. Bubble treemaps for uncertainty visualization. *IEEE transactions on visualization and computer graphics*, 24(1):719–728, 2017.

[132] C. C. Gramazio, J. Huang, and D. H. Laidlaw. An analysis of automated visual analysis classification: Interactive visualization task inference of cancer genomics domain experts. *IEEE Transactions on Visualization and Computer Graphics*, 24(8):2270–2283, 2017.

[133] B. Gretarsson, J. O'donovan, S. Bostandjiev, T. Höllerer, A. Asuncion, D. Newman, and P. Smyth. TopicNets: visual analysis of large text corpora with topic modeling. *ACM Transactions on Intelligent Systems and Technology*, 3(2):23, 2012. doi: 10.1145/2089094.2089099

[134] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[135] S. Grover. The two sides of the food crisis: Want and waste. https://www.treehugger.com/green-food/the-two-sides-of-the-food-crisis-want-waste-infographic.html, 2011.

[136] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993. doi: 10.1006/knac.1993.1008

[137] H. Guo, S. Di, R. Gupta, T. Peterka, and F. Cappello. La VALSE: Scalable log visualization for fault characterization in supercomputers. In *Proc. EGPGV*, pp. 91–100. Eurographics, 2018. doi: 10.5555/3293524.3293533

[138] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao. Eventthread: Visual summarization and stage analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):56–65, 2017.

[139] S. Gupta and S. K. Gupta. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65, 2019. doi: 10.1016/j.eswa.2018.12.011

[140] V. Gupta and G. S. Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010. doi: 10.4304/jetwi.2.3.258-268

[141] S. Ha, S. Monadjemi, R. Garnett, and A. Ottley. A unified comparison of user modeling techniques for predicting data interaction and detecting exploration bias. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[142] A. Hakone, L. Harrison, A. Ottley, N. Winters, C. Gutheil, P. K. Han, and R. Chang. Proact: Iterative design of a patient-centered visualization for effective prostate cancer health risk communication. *IEEE transactions on visualization and computer graphics*, 23(1):601–610, 2016.

[143] H. Hamooni, B. Debnath, J. Xu, H. Zhang, G. Jiang, and A. Mueen. Logmine: Fast pattern recognition for log analytics. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 1573–1582, 2016.

[144] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *Data mining and knowledge discovery*, 15(1):55–86, 2007.

[145] S. Hariharan and R. Srinivasan. Studies on intrinsic summary evaluation. *International Journal of Artificial Intelligence and Soft Computing*, 2(1-2):58–76, 2010. doi: 10.1504/IJAISC.2010.032513

[146] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2

[147] L. Harrison, K. Reinecke, and R. Chang. Infographic aesthetics: Designing for the first impression. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1187–1190, 2015.

[148] M. Harrower and C. A. Brewer. ColorBrewer.org: An misc tool for selecting colour schemes for maps. *Cartographic J.*, 40(1):27–37, 2003.

[149] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, vol. 52, pp. 139–183. Elsevier, 1988.

[150] W. He, L. Zou, A. K. Shekar, L. Gou, and L. Ren. Where can we help? A visual analytics approach to diagnosing and improving semantic segmentation of movable objects. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1040–1050, 2021.

[151] U. Hébert-Johnson, M. Kim, O. Reingold, and G. Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pp. 1939–1948. PMLR, 2018.

[152] F. Heimerl and M. Gleicher. Interactive analysis of word vector embeddings. *Computer Graphics Forum*, 37(3):253–265, 2018. doi: 10.1111/cgf.13417

[153] S. Hellmann. Dbpedia lookup | dbpedia. https://wiki.dbpedia.org/lookup, April 2015. Accessed October 3, 2019.

[154] N. Henry, J.-D. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics*, 13(6):1302–1309, 2007.

[155] T. Hirao, Y. Sasaki, and H. Isozaki. An extrinsic evaluation for question-biased text summarization on QA tasks. In *Proceedings of the NAACL Workshop on Automatic Summarization*, pp. 61–68. Association for Computational Linguistics, Pittsburgh, PA, 2001.

[156] B. Hjørland and H. Albrechtsen. Toward a new horizon in information science: Domain-analysis. *Journal of the American society for information science*, 46(6):400–425, 1995.

[157] M. Hodkiewicz and M. T.-W. Ho. Cleaning historical maintenance work order data for reliability analysis. *J. of Quality in Maintenance Engineering*, 22(2):146–163, 2016. doi: 10.1108/jqme-04-2015-0013

[158] A. Holzinger and G. Pasi. *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data: Third International Workshop, HCI-KDD 2013, Held at SouthCHI 2013, Maribor, Slovenia, July 1-3, 2013, Proceedings*, vol. 7947. Springer, 2013.

[159] M. N. Hoque, M. Ehtesham-Ul-Haque, N. Elmqvist, and S. M. Billah. Accessible data representation with natural sound. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, 2023.

[160] J. Howard. The wonderful and terrifying implications of computers that can learn. `https://www.ted.com/talks/jeremy_howard_the_wonderful_and_terrifying_implications_of_computers_that_can_learn?language=en`, 2014. Accessed December 3, 2019.

[161] W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pp. 2029–2037. PMLR, 2018.

[162] J. Hullman, E. Adar, and P. Shah. The impact of social information on visual judgments. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1461–1470, 2011.

[163] IPAM. Workshop: Deep Learning and Combinatorial Optimization. http://www.ipam.ucla.edu/programs/workshops/deep-learning-and-combinatorial-optimization/. publisher: IPAM 2021.

[164] D. Jain, M. D. Borah, and A. Biswas. Summarization of legal documents: Where are we now and the way forward. *Computer Science Review*, 40:100388, 2021. doi: 10.1016/j.cosrev.2021.100388

[165] Jeff Howe. Crowdsourcing: A definition. `https://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html`, 2006. [misc; accessed 28-Oct-2022].

[166] X. Jin, B. A. Weiss, D. Siegel, and J. Lee. Present status and future growth of advanced maintenance technology and strategy in us manufacturing. *International J. of Prognostics and Health Management*, 7(Special Issue on Smart Manufacturing PHM), 2016.

[167] Joblib Development Team. Joblib: running python functions as pipeline jobs. https://joblib.readthedocs.io/, 2020.

[168] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. doi: 10.1109/TBDATA.2019.2921572

[169] P. Joia, F. Petronetto, and L. G. Nonato. Uncovering representative groups in multidimensional projections. *Computer Graphics Forum*, 34(3):281–290, 2015. doi: 10.1111/cgf.12640

[170] D. Jordan, M. Steiner, E. F. Kochs, and G. Schneider. A program for computing the prediction probability and the related receiver operating characteristic graph. *Anesthesia & Analgesia*, 111(6):1416–1421, 2010.

[171] M. Kahng, N. Thorat, D. H. P. Chau, F. B. Viégas, and M. Wattenberg. Gan lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE transactions on visualization and computer graphics*, 25(1):1–11, 2018.

[172] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer. Enterprise data analysis and visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2917–2926, 2012.

[173] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology visualization methods—a survey. *ACM Computing Surveys*, 39(4):10, 2007.

[174] S. Kaul, D. Borland, N. Cao, and D. Gotz. Improving visualization interpretation using counterfactuals. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):998–1008, 2021.

[175] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[176] T. Kehrenberg, M. Bartlett, O. Thomas, and N. Quadrianto. Null-sampling for interpretable and fair representations. In *European Conference on Computer Vision*, pp. 565–580. Springer, 2020.

[177] D. A. Keim. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[178] D. A. Keim and D. Oelke. Literature fingerprinting: A new method for visual literary analysis. In *IEEE Symposium on Visual Analytics Science and Technology*, pp. 115–122, 2007. doi: 10.1109/VAST.2007.4389004

[179] S. Kenner. 17 apr infographics best practices. http://grasshoppermarketing.com/infographics-best-practices/, 2014.

[180] S. P. Kesavan, T. Fujiwara, J. K. Li, C. Ross, M. Mubarak, C. D. Carothers, R. B. Ross, and K.-L. Ma. A visual analytics framework for reviewing streaming performance data. In *IEEE Pacific Visualization Symposium (PacificVis)*, pp. 206–215, 2020.

[181] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*. OpenReview.net, Addis Ababa, Ethiopia, 2020.

[182] D. H. Kim, E. Hoque, J. Kim, and M. Agrawala. Facilitating document reading by linking text and tables. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp. 423–434. Association for Computing Machinery, New York, NY, 2018. doi: 10.1145/3242587.3242617

[183] M. P. Kim, A. Ghorbani, and J. Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 247–254, 2019.

[184] N. W. Kim, E. Schweickart, Z. Liu, M. Dontcheva, W. Li, J. Popovic, and H. Pfister. Data-driven guides: Supporting expressive design for information graphics. *IEEE transactions on visualization and computer graphics*, 23(1):491–500, 2016.

[185] A. Kotriwala, B. Klöpper, M. Dix, G. Gopalakrishnan, D. Ziobro, and A. Potschka. Xai for operations in the process industry-applications, theses, and research directions. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*, 2021.

[186] P. Kovesi. Good colour maps: How to design them. *arXiv preprint arXiv:1509.03700*, 2015.

[187] K. Kowa. Spooky student loans statistics infographic-sofi. https://visual.ly/community/infographic/business/spooky-student-loans-statistics-infographic—sofi, 2015.

[188] K. Kucher, C. Paradis, and A. Kerren. The state of the art in sentiment visualization. *Computer Graphics Forum*, 37(1):71–96, 2018. doi: 10.1111/cgf.13217

[189] B. C. Kwon, B. Eysenbach, J. Verma, K. Ng, C. De Filippi, W. F. Stewart, and A. Perer. Clustervision: Visual supervision of unsupervised clustering. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):142–151, 2017. doi: 10.1109/tvcg.2017.2745085

[190] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nmiscar axis mapping of visualizations through user drawings. *IEEE transactions on visualization and computer graphics*, 23(1):221–230, 2016.

[191] S. Lai, K. Liu, S. He, and J. Zhao. How to generate a good word embedding. *IEEE Intelligent Systems*, 31(6):5–14, 2016.

[192] V. Lai, A. Smith-Renner, K. Zhang, R. Cheng, W. Zhang, J. Tetreault, and A. Jaimes-Larrarte. An exploration of post-editing effectiveness in text summarization. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 475–493. Association for Computational Linguistics, Seattle, United States, 2022. doi: 10.18653/v1/2022.naacl-main.35

[193] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

[194] P.-M. Law, Z. Liu, S. Malik, and R. C. Basole. Maqui: Interweaving queries and pattern mining for recursive event sequence exploration. *IEEE transactions on visualization and computer graphics*, 25(1):396–406, 2018.

[195] M. Le Goc, C. Perin, S. Follmer, J.-D. Fekete, and P. Dragicevic. Dynamic composite data physicalization using wheeled micro-robots. *IEEE transactions on visualization and computer graphics*, 25(1):737–747, 2018.

[196] B. Lee, A. Srinivasan, P. Isenberg, J. Stasko, et al. Post-wimp interaction for information visualization. *Foundations and Trends® in Human–Computer Interaction*, 14(1):1–95, 2021.

[197] D. J.-L. Lee, J. Lee, T. Siddiqui, J. Kim, K. Karahalios, and A. Parameswaran. You can't always sketch what you want: Understanding sensemaking in visual query systems. *IEEE transactions on visualization and computer graphics*, 26(1):1267–1277, 2019.

[198] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, and C. Bizer. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[199] D. Levy, Y. Carmon, J. C. Duchi, and A. Sidford. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33:8847–8860, 2020.

[200] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880. Association for Computational Linguistics, misc, 2020. doi: 10.18653/v1/2020.acl-main.703

[201] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister. Upset: visualization of intersecting sets. *IEEE transactions on visualization and computer graphics*, 20(12):1983–1992, 2014.

[202] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[203] H. Li, L. Ying, H. Zhang, Y. Wu, H. Qu, and Y. Wang. Notable: On-the-fly assistant for data storytelling in computational notebooks. *arXiv preprint arXiv:2303.04059*, 2023.

[204] H. Li, J. Zhu, J. Zhang, C. Zong, and X. He. Keywords-guided abstractive sentence summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8196–8203, 2020. doi: 10.1609/aaai.v34i05.6333

[205] J. K. Li, T. Fujiwara, S. P. Kesavan, C. Ross, M. Mubarak, C. D. Carothers, R. B. Ross, and K.-L. Ma. A visual analytics framework for analyzing parallel and distributed computing applications. In *IEEE Visualization in Data Science*, pp. 1–9, 2019.

[206] Z. Li, C. Zhang, S. Jia, and J. Zhang. Galex: Exploring the evolution and intersection of disciplines. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1182–1192, 2019.

[207] E. Liberty, Z. Karnin, B. Xiang, L. Rouesnel, B. Coskun, R. Nallapati, J. Delgado, A. Sadoughi, Y. Astashonok, P. Das, et al. Elastic machine learning algorithms in amazon sagemaker. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 731–737, 2020.

[208] A. Y. Lin, J. Ford, E. Adar, and B. Hecht. VizByWiki: mining data visualizations from the web to enrich news articles. In *Proceedings of the World Wide Web Conference*, pp. 873–882. ACM, New York, NY, 2018. doi: 10.1145/3178876.3186135

[209] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81. Association for Computational Linguistics, Barcelona, Spain, 2004.

[210] C.-Y. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*, pp. 495–501. Association for Computational Linguistics, misc, 2000. doi: 10.3115/990820.990892

[211] P. Lindstrom, D. Koller, W. Ribarsky, L. F. Hodges, N. Faust, and G. A. Turner. Real-time, continuous level of detail rendering of height fields. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 109–118, 1996.

[212] P. Lison, J. Barnes, and A. Hubin. skweak: Weak supervision made easy for nlp, 2021.

[213] M. Liu, S. Liu, X. Zhu, Q. Liao, F. Wei, and S. Pan. An uncertainty-aware approach for exploratory microblog retrieval. *IEEE transactions on visualization and computer graphics*, 22(1):250–259, 2015.

[214] S. Liu, P.-T. Bremer, J. J. Thiagarajan, V. Srikumar, B. Wang, Y. Livnat, and V. Pascucci. Visual exploration of semantic relationships in neural word embeddings. *IEEE transactions on visualization and computer graphics*, 24(1):553–562, 2017. doi: 10.1109/TVCG.2017.2745141

[215] S. Liu, C. Chen, Y. Lu, F. Ouyang, and B. Wang. An interactive method to improve crowdsourced annotations. *IEEE transactions on visualization and computer graphics*, 25(1):235–245, 2018.

[216] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Trans. on Visualization and Computer Graphics*, 23(3):1249–1268, 2017. doi: 10.1109/tvcg.2016.2640960

[217] S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu. Storyflow: Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2436–2445, 2013.

[218] W. Liu. Knowledge exploitation, knowledge exploration, and competency trap. *Knowledge and Process Management*, 13(3):144–161, 2006.

[219] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[220] Z. Liu, J. Thompson, A. Wilson, M. Dontcheva, J. Delorey, S. Grigg, B. Kerr, and J. Stasko. Data illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2018.

[221] E. Loper and S. Bird. NLTK: The natural language toolkit. In *Proc. ETMTNLP*, pp. 63–70. ACL, 2002. doi: 10.3115/1118108.1118117

[222] Z. Lu, M. Fan, Y. Wang, J. Zhao, M. Annett, and D. Wigdor. Inkplanner: Supporting prewriting via intelligent visual diagramming. *IEEE transactions on visualization and computer graphics*, 25(1):277–287, 2018.

[223] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958. doi: 10.1147/rd.22.0159

[224] S. Lukens, M. Naik, K. Saetia, and X. Hu. Best practices framework for improving maintenance data quality to enable asset performance analytics. *Annual Conference of the PHM Society*, 11(1), 2019.

[225] M. A. Lyles and C. R. Schwenk. Top management, strategy and organizational knowledge structures. *Journal of management studies*, 29(2):155–174, 1992.

[226] J. Ma, K.-L. Ma, and J. Frazier. Decoding a complex visualization in a science museum–an empirical study. *IEEE transactions on visualization and computer graphics*, 26(1):472–481, 2019.

[227] E. Mahfoud, K. Wegba, Y. Li, H. Han, and A. Lu. Immersive visualization for abnormal detection in heterogeneous data for on-site decision making. In *Proc. HICSS*, 2018. doi: 10.24251/hicss.2018.160

[228] G. E. Marai, J. Leigh, and A. Johnson. Immersive analytics lessons from the electronic visualization laboratory: a 25-year perspective. *IEEE computer graphics and applications*, 39(3):54–66, 2019.

[229] J. G. March. Exploration and exploitation in organizational learning. *Organization science*, 2(1):71–87, 1991.

[230] V. Marin, S. Jialin, F. Aaron, A. Brandon, M. Georg, D. Bistra, and Y. Yisong. Workshop: Learning Meets Combinatorial Algorithms. https://nips.cc/Conferences/2020/ScheduleMultitrack?event=16128. publisher: NeurIPS 2020.

[231] C. J. Matheus, P. K. Chan, and G. Piatetsky-Shapiro. Systems for knowledge discovery in databases. *IEEE Transactions on knowledge and data engineering*, 5(6):903–913, 1993.

[232] F. Matsuda, Y. Shinbo, A. Oikawa, M. Y. Hirai, O. Fiehn, S. Kanaya, and K. Saito. Assessment of metabolome annotation quality: a method for evaluating the false discovery rate of elemental composition searches. *PLoS One*, 4(10):e7490, 2009.

[233] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*, 2018.

[234] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *CoRR*, 1802.03426, 2018. doi: 10.48550/ARXIV.1802.03426

[235] B. Media. Picture this-the infographic. https://visual.ly/community/infographic/business/pic infographic, 2013.

[236] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[237] D. Mekala, X. Zhang, and J. Shang. Meta: Metadata-empowered weak supervision for text classification. In *Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[238] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the ACM International Conference on Semantic Systems*, p. 1–8. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/2063518.2063519

[239] D. Miller. BERT extractive summarizer. https://github.com/dmmiller612/bert-extractive-summarizer, 2019.

[240] J. P. Miller, R. Taori, A. Raghunathan, S. Sagawa, P. W. Koh, V. Shankar, P. Liang, Y. Carmon, and L. Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pp. 7721–7735. PMLR, 2021.

[241] Y. Ming, P. Xu, F. Cheng, H. Qu, and L. Ren. Protosteer: Steering deep sequence model with prototypes. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):238–248, 2019.

[242] Y. Ming, P. Xu, H. Qu, and L. Ren. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 903–913, 2019.

[243] I. Misra and L. v. d. Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020.

[244] S. Mitra, S. K. Pal, and P. Mitra. Data mining in soft computing framework: a survey. *IEEE transactions on neural networks*, 13(1):3–14, 2002.

[245] B. Mobasher, N. Jain, E.-H. Han, and J. Srivastava. Web mining: Pattern discovery from world wide web transactions. Technical report, Technical Report TR96-050, Department of Computer Science, University of Minnesota, 1996.

[246] C. "moot" Poole. The case for anonymity misc. https://www.ted.com/talks/christopher_moot_poole_the_case_for_anonymity_misc, 2010. Accessed December 3, 2019.

[247] F. Moramarco, A. Papadopoulos Korfiatis, A. Savkov, and E. Reiter. A preliminary study on evaluating consultation notes with post-editing. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pp. 62–68. Association for Computational Linguistics, misc, 2021.

[248] F. Moretti. *Graphs, maps, trees: abstract models for a literary history*. Verso, 2005.

[249] M. Motamedi, N. Sakharnykh, and T. Kaldewey. A data-centric approach for training deep neural networks with less data. *arXiv preprint arXiv:2110.03613*, 2021.

[250] T. Munzner. A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics*, 15(6):921–928, 2009.

[251] T. Munzner. *Visualization analysis and design*. CRC press, 2014.

[252] G. Murray, T. Kleinbauer, P. Poller, T. Becker, S. Renals, and J. Kilgour. Extrinsic summarization evaluation: A decision audit task. *ACM Transactions on Speech and Language Processing*, 6(2):1–29, 2009. doi: 10.1145/1596517.1596518

[253] V. Nagarajan, A. Andreassen, and B. Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.

[254] A. Narechania, A. Karduni, R. Wesslen, and E. Wall. vitaLITy: promoting serendipitous discovery of academic literature with transformers & visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):486–496, 2021. doi: 10.1109/TVCG.2021.3114820

[255] H. T. P. Nguyen, A. Bhatele, N. Jain, S. Kesavan, H. Bhatia, T. Gamblin, K.-L. Ma, and P.-T. Bremer. Visualizing hierarchical performance profiles of parallel codes using callflow. *IEEE transactions on visualization and computer graphics*, 2019.

[256] C. Nobre, N. Gehlenborg, H. Coon, and A. Lex. Lineage: Visualizing multivariate clinical data in genealogy graphs. *IEEE transactions on visualization and computer graphics*, 25(3):1543–1558, 2018.

[257] I. Nonaka and H. Takeuchi. *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford university press, 1995.

[258] L. Norooz, M. L. Mauriello, A. Jorgensen, B. McNally, and J. E. Froehlich. BodyVis: a new approach to body learning through wearable sensing and visualization. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pp. 1025–1034. ACM, New York, NY, 2015. doi: 10.1145/2702123.2702299

[259] C. G. Northcutt, A. Athalye, and J. Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021.

[260] D. Oelke, D. Spretke, A. Stoffel, and D. A. Keim. Visual readability analysis: How to make your writings easier to read. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):662–674, 2011.

[261] D. Oelke, H. Strobelt, C. Rohrdantz, I. Gurevych, and O. Deussen. Comparative exploration of document collections: a visual analytics approach. In *Computer Graphics Forum*, vol. 33, pp. 201–210, 2014.

[262] The psychology of poverty and its impact on mental health in america. http://www.bestmswprograms.com/mental-health/.

[263] J. P. Ono, S. Castelo, R. Lopez, E. Bertini, J. Freire, and C. Silva. Pipelineprofiler: A visual analytics tool for the exploration of automl pipelines. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):390–400, 2020.

[264] OpenAI. Clip: Connecting text and images. `https://openai.com/research/clip`, 2023. Accessed: 2023-4-3.

[265] OpenAI. Dall·e 2. `https://openai.com/product/dall-e-2`, 2023. Accessed: 2023-4-3.

[266] OpenAI. Gpt-4 technical report, 2023.

[267] OpenAI. https://openai.com/blog/chatgpt. `https://openai.com/blog/chatgpt`, 2023. Accessed: 2023-4-3.

[268] Y. Oren, S. Sagawa, T. B. Hashimoto, and P. Liang. Distributionally robust language modeling. *arXiv preprint arXiv:1909.02060*, 2019.

[269] T. pandas development team. pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.3509134, Feb. 2020. doi: 10.5281/zenodo.3509134

[270] C. Panigutti, A. Perotti, and D. Pedreschi. Doctor xai: an ontology-based approach to black-box sequential data classification explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pp. 629–639, 2020.

[271] C. Papadopoulos, I. Gutenko, and A. E. Kaufman. Veevvie: visual explorer for empirical visualization, vr and interaction experiments. *IEEE transactions on visualization and computer graphics*, 22(1):111–120, 2015.

[272] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmqvist. ConceptVector: Text visual analytics via interactive lexicon building using word embedding. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):361–370, 2017. doi: 10.1109/tvcg.2017.2744478

[273] H. Park, N. Das, R. Duggal, A. P. Wright, O. Shaikh, F. Hohman, and D. H. P. Chau. Neurocartography: Scalable automatic visual summarization of concepts in deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):813–823, 2021. doi: 10.1109/TVCG.2021.3114858

[274] E. Pastor, L. de Alfaro, and E. Baralis. Identifying biased subgroups in ranking and classification. *arXiv preprint arXiv:2108.07450*, 2021.

[275] E. Pastor, L. de Alfaro, and E. Baralis. Looking for trouble: Analyzing classifier behavior via pattern divergence. In *Proceedings of the 2021 International Conference on Management of Data*, pp. 1400–1412, 2021.

[276] E. Pastor, A. Gavgavian, E. Baralis, and L. de Alfaro. How divergent is your data? *Proceedings of the VLDB Endowment*, 14(12):2835–2838, 2021.

[277] B. Patnaik, A. Batch, and N. Elmqvist. Information olfactation: Harnessing scent to convey data. *IEEE transactions on visualization and computer graphics*, 25(1):726–736, 2018.

[278] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *J. of Machine Learning Research*, 12:2825–2830, 2011.

[279] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[280] Y. Peng, G. Kou, Y. Shi, and Z. Chen. A descriptive framework for the field of data mining and knowledge discovery. *International Journal of Information Technology & Decision Making*, 7(04):639–682, 2008.

[281] J. W. Pennebaker, C. K. Chung, M. Ireland, A. Gonzales, and R. J. Booth. The development and psychometric properties of LIWC2007. LIWC Manual, `https://www.liwc.net/LIWC2007LanguageManual.pdf`. Accessed: 2020-4-30.

[282] H. Pohl. Necktie infographic. https://visual.ly/community/infographic/necktie-infographic, 2014.

[283] H. C. Purchase, K. Isaacs, T. Bueti, B. Hastings, A. Kassam, A. Kim, and S. van Hoesen. A classification of infographics. In *International Conference on Theory and Application of Diagrams*, pp. 210–218. Springer, 2018.

[284] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019.

[285] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. doi: 10.5555/3455716.3455856

[286] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350, 2001. doi: 10.1007/S007780100057

[287] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, vol. 11, p. 269. NIH Public Access, 2017.

[288] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29, 2016.

[289] M. Rauf, S. Padó, and M. Pradel. Meta learning for code summarization. *CoRR*, 2201.08310, 2022. doi: 10.48550/arxiv.2201.08310

[290] K. Reda, A. Febretti, A. Knoll, J. Aurisano, J. Leigh, A. Johnson, M. E. Papka, and M. Hereld. Visualizing large, heterogeneous data in hybrid-reality environments. *IEEE Computer Graphics and Applications*, 33(4):38–48, 2013. doi: 10.1109/MCG.2013.37

[291] R. Řehůřek and P. Sojka. Software framework for topic modelling with large corpora. In *Proc. LREC Workshop on New Challenges for NLP Frameworks*, pp. 45–50, 2010.

[292] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China, 2019. doi: 10.18653/v1/D19-1410

[293] S. Ribecca. The data visualisation catalogue. https://datavizcatalogue.com/search.html, 2018.

[294] M. T. Ribeiro, S. Singh, and C. 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, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016.

[295] T. Rietz and A. Maedche. Cody: An ai-based system to semi-automate coding for qualitative research. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2021.

[296] J. Risch, A. Kao, S. R. Poteet, and Y.-J. J. Wu. Text visualization for visual text analytics. In *Visual data mining*, pp. 154–171. Springer, 2008.

[297] rmmojado. Seo vs ppc. https://visual.ly/community/infographic/social-media/-infographic, 2011.

[298] J. C. Roberts, P. D. Ritsos, J. R. Jackson, and C. Headleand. The explanatory visualization framework: An active learning framework for teaching creative computing using explanatory visualizations. *IEEE transactions on Visualization and Computer Graphics*, 24(1):791–801, 2017.

[299] B.-M. Roh, S. R. Kumara, T. W. Simpson, and P. Witherell. Ontology-based laser and thermal metamodels for metal-based additive manufacturing. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 21–24, 2016.

[300] M. Röhlig, M. Luboschik, F. Krüger, T. Kirste, H. Schumann, M. Bögl, B. Alsallakh, and S. Miksch. Supporting activity recognition by visual analytics. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 41–48. IEEE, 2015.

[301] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):241–250, 2017.

[302] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):241–250, 2017. doi: 10.1109/tvcg.2016.2598495

[303] S. Sagadeeva and M. Boehm. Sliceline: Fast, linear-algebra-based slice finding for ml model debugging. In *Proceedings of the 2021 International Conference on Management of Data*, pp. 2290–2299, 2021.

[304] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[305] A. A. Salatino, T. Thanapalasingam, A. Mannocci, F. Osborne, and E. Motta. The computer science ontology: a large-scale taxonomy of research areas. In *International Semantic Web Conference*, pp. 187–205, 2018.

[306] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher. What do we talk about when we talk about dashboards? *IEEE transactions on visualization and computer graphics*, 25(1):682–692, 2018.

[307] A. Sarkar. education around the world. https://board-12thresults.in/education-around-world/education-around-the-world/, 2017.

[308] A. Sarvghad, M. Tory, and N. Mahyar. Visualizing dimension coverage to support exploratory analysis. *IEEE transactions on visualization and computer graphics*, 23(1):21–30, 2016.

[309] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. In *Computer Graphics Forum*, vol. 33, pp. 351–360. Wiley misc Library, 2014.

[310] Y. Scherrer. TaPaCo: A corpus of sentential paraphrases for 73 languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 6868–6873. European Language Resources Association, Marseille, France, 2020.

[311] P. Schober, C. Boer, and L. A. Schwarte. Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5):1763–1768, 2018.

[312] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE transactions on visualization and computer graphics*, 16(6):1139–1148, 2010.

[313] T. Sexton, M. P. Brundage, M. Hoffman, and K. C. Morris. Hybrid datafication of maintenance logs from ai-assisted human tags. In *Proc. IEEE Big Data*, pp. 1769–1777, 2017. doi: 10.1109/bigdata.2017.8258120

[314] T. Sexton, M. Hodkiewicz, and M. P. Brundage. Categorization errors for data entry in maintenance work-orders. *Proc. PHM*, 11(1), 2019. doi: 10.36001/phmconf.2019.v11i1.790

[315] T. B. Sexton and M. P. Brundage. Nestor: A tool for natural language annotation of short texts. *J. of Research of National Institute of Standards and Technology*, 124:124029, 2019. doi: 10.6028/jres.124.029

[316] M. J. Shaw, C. Subramaniam, G. W. Tan, and M. E. Welge. Knowledge management and data mining for marketing. *Decision support systems*, 31(1):127–137, 2001.

[317] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy. Neural abstractive text summarization with sequence-to-sequence models. *ACM Transactions on Data Science*, 2(1):1–37, 2021. doi: 10.1145/3419106

[318] Shilpika, B. Lusch, M. Emani, V. Vishwanath, M. E. Papka, and K.-L. Ma. MELA: A visual analytics tool for studying multifidelity hpc system logs. In *Proc. DAAC*, pp. 13–18. IEEE, 2019.

[319] M. Shimabukuro, J. Zipf, M. El-Assady, and C. Collins. H-matrix: Hierarchical matrix for visual analysis of cross-linguistic features in large learner corpora. In *2019 IEEE Visualization Conference (VIS)*, pp. 61–65. IEEE, 2019.

[320] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan. Big data clustering: A review. In *Proc. ICCSA*, pp. 707–720, 2014. doi: 10.1007/978-3-319-09156-3_49

[321] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[322] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE symposium on visual languages*, pp. 336–343. IEEE, 1996.

[323] B. Shneiderman. Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human–Computer Interaction*, 36(6):495–504, 2020.

[324] A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proceedings of KDD*, vol. 95, pp. 275–281, 1995.

[325] A. Silberschatz and A. Tuzhilin. User-assisted knowledge discovery: How much should the user be involved. In *In Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, 1996.

[326] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and data engineering*, 8(6):970–974, 1996.

[327] W. V. Siricharoen. Infographics: the new communication tools in digital age. In *The international conference on e-technologies and business on the web (ebw2013)*, pp. 169–174, 2013.

[328] D. Skau and R. Kosara. Readability and precision in pictorial bar charts. In *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers*, pp. 91–95, 2017.

[329] SMACK. Collinson latitude infographic. https://visual.ly/community/infographic/business/collinson-latitude-infographic, 2015.

[330] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg. Embedding projector: Interactive visualization and interpretation of embeddings. *CoRR*, 1611.05469, 2016. doi: 10.48550/arxiv.1611.05469

[331] J. Soo Yi, R. Melton, J. Stasko, and J. A. Jacko. Dust & magnet: multivariate information visualization using a magnet metaphor. *Information visualization*, 4(4):239–256, 2005. doi: 10.1057/palgrave.ivs.9500099

[332] R. Speer, J. Chin, and C. Havasi. ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proc. AAAI*, pp. 4444–4451, 2017.

[333] F. Sperrle, R. Sevastjanova, R. Kehlbeck, and M. El-Assady. Viana: Visual interactive annotation of argumentation. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 11–22. IEEE, 2019.

[334] M. Srivastava, T. Hashimoto, and P. Liang. Robustness to spurious correlations via human annotations. In *International Conference on Machine Learning*, pp. 9109–9119. PMLR, 2020.

[335] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Trans. on Visualization and Computer Graphics*, 22(1):629–638, 2016. doi: 10.1109/tvcg.2015.2467717

[336] J. Stasko, R. Catrambone, M. Guzdial, and K. McDonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. *International journal of human-computer studies*, 53(5):663–694, 2000.

[337] J. Stasko, C. Görg, and Z. Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization*, 7(2):118–132, 2008.

[338] H. Stitz, S. Gratzl, H. Piringer, T. Zichner, and M. Streit. Knowledgepearls: Provenance-based visualization retrieval. *IEEE transactions on visualization and computer graphics*, 25(1):120–130, 2018.

[339] P. J. Stone, D. C. Dunphy, and M. S. Smith. *The general inquirer: A computer approach to content analysis.* MIT press, 1966.

[340] M.-A. Storey, M. Musen, J. Silva, C. Best, N. Ernst, R. Fergerson, and N. Noy. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé. In *Workshop on interactive tools for knowledge capture*, vol. 73, 2001.

[341] M. Sun, P. Mi, C. North, and N. Ramakrishnan. Biset: Semantic edge bundling with biclusters for sensemaking. *IEEE transactions on visualization and computer graphics*, 22(1):310–319, 2015.

[342] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi. Machine learning for predictive maintenance: A multi-classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820, 2014.

[343] G. K. Tam, V. Kothari, and M. Chen. An analysis of machine-and human-analytics in classification. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):71–80, 2017.

[344] J. Tan, X. Wan, and J. Xiao. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1171–1181. Association for Computational Linguistics, misc, 2017.

[345] J. Tang, J. Liu, M. Zhang, and Q. Mei. Visualizing large-scale and high-dimensional data. In *Proc. IW3C2*, pp. 287–297. International World Wide Web Conferences Steering Committee, 2016.

[346] T. Tang, Y. Wu, Y. Wu, L. Yu, and Y. Li. Videomoderator: A risk-aware framework for multimodal video moderation in e-commerce. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):846–856, 2021.

[347] **Xiaoyu Zhang**, S. Chandrasegaran, and K.-L. Ma. Conceptscope: Organizing and visualizing knowledge in documents based on domain ontology. In *Proceedings of the 2021 chi conference on human factors in computing systems*, pp. 1–13, 2021.

[348] **Xiaoyu Zhang**, T. Fujiwara, S. Chandrasegaran, M. P. Brundage, T. Sexton, A. Dima, and K.-L. Ma. A visual analytics approach for the diagnosis of heterogeneous and multidimensional machine maintenance data. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*, pp. 196–205. IEEE, 2021.

[349] **Xiaoyu Zhang**, K. Li, P.-W. Chi, S. Chandrasegaran, and K.-L. Ma. Concepteva: Concept-based interactive exploration and customization of document summaries. In *CHI Conference on Human Factors in Computing Systems 2023*, 2023.

[350] **Xiaoyu Zhang**, J. Ono, H. Song, L. Gou, K.-L. Ma, and L. Ren. Sliceteller: A data slice-driven approach for machine learning model validation. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[351] **Xiaoyu Zhang**, X. Xuan, A. Dima, T. Sexton, and K.-L. Ma. A visual analytics approach for the diagnosis of heterogeneous and multidimensional machine maintenance data. In *conditionally accepted by 2023 IEEE 16th Pacific Visualization Symposium (PacificVis)*, 2023.

[352] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 02 1999. doi: 10.1162/089976699300016728

[353] E. Tokpo, P. Delobelle, B. Berendt, and T. Calders. How far can it go?: On intrinsic gender bias mitigation for text classification. *arXiv preprint arXiv:2301.12855*, 2023.

[354] E. Tsang. *Foundations of constraint satisfaction: the classic text*. BoD–Books on Demand, 2014.

[355] Z. Tufekci. Machine intelligence makes human morals more important. `https://www.ted.com/talks/zeynep_tufekci_machine_intelligence_makes_human_morals_more_important/transcript?referrer=playlist-talks_on_artificial_intelligen#t-3550`, 2016. Accessed October, 2020.

[356] J. W. Tukey. *Exploratory data analysis*, vol. 2. Reading, MA, 1977.

[357] J. Turian, L. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proc. ACL*, pp. 384–394, 2010. doi: 10.5555/1858681.1858721

[358] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *J. of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[359] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[360] L. van der Maaten, E. Postma, and J. van den Herik. Dimensionality reduction: A comparative review. *J. of Machine Learning Research*, 10:66–71, 2009.

[361] F. Van Ham, M. Wattenberg, and F. B. Viégas. Mapping text with phrase nets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 2009. doi: 10.1109/TVCG.2009.165

[362] P. Varma and C. Ré. Snuba: Automating weak supervision to label training data. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, vol. 12, p. 223. NIH Public Access, 2018.

[363] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[364] F. B. Viégas, M. Wattenberg, and J. Feinberg. Participatory visualization with wordle. *IEEE transactions on visualization and computer graphics*, 15(6):1137–1144, 2009.

[365] w3.org. Sparql 1.1 query language. https://www.w3.org/TR/sparql11-query/, March 2013. Accessed June 9, 2019.

[366] M. Wagner, D. Slijepcevic, B. Horsak, A. Rind, M. Zeppelzauer, and W. Aigner. Kavagait: Knowledge-assisted visual analytics for clinical gait analysis. *IEEE transactions on visualization and computer graphics*, 25(3):1528–1542, 2018.

[367] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355. Association for Computational Linguistics, Brussels, Belgium, 2018. doi: 10.18653/v1/W18-5446

[368] F. Wang, B. Hansen, R. Simmons, and R. Maciejewski. Name profiler toolkit. *IEEE Computer Graphics and Applications*, 37(5):61–71, 2017.

[369] J. Wang, K. Zhao, D. Deng, A. Cao, X. Xie, Z. Zhou, H. Zhang, and Y. Wu. Tac-simur: Tactic-based simulative visual analytics of table tennis. *IEEE transactions on visualization and computer graphics*, 26(1):407–417, 2019.

[370] Q. Wang, Z. Li, S. Fu, W. Cui, and H. Qu. Narvis: Authoring narrative slideshows for introducing data visualization designs. *IEEE transactions on visualization and computer graphics*, 25(1):779–788, 2018.

[371] Q. Wang, J. Yuan, S. Chen, H. Su, H. Qu, and S. Liu. Visual genealogy of deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 26(11):3340–3352, 2020. doi: 10.1109/TVCG.2019.2921323

[372] W. Wang, H. Wang, G. Dai, and H. Wang. Visualization of large hierarchical data by circle packing. In *Proceedings of the ACM CHI conference on Human Factors in computing systems*, pp. 517–520, 2006.

[373] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. Datashot: Automatic generation of fact sheets from tabular data. *IEEE transactions on visualization and computer graphics*, 26(1):895–905, 2019.

[374] Y. Wang, H. Zhang, H. Huang, X. Chen, Q. Yin, Z. Hou, D. Zhang, Q. Luo, and H. Qu. Infonice: Easy creation of information graphics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2018.

[375] M. Wattenberg and F. B. Viégas. The Word Tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1221–1228, 2008.

[376] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. Tiara: a visual exploratory text analytic system. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pp. 153–162, 2010.

[377] Wikipedia contributors. Tf–idf — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Tf2022. [misc; accessed 28-March-2022].

[378] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.

[379] A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics, New Orleans, Louisiana, 2018.

[380] F. Windhager, P. Federico, G. Schreder, K. Glinka, M. Dörk, S. Miksch, and E. Mayr. Visualization of cultural heritage collection data: State of the art and future challenges. *IEEE transactions on visualization and computer graphics*, 25(6):2311–2330, 2018.

[381] P. Witherell, S. Krishnamurty, and I. R. Grosse. Ontologies for supporting engineering design optimization. *Journal of Computing and Information Science in Engineering*, 7(2):141–150, 2007.

[382] J. Wu, L. Ouyang, D. M. Ziegler, N. Stiennon, R. Lowe, J. Leike, and P. Christiano. Recursively summarizing books with human feedback. *CoRR*, 2109.10862, 2021. doi: 10.48550/arXiv.2109.10862

[383] T. Wu, M. T. Ribeiro, J. Heer, and D. Weld. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[384] Y. Wu, Z. Chen, G. Sun, X. Xie, N. Cao, S. Liu, and W. Cui. Streamexplorer: A multi-stage system for visually exploring events in social streams. *IEEE transactions on visualization and computer graphics*, 24(10):2758–2772, 2018.

[385] J. Xia, W. Chen, Y. Hou, W. Hu, X. Huang, and D. S. Ebertk. DimScanner: A relation-based visual exploration approach towards data dimension inspection. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pp. 81–90, 2016.

[386] S. Xiang, X. Ye, J. Xia, J. Wu, Y. Chen, and S. Liu. Interactive correction of mislabeled training data. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 57–68. IEEE, 2019.

[387] C. Xiong, L. Van Weelden, and S. Franconeri. The curse of knowledge in visual data communication. *IEEE Transactions on Visualization and Computer Graphics*, 26(10):3051–3062, 2020. doi: 10.1109/TVCG.2019.2917689

[388] K. Xu, M. Xia, X. Mu, Y. Wang, and N. Cao. Ensemblelens: Ensemble-based visual exploration of anomaly detection algorithms with multidimensional data. *IEEE transactions on visualization and computer graphics*, 25(1):109–119, 2018.

[389] P. Xu, H. Mei, L. Ren, and W. Chen. ViDX: Visual diagnostics of assembly line performance in smart factories. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):291–300, 2016. doi: 10.1109/tvcg.2016.2598664

[390] S. Xu, C. Bryan, J. K. Li, J. Zhao, and K.-L. Ma. Chart constellations: Effective chart summarization for collaborative and multi-user analyses. In *Computer Graphics Forum*, vol. 37, pp. 75–86. Wiley misc Library, 2018.

[391] M. Yang, Q. Qu, Y. Shen, Q. Liu, W. Zhao, and J. Zhu. Aspect and sentiment aware abstractive review summarization. In *Proceedings of the International Conference on Computational Linguistics*, pp. 1110–1120. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018.

[392] J. S. Yi, Y. ah Kang, J. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics*, 13(6):1224–1231, 2007.

[393] S. Young. Infographic: Technology is leading the health care revolution. https://healthcaremba.gwu.edu/blog/technology-is-leading-a-healthcare-revolution, 2014.

[394] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[395] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

[396] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE transactions on visualization and computer graphics*, 25(1):364–373, 2018.

[397] J. Zhang, Y. Zhao, M. Saleh, and P. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pp. 11328–11339. PMLR, misc, 2020.

[398] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. BERTScore: evaluating text generation with BERT. In *Proceedings of the International Conference on Learning Representations*. Openreview.net, misc, 2020.

[399] Y. Zhang, J. Baldridge, and L. He. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1298–1308. Association for Computational Linguistics, Minneapolis, Minnesota, 2019. doi: 10.18653/v1/N19-1131

[400] Y. Zhang, Y. Wang, H. Zhang, B. Zhu, S. Chen, and D. Zhang. Onelabeler: A flexible system for building data labeling tools. In *CHI Conference on Human Factors in Computing Systems*, pp. 1–22, 2022.

[401] Z. Zhang, K. T. McDonnell, E. Zadok, and K. Mueller. Visual correlation analysis of numerical and categorical data on the correlation map. *IEEE transactions on visualization and computer graphics*, 21(2):289–303, 2014.

[402] J. Zhao, M. Glueck, P. Isenberg, F. Chevalier, and A. Khan. Supporting handoff in asynchronous collaborative sensemaking using knowledge-transfer graphs. *IEEE transactions on visualization and computer graphics*, 24(1):340–350, 2017.

[403] C. Zheng, D. Wang, A. Y. Wang, and X. Ma. Telling stories from computational notebooks: Ai-assisted presentation slides creation for presenting data science work. In *CHI Conference on Human Factors in Computing Systems*, pp. 1–20, 2022.

[404] Z. Zheng, Z. Lan, B. H. Park, and A. Geist. System log pre-processing to improve failure prediction. In *IEEE/IFIP International Conference on Dependable Systems & Networks*, pp. 572–577, 2009.

[405] C. Zhou, X. Ma, P. Michel, and G. Neubig. Examining and combating spurious features under distribution shift. In *International Conference on Machine Learning*, pp. 12857–12867. PMLR, 2021.

[406] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. Change Loy. Domain generalization: A survey. *arXiv e-prints*, pp. arXiv–2103, 2021.

[407] M. X. Zhou and S. K. Feiner. Data characterization for automatically visualizing heterogeneous information. In *Proc. InfoVis*, pp. 13–20, 1996. doi: 10.1109/infvis.1996.559211