MODELING, STRUCTURING, STORING, QUERYING AND APPRAISING THE PROCESS OF USER KNOWLEDGE GENERATION IN VISUAL ANALYTICS

by

Leonardo Christino

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
January 2024

Gloria in excelsis Deo.

Table of Contents

List of Tables

List of Figures

Abstract

Visual Analytics (VA) enable users to gain new knowledge through an iterative process of visualizing and interacting with data. VA's intrinsic complexity and flexibility can be used with many different goals in mind, such as data exploration or explainable AI. However, the same complexity and flexibility also impact user experience and evaluation. Domain-specific tools and complex visualizations are commonplace in VA, but they limit the number of potential users. This dissertation explores the concept of *Visual Analytics Democratization* wherein I seek to semi-automate the *provenance of knowledge* of VA workflows and model the exchange of knowledge between user and data as a *knowledge graph*. Such a graph can be used as a relationship database of visual insights and their underlying knowledge. It can also be used as a provenance database to relate all insights reached when using a VA tool to each other and the various steps taken for its acquisition. The proposed modeling process allows users to view and analyze the knowledge gained by past users. By linking the accumulated knowledge of "knowledge generators," which include other users, AIs like ChatGPT, or knowledge bases like Wikipedia, the proposed method opens a path for democratizing the results of analysis sessions to a broader, including non-technical, audience.

List of Abbreviations Used

AI Artificial Intelligence. 2, 3, 7–9, 12, 14, 20, 167–169, 171–174

EDA Exploratory Data Analysis. 21, 23

IA Intelligent Agent. viii, ix, 7, 10–12, 20, 74–80, 82–84, 86–91, 94–96, 98–100, 102–104, 166–180

KD Knowledge Decks. viii, ix, 10, 11, 137, 138, 140–165, 176, 177, 179, 180

KG Knowledge Graph. 8–13, 26, 27, 77, 78, 81, 82, 84–86, 94–96, 98, 103, 104, 175, 176, 178–180

LLM Large Language Model. 7, 9, 12, 20, 171, 172

ML Machine Learning. 7, 8, 25, 26, 178

NLP Natural Language Processing. 33, 43, 48, 49, 71

Q4EDA QuEry for visual Data Analysis. vi, vii, 9–12, 30–50, 52, 53, 55–57, 59, 62, 64–74, 105, 175–180

SE Search Engine. vi, 19, 27, 30, 31, 34–40, 42, 46, 47, 59, 70, 71, 175

SQ Search Query. 19, 30, 31, 34–40, 42, 45–47

VAKG Visual Analytics Knowledge Graph. viii, 9–12, 105–119, 121–137, 175, 176, 178–180

VA Visual Analytics. viii, 2–5, 8, 15, 38, 74–79, 81, 82, 88, 104–114, 116–122, 124–128, 132, 133, 135, 136, 177

VQ Visual Selection Query. vii, 31–33, 36–41, 46–51, 72

Vi&VA Visualization & Visual Analytics. vii, xii, 2, 4–27, 111, 137–145, 149, 150, 153, 159, 163, 164, 167, 175–179

XAI Explainable AI. 18

Glossary

behavior provenance a type of provenance in regards to the behavior of humans when performing actions, which includes the validity and origin of events by means of an information source and dependency. 6, 22, 23, 25, 27, 107–110, 113, 114, 116, 133, 134, 136, 176, 179

explicit knowledge a concrete type of knowledge which has been or can be written, saved, communicated and, consequently, processed by the computer [72]. 15, 20, 21, 26, 74, 76–78, 80–84, 86, 88–91, 94, 95, 102, 104, 176

knowledge provenance a type of provenance of the validity and origin of information/knowledge by means of modeling and maintaining information source and dependency [78]. 6, 7, 13, 23, 27, 107–110, 114, 116, 134, 136, 176, 179

knowledge (or knowledge learned) is a justified belief [193] in the reality or truthfulness of a piece of information, which may be by definition or acquired by study or investigation. 8, 173

ontology define domain concepts and the relationships between them, and thus provide a domain language that is meaningful to both humans and machines [202]. 16, 18, 25, 26, 82, 84, 102, 107–112, 115, 116, 118, 121, 122, 127, 133

provenance tracking and using data collected from a process, such as a *Vi&VA* tool being used by a user [177]. 8, 9, 12, 13, 22, 24–27, 175, 176, 179

state space the set containing all possible configurations of a system or tool. 24, 25, 106, 108, 113–115, 117, 118, 121, 123, 126, 133

storyline the process taken and results obtained by users during user-guided knowledge-gathering sessions within a *Vi&VA* tool. 4–12, 21, 23–26, 138–146, 149, 151–156, 160, 164, 175–177, 179, 180

tacit knowledge an abstract type of knowledge acquired through personal experience which users hold in their minds. It can only be acquired by humans through their cognitive processes [72]. 15, 20–22

Acknowledgements

To God first and foremost as who aided and was present throughout all. Second, to my family, especially my wife and precious friend Ana Elisa, who not just accepted the challenge of moving several times but also made my every day. Third, to Prof. Fernando Paulovich and his family who, despite many hurdles, supervised me admirably until the end. And also my extended friends/family in Bayers Road Baptist Church who were always there as true friends. I thank you all from the bottom of my heart. I also thank the institutions which funded my studies and research: Dalhousie University, Engage Nova Scotia, and Mitacs.

Chapter 1

Introduction

IT is undeniable that the past decades represent a revolution without precedent in human history for information and knowledge dissemination. The amount of data available and accessible is increasing at a pace never seen before, with massive efforts involving companies, non-profit organizations, governments, and others to support its democratization. The concept of giving access to everyone, everywhere, anytime, unthinkable in the past, is becoming the norm. Initiatives like GapMinder [186] to help educate people to confront information with data have shown how wrong their premises about the world are, biased by outdated information that does not reflect the current reality [190]. It is a paradigm shift in how information is handled, even when official and (auto-declared) credible sources release it.

Through this democratization effort, we now have access to a wide array of datasets for data analysis. However, despite all these sources of information, the use of such data is handled by a team of specialized experts who have the knowledge of data visualization, knowledge of data analysis, knowledge of the data's domain, and the technical expertise for developing tools for data exploration and analysis [113]. Therefore, even though data democratization is one of the most critical advances in our free world, in the context of a non-expert user, this initiative is only relevant when hypotheses are known or when a domain expert is present to explain findings and insights throughout the data which limits the data analysis when such assumptions do not hold [197, 105].

This conflict is heightened in exploratory scenarios due to a lack of concrete questions or facts to check on the users' part. The lack of a domain expert to help analyze the data hampers the intrinsic value of data, lessening the advantages of its democratization. For example, GapMinder [186] is a great tool for exploring demographic data. Though its users may find unexpected patterns in the data,

the tool itself does not explain why such patterns exist, requiring the user to seek external aid from a domain expert, a search engine, an *AI* model like ChatGPT [169], or some other form of *explicit knowledge* external to GapMinder. This severely hampers users', especially lay users', ability to gain new knowledge during the analysis and, consequently, in sharing their findings.

This idea of reducing the need for the manual search for external knowledge, such as through domain experts, is a recent trend in the visualization community with the proliferation of approaches to facilitate and automate parts of the visualization and storytelling process [145, 140, 31], including automatic infographics creation [59] and pattern identification [219, 66]. Although this represents a step towards supporting unconstrained access to data analysis, it still relies on manual processes, such as the intervention of domain experts during the analysis. This limits the extent of how much a user can understand and use an existing tool on their own to harness new knowledge. This issue can be traced back to how knowledge itself is modeled and how the transfer of knowledge between the user and the tool is defined.

1.1 Visual Analytic Democratization

Taking a step back, Visualization & Visual Analytics (*Vi&VA*) frameworks and tools have had an immense impact in aiding users in interactively exploring and analyzing data through visual means. Visual Analytics (*VA*), sometimes called Visual Data Mining, is an area of Computer Science that attempts to expose the data processes, such as data mining and data science, and the information within as interactive visualizations [205]. Simultaneously, visualizations used in *VA* allow users to interact with the data and the data mining processes utilized within, as seen in Figure 1.1. This iterative process of visualizing and interacting with the data and processes provides new findings, insights, and knowledge to the user [193]. Visualizations within *VA* may display raw data but also results from machine learning or information retrieval algorithms. *VA* also allows users to interact with the visualization to modify the algorithm parameters. As Keim et al. [123] says: "The core of our view on Visual Analytics is the new enabling and accessible analytic reasoning interactions supported by the combination of automated and

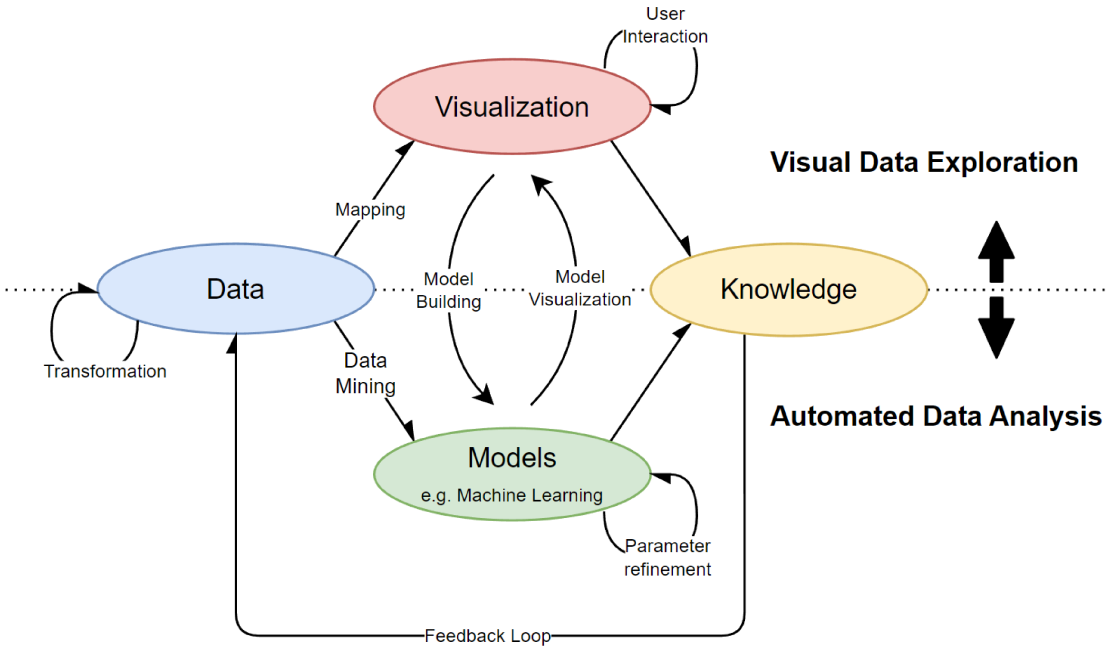


Figure 1.1: Visual Representation of the Visual Analytics loop [231].

visual analysis.".

VA attempts to handle the use of visualizations and interactivity among the vastness of data mining and data science. Consequently, *VA* is considered a complex field with many moving parts. For instance, *VA* tools require visual design, user interface design, data sourcing, storage, pre-processing, mining, service architecture, data security, and many other elements in a single orchestrated system or tool [232]. *VA* is also flexible, allowing it to be used with many different goals in mind, such as data exploration [123] or explainable *AI* [163]. This complexity also impacts user evaluation [237], especially seeing that contributions within the *VA* literature tend to either be a domain-specific tool with complex visualizations or be theoretical in nature. In other words, users are generally required to be knowledgeable about the domain and the machine learning or data mining techniques used within a tool prior to using it to be able to effectively use the tool's visualizations. This motivated me to align this dissertation to a concept I call "Visual Analytic Democratization", from where I derive my research theme:

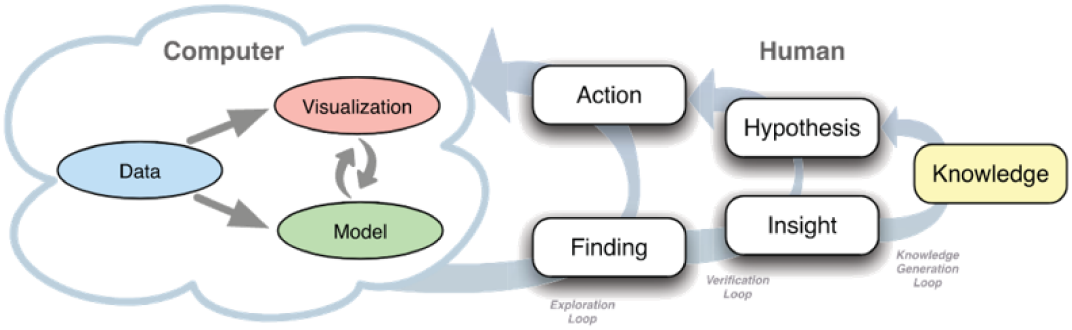


Figure 1.2: Knowledge Generation Model for Visual Analytics. The model describes the interaction between *human* and *computer* to generate knowledge from data. [193].

> How can the search for knowledge in a *VA* tool be simplified or automated so that it can be used and further democratized, that is, developed and used more broadly?

1.2 Motivation

In general, not only *VA* tools but Visualization & Visual Analytics (*Vi&VA*) tools provide ways for users to harness insights and knowledge from datasets [193]. This process of harnessing new knowledge as defined by Sacha et al. [193] and shown in Figure 1.2 involves providing users with interfaces where they can uncover findings during visual exploration, verifying these findings to generate insights and hypothesis, and finally, users can test these insights against the contents of the *Vi&VA* tool. During this process, the user acquires new *Knowledge*. In practice, users can use *Vi&VA* tools to answer an existing question, execute pre-planned tasks, or explore the information contained in the available visualizations [193]. The insights gathered from such use cases allow one to harness knowledge from data and then act upon the newly acquired information [72]. This process of user-guided knowledge discovery, which we define as a user *Storyline*, is extensively studied within *Vi&VA*.

The *Vi&VA* knowledge generation models [193] are currently used as the theoretical foundation of modeling user *storyline*s. *Vi&VA* developers can use these models to better understand their users and refine their tools. Additionally, *Vi&VA*

researchers proposed mathematical frameworks that describe the workflow of knowledge gathering as an iterative process between user and machine [72]. Specific taxonomies of conceptual structures [44] are also used to represent user behavior [6, 241] within the *Vi&VA* model. The practical application of these models and taxonomies have led *Vi&VA* to attempt collecting users' behavior [152, 98] and, by analyzing it [107, 20], we can depict a picture of the *Vi&VA* tool users' profile, the tools' performance in providing new knowledge to users, and the recall and storytelling of *storyline*s [141, 171] becomes possible.

Although *Vi&VA* tools provide users with the ability to seek and discover new knowledge from data, each user's *storyline*s is typically unique and non-linear. For instance, users may have had the same intention when opening and using a *Vi&VA* tool, but due to their own past experiences, some users may focus on one specific part of the tool at first, while others may instead want to understand the tool in a generalized way. Users may also uncover different insights even when looking at the exact same visualization due to differences in their own goals and questions or due to some personal pre-conceived bias [26]. Likewise, even when the same insight is found, users may have used different ways to arrive at it [107]. When collecting user behavior for analysis, these inconsistencies generally cause significant manual labor of transforming the collected user data, such as video/audio/logs recordings of surveys [250] into a cohesive data structure. The work involved in collecting and transforming such data is not just time consuming [171, 200], but also the process used in each work is often not reproducible [250, 72]. However, with the aim of advancing the democratization of Visual Analytics, would a better understanding of how experts can successfully extract knowledge from *Vi&VA* aid in the endeavor? And would sharing this better understanding aid in moving *Vi&VA* towards democratization of *VA*? In other words:

> Inconsistencies between different users' behavior and insights due to pre-conceived bias or personal preferences and a lack of a general technique to automatically relate users under these conditions are limitations found during the research of this dissertation that hinder the collection and sharing of expert users' *Storyline*s.

To collect and analyze user behavior and interactions during their knowledge-gathering sessions, researchers use a process called provenance [177, 20, 251]. There are three steps in this provenance process: *user tracking*, which collects data; *structuring*, which organizes the data for use in data mining; and *appraising*, which analyses the data to extract results such as statistical values, new visualizations, suggestions of next steps and auto-generated text summaries. Examples of user tracking can be seen when collecting changes in datasets [62], updates in visualizations [20, 251], requesting feedback from users [21, 152] or by collecting user annotations [208]. Data mining and data science processes can then structure such data for appraising by, for instance, creating databases with said data [204].

Examples of the appraising step are often found in the user evaluation section in much of the *Vi&VA* literature, but another relevant example is the recall of the information found by the user and the *storyline* of how this information was acquired [250]. By extracting the linear sequence of events or insights as temporal snapshots from provenance, one can provide a step-by-step retelling of the users *storyline*, which can then be displayed as info-graphs [128, 264], slide decks [171, 200], or included in visualizations and tools as what is called *explicit knowledge* [251]. Although news, blogs, and other means of mass communications have successfully used info-graphs and static visualizations to summarize a specific piece of knowledge extracted from such process, a broader application of provenance to automatically or semi-automatically transform user *storyline*s and insights into shareable stories has not been done. For instance, if a third party (e.g., the user's manager) requires a user to perform some data analysis and, from its results, share the findings with others, the usually expected process is for the user to manually perform data analysis with tools like Microsoft Excel [155], then manually generate visualizations of the findings and present them. This is a sub-type of provenance called *Knowledge provenance*. In the example above, users keep track of their analysis and insights and then summarise their knowledge-generation process into a presentation.

Although *Vi&VA* has invested in ways to capture the user behavior within provenance (*Behavior provenance*), the concept of capturing the knowledge-discovery process itself as *storyline*s (*Knowledge provenance*) has yet to be applied in practice in

Vi&VA. In other words, there is no widely used mechanism in *Vi&VA* that tracks the user's behavior and their gained knowledge during this visual querying and knowledge retrieval process. Instead, current *Knowledge provenance* techniques collect users' *storyline*s in a completely manual fashion. Though they have proven valuable for the user to recall users' past experiences [141], only application-specific or domain-specific methods have been proposed. Such *knowledge provenance* would automatically *collect* how the user behaved, what the user discovered, and what the exact information shown to the user at any point in time, provide a *structure* for utilization of said information and generate a presentation to disclose or *appraise* knowledge discovery to others [250]. The main novelty in the provenance technique that will be discussed in this dissertation is in automating many of the steps that are currently done manually throughout *Vi&VA*.

The concept of automation is a recent trend not just in *Vi&VA* community, but also in other Machine Learning (*ML*) fields. AutoML methods [102, 101, 17, 167, 76], for instance, automates parts of the process of *ML* development by exhaustive procedures. Artificial Intelligence (*AI*) research assistants powered by Large Language Models (*LLM*s) have also shown significant potential to revolutionize the way scientific work is performed. The potential impacts of *AI*-based tools on scientific practices are anticipated and feared by researchers [235] due to the amount of automation Intelligent Agent (*IA*) such as ChatGPT [169] may bring. In fact, the topic of Intelligent Agent (*IA*) can already search for knowledge sources, summarize them into bullet points, write a story with convincing arguments about the contents, and generate visualizations to express the results. Only a few years ago, this possibility was considered too far into the future to be worth discussing; today, we see an increasing number of writing materials, from blog posts to entire books, being partially or fully automated. Intelligent Agent (*IA*) using *LLM AI*s may soon incorporate established strategies into *Vi&VA* provenance methods, particularly to cater to the interface between humans and machines.

A similar trend is also observed in the visualization community with the proliferation of approaches to facilitate and automate parts of the visualization and storytelling process [145, 140, 31], including automatic infographics creation [59]

and pattern identification [219, 66]. Although this represents a step towards supporting unconstrained access to data analysis, *VA* has yet to use automation in the provenance of its knowledge discovery process. Similar to how AutoML has been proposed to increase the widespread usage of *AI* and *ML* [178], automation of knowledge-related processes in *Vi&VA*, such as the collection and analysis of how users acquire new knowledge, of *Vi&VA* would also increase the use of *Vi&VA* itself, providing a new degree of *Vi&VA* democratization throughout computer science and beyond.

With this, I can present this dissertation's research question:

> *How to use Provenance to automate the Visualization & Visual Analytics' knowledge gathering process in order to model, structure, store, query, and share the users' storylines? And does this bring advantages back to the users and, consequently, to the idea of Visual Analytic Democratization?*

1.3 Objectives

To answer this research question, the objectives of this dissertation are:

O_1: To review the literature on the current state of *knowledge* modeling in *Vi&VA* and its usage as part of research and development of *Vi&VA* tools;

O_2: To discuss how knowledge automation in *Vi&VA* can help in democratizing visual analytics by providing ways for users to *request* and *provide* new pieces of knowledge to *Vi&VA* tools, including exemplifying manual and automatic means of requesting or providing knowledge through visual queries and *AI*;

O_3: To explore the usage of Knowledge Graphs (*KG*s) as a means to store knowledge of user *storyline*s and define how *KG*s can connect the theoretical frameworks of knowledge modeling and *provenance* to practical uses of the recorded user *storyline*s.

In this dissertation, the research question involves exploring how *Provenance* can help democratize *Vi&VA*.

1.4 Thesis Contributions

In order to achieve these objectives, I first investigate the current knowledge modeling and means to structure this knowledge to depict a clear picture of the current state-of-the-art. The content of chapter 2 describes the current state of the literature on how one collects, structures, and analyzes knowledge (O_1). Then, I investigate the process of how users search for knowledge. I propose a framework called QuEry for visual Data Analysis (*Q4EDA*) (chapter 3) and a *Vi&VA* tool (LINKED) wherein users can request relevant information from Wikipedia [247] given a *visual finding*. This tool shows the first example of a user *storyline* where users search for knowledge based on a question. After discussing how a Knowledge Graph (*KG*) [40, 82] can be used to structure knowledge, I then propose an approach called ChatKG (chapter 4) where *Vi&VA* tools can use an *LLM AI* assistants to perform the search for knowledge automatically, structure the knowledge within a *KG*, and display to the user. This way, the *Vi&VA* tool itself is simplified and becomes more approachable to users (O_2) by using *KG* as the intermediary knowledge repository (O_3).

I developed several *Vi&VA* tools that aided me to better understand and analyze how users can search for and gain new knowledge while performing visual data analysis and visual data exploration tasks, of which five will be mentioned in this dissertation. Three of the tools were built to verify how linked visualizations provide a way for exploratory visual analysis (EVA) [263] to provide new tacit knowledge to the user. The fourth tool, which was already mentioned, is called LINKED and relates to aiding in information search from *visual selections*. The final tool implements a set of libraries that can be attached to any of the previous tools to perform Knowledge Provenance, structure it as a KG, and, with it, generate slide decks from the aggregated user knowledge. These tools and methods allowed me to better conceptualize what knowledge in the context of *Vi&VA* is.

Using the foothold of *Q4EDA* and *ChatKG*, I present a formal model of knowledge called Visual Analytics Knowledge Graph (*VAKG*) (chapter 5) specifically designed to aid in the automatic structuring of *storyline*s out of *provenance* methods. In other words, just as I modeled the user-guided process of *Q4EDA* into a *KG* in ChatKG, I generalize the user-guided process described in *Q4EDA* by following

the existing theoretical methods discussed in chapter 2 and model the result into a *KG* in *VAKG*. This model is then exemplified by being applied to Visualization & Visual Analytics (*Vi&VA*) tools, where I showcase how once the collected data is within a *KG*, it can be analyzed using various existing techniques. I apply the same process to real *Vi&VA* tools with Knowledge Decks (*KD*), where I demonstrate an approach to (semi)automatically collect and store user *storyline*s. The results of *KD* enable users to generate slide decks out of the interwoven user *storyline*s for sharing or presenting a summary of the linear process taken by the user in their search for knowledge or the listing of the collective knowledge generated by users.

The complete map of all contents of this dissertation is shown in Figure 1.3 and in the below list of publications which were made in the research of this dissertation:

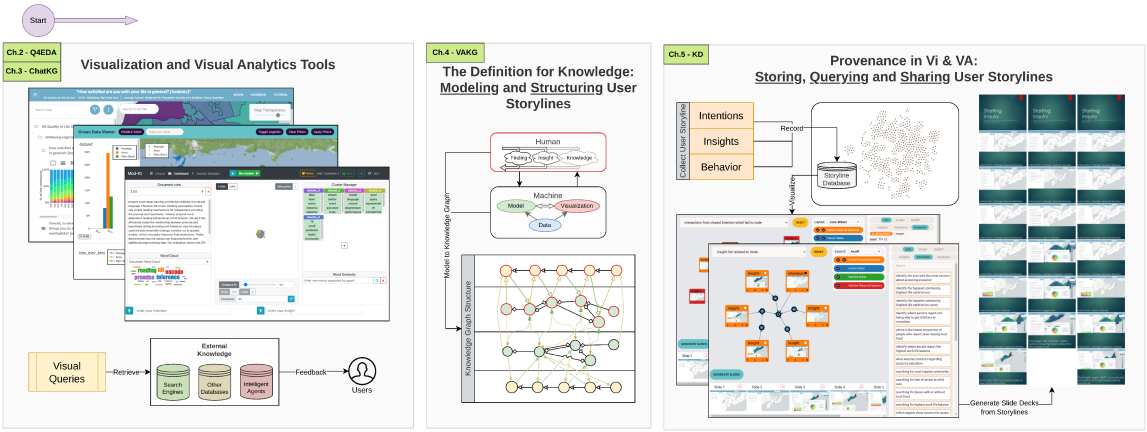


Figure 1.3: Towards Visual Analytic Democratization: the map of all contents discussed in the dissertation. First, I investigate how visual exploration and visual queries influence user *storyline* with *Q4EDA*. Then, I automate parts of the user's *storyline* with *ChatKG* though Intelligent Agents. I generalize the modeling process of user *storyline*s with *VAKG* and use *KG*s to structure user *storyline*s. I apply the *VAKG* process in *KD* by collecting user intentions, behavior, and insights. By populating a *KG* with user data, *KD* displays an interface for users to explore the intertwined *storyline*s of their tool and export it as slide decks.

Q4EDA: Christino, L., Ferreira, M. D., & Paulovich, F. V. (2022). Q4EDA: A novel strategy for textual information retrieval based on user interactions with visual representations of time series. Published in Information, 13(8), 368.

ChatKG: Christino, L., & Paulovich, F. V. (2023). ChatKG: Visualizing Temporal

Patterns as Knowledge Graph. Published in Eurographics Association, ISBN 978-3-03868-222-6. doi: 10.2312/eurova.20231090. Extension submitted to Computer & Graphics in November 2023.

VAKG: Christino, L., & Paulovich, F. V. (2023). From Data to Knowledge Graphs: A Multi-Layered Method to Model User's Visual Analytics Workflow for Analytical Purposes. arXiv preprint arXiv:2204.00585. Submitted to Computer Graphics Forum in October 2023.

KD: Christino, L., Hill, T., & Paulovich, F. (2022). Knowledge-Decks: Automatically Generating Presentation Slide Decks of Visual Analytics Knowledge Discovery Applications. arXiv preprint arXiv:2212.01469. Submitted to EuroVIS in November 2023.

The results of each of these contributions enabled me to investigate how *automation* and *provenance* impacted Visual Analytic Democratization. First, by surveying the literature on how knowledge is seen in *Vi&VA* (chapter 2), I identified the research directions of *automation* and *provenance* as potential ways to push *Vi&VA* towards Visual Analytic Democratization (O_1). With *Q4EDA* (Figure 1.3[2]), users showed that *Vi&VA* is more useful and provides more consistent insights when it is not just able to show data, but also able to automatically provide relevant knowledge of user-selections (O_2). With ChatKG (Figure 1.3[3]), the additional automation of detecting "sub-data of interest", retrieving contextualized knowledge from Intelligent Agents (*IA*s), and displaying to users data, detected sub-data of interest and related knowledge simultaneously provided further ease-of-use to users compared to Q4EDA (O_2). Users said that ChatKG reduced the training necessary while at the same time providing the value of a contextualized analysis only possible because of *IA*s. In order to investigate *provenance*, I proposed *VAKG* (Figure 1.3[4]), which was shown to be a valuable method to define the user *storyline*s as a Knowledge Graph (*KG*) which can be automatically captured and saved. The results of *VAKG* are then used in *KD* (Figure 1.3[5]) for the exploration of *storyline*s and the generation of slide decks. By allowing experts to use these methods, they were able to better understand their own tools, explore what and how their users learned while using a *Vi&VA* tool, and share the findings with colleagues

(O_3). Finally, I expanded the investigation to recent advances in Intelligent Agent (*IA*)s, which is shown to successfully process and efficiently convey information to the general population, leading me to conclude that Intelligent Agent (*IA*)a have immense potential as the next research direction to further democratize *Vi&VA*.

In conclusion, this dissertation shows how automation in the detection of patterns of interest and in the retrieval of contextualized external knowledge has great potential to aid in democratizing *Vi&VA*. Similarly, I show that *provenance* of user *storyline*s provides users ways to harness the knowledge from other users through slide decks.

1.5 Dissertation Structure

The dissertation is organized as follows: In chapter 2, I review previous works and surveys knowledge modeling and its ramifications within *Vi&VA*; In chapter 3, I describe the implementation and study of *Q4EDA*, a *Vi&VA* framework that generates search queries to retrieve relevant information from visual selections; In chapter 4, I describe the implementation and study of ChatKG, a novel visualization strategy that visualizes the structure of a Knowledge Graph which relates extracted knowledge from ChatGPT [169] and automatically detected patterns within a temporal dataset; In chapter 5, I generalize the modeling process done in ChatKG by using existing knowledge modeling literature to provide a theoretical framework called *VAKG*, defines how to perform *provenance* of user *storyline*s as a set of linked *KG*s; In chapter 6, I demonstrate VAKG being used in practice as to model, structure, store, query and appraise the users' knowledge generation process; In chapter 7, I discuss limitations of my findings when considering how recent advances in *AI*, specially in *LLM*s, can and already are impacting the development of *Vi&VA* tools and scientific research. In chapter 8, I conclude my dissertation by discussing whether Visual Analytics Democratization is realistically attainable with the currently available work or if further work is required.

Chapter 2

Background and Related Work

In this chapter, I describe and discuss related work relevant to the concept of Visual Analytic Democratization being investigated in this dissertation. For this, I discuss the theory of how knowledge is modeled in Visualization & Visual Analytics (*Vi&VA*) tools, what is *Provenance*, how provenance, especially a type of provenance called *Knowledge provenance* is applied in the *Vi&VA* context. Finally, I discuss the relationship between *knowledge provenance* as a data collection methodology and the concept of Knowledge Ontology, a knowledge structuring methodology, and *Knowledge Graph (KG)s* as a method to store and query knowledge.

2.1 Visual Analytic Democratization, an Introduction

It is undeniable that the past decades represent a revolution without precedent in human history for information and knowledge dissemination. The amount of data available and accessible is increasing at a pace never seen before, with massive efforts involving companies, non-profit organizations, governments, and others to support its democratization. The concept of giving access to everyone, everywhere, anytime, unthinkable in the past, is becoming the norm. Initiatives like Gapminder [185] to help educate people to confront information with data have shown how wrong their premises about the world are, biased by outdated information that does not reflect the current reality [190]. It is a paradigm shift in how information is handled, even when official and (auto-declared) credible sources release it.

Through this democratization effort, we now have access to a wide array of datasets for data analysis. However, despite all these sources of information, the use of such data within *Vi&VA* is handled by a team of specialized experts who have the knowledge of data visualization, knowledge of data analysis, knowledge

of the data's domain, and the technical expertise for developing a *Vi&VA* tool, which is then used by the user [113]. Therefore, even though data democratization is one of the most critical advances in our free world, in the context of a non-expert user, this initiative is only relevant when hypotheses are known or when a domain expert is present to explain findings and insights throughout the data which limits the data analysis when such assumptions do not hold [197, 105].

This conflict is heightened in exploratory scenarios due to a lack of concrete questions or facts to check on the users' part. The lack of a domain expert to help analyze the data hampers the intrinsic value of data, lessening the advantages of its democratization. For example, GapMinder [186] is a great tool for exploring demographic data. Though its users may find unexpected patterns in the data, the tool itself does not explain why such patterns exist, requiring the user to seek external aid from a domain expert, a search engine (e.g., Google or Wikipedia), an *AI* model like ChatGPT [169], or some other form of *explicit knowledge* external to GapMinder. This severely hampers users', especially lay users', ability to gain new knowledge during the analysis and, consequently, in sharing their findings.

This idea of reducing the need for the manual search for external knowledge, such as through domain experts, is a recent trend in the visualization community with the proliferation of approaches to facilitate and automate parts of the visualization and storytelling process [145, 140, 31], including automatic info-graphs creation [59] and pattern identification [219, 66]. Although this represents a step towards supporting unconstrained access to data analysis, it still relies on manual processes, such as the intervention of domain experts during the analysis. This limits the extent of how much a user can understand and use a *Vi&VA* tool on their own to harness new knowledge. This issue can be traced back to how knowledge itself is modeled and how the transfer of knowledge between the user and the *Vi&VA* tool is defined. Let us take a step back and discuss how the *Vi&VA* literature has modeled Knowledge and how its automation is key to Visual Analytic Democratization.

2.2 Knowledge In Visualization and Visual Analytics

2.2.1 Knowledge Modeling

Vi&VA at its core is characterized by the iterative loop of users' insight harnessing, knowledge generation, and interaction with visualizations [193]. This loop, which will henceforth be called *knowledge gathering process*, shows that there are two sides to the *Vi&VA* equation: HUMAN and MACHINE, and each side uses the information within to influence the other throughout the knowledge gathering process. For instance, users may wish to gain some new knowledge, which causes them to interact with a visualization. This way, the *explicit knowledge* within the MACHINE side becomes *tacit knowledge* to the HUMAN through a visualization [72]. That is, *Vi&VA* handles two types of knowledge: **tacit** and **explicit** knowledge. As Federico et al. [72] says: "Tacit knowledge can be understood as knowledge which users hold in their minds, it is personal and specialized, and it can only be acquired by humans through their cognitive processes; explicit knowledge has been written, saved, or communicated and, therefore, can be stored in a database and processed by a computer". Therefore, to understand knowledge in *Vi&VA*, one must encompass both sides of *Vi&VA*: human and computer.

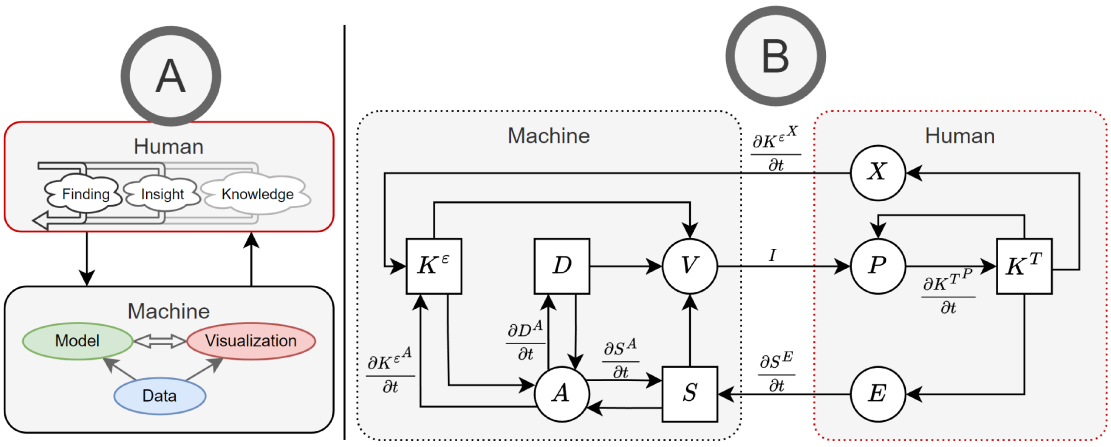


Figure 2.1: *Vi&VA* Knowledge Model. In [A] is shown a representation of *VA* as defined by Sacha et al. [193], while [B] shows the Knowledge-Assisted workflow of *VA* as defined by Federico et al. [72]. A Visualization model is the same as above but without "Model" in [A] and without any automated analysis process A in [B].

A representation of the relationship between the two knowledge types can

be seen in Figure 2.1. Following the definitions of Federico et al. [72], circles represent processes, and boxes represent containers of data that are continuously accumulated and accessed. The nodes definitions are: visualization V, perception and cognition P, exploration E, data D, and specification S, and tacit knowledge K^T. To capture the role of explicit knowledge, Federico et al. [72] incorporated two additional elements compared to prior work: a container accounting for the existence of explicit knowledge itself, K^ϵ and a process that accounts for automatic analysis methods A.

2.2.2 Knowledge Theory Concepts

Typically, researchers prefer to define their workflow as descriptively as possible for particular use cases or by following certain well-tested processes. Theoretical research in the model design of the *Vi&VA* workflow, for instance, depicts this diversity very well. In this section, I describe the *Vi&VA* theoretical literature by following the definitions of Chen et al. [44]. The contribution of theoretical *Vi&VA* works can be categorized as one or more of the following:

Principles and Guidelines: Qualitative descriptions or rules that define a process that may lead to the desired outcome. Works that extract the qualitative elements of a *Vi&VA* workflow and define rules based on it are examples of such concepts [195, 28].

Taxonomy and Ontology: A collection of concepts that defines a structure. Such research usually focuses on novel theoretical *ontology* to structure the knowledge generation workflow [193, 241, 195, 45, 43, 175].

Conceptual models: Abstract representation of a real-world process by using a collection of theoretical taxonomies, typologies, and guidelines. For the purposes of this dissertation, a *Vi&VA knowledge model* is a model of a user's knowledge generation throughout a *Vi&VA* process. Arguably, the most prominent example of such a model is of [193]. Generally speaking, knowledge modeling defines a workflow where interactions with intent lead to knowledge generation [6]

Theoretic frameworks: Collection of operators which to measure a process (e.g., mathematical operators). The *theoretic system* defined by Federico et al. [72], for instance, is able to describe and measure the process of many existing *Vi&VA* systems and tools.

Quantitative laws: Describes causal relationships between conceptual models by means of a theoretic framework. As an example, Federico et al. [72] applies this concept when comparing multiple *Vi&VA* knowledge models.

Theoretic systems: An extension of a conceptual model that uses theoretical frameworks to define a real-world process formally. Federico et al. [72] extends several conceptual models in such a way as to formalize its methodology.

As part of *Vi&VA* research, many works have modeled tacit and explicit knowledge as an iterative process of generating visualizations from data, expecting the user to interact with visualizations and updating the visualizations through interactions. The knowledge generation model of Sacha et al. [193] has been seen as the theoretical foundation to understand this knowledge discovery process (Figure 2.1[A]). Some use this model to describe the theory behind knowledge, such as through mathematical frameworks [72] (Figure 2.1[B]) or other models [6], while others extend the model beyond theory and into the realm of conceptual structures [44, 195], frameworks or taxonomies [241].

2.2.3 Machine Knowledge Visual Analytics

Visualization & Visual Analytics tools use the theoretical taxonomy above as a means to describe and structure the process where users gather knowledge. We can see this by specifically investigating the MACHINE aspect of *Vi&VA* of Figure 2.1, where a diverse set of operations, such as machine learning, data mining, and other manual or automated processes [193, 72] are being modeled. As Keim et al. [123] says:"The core of our view on Visual Analytics is the new enabling and accessible analytic reasoning interactions supported by the combination of automated and visual analysis". However, each of these MACHINE aspects of *Vi&VA* requires a diverse set of experts in order to refine the raw data into what can be used in *Vi&VA* tools. That is, existing non-theoretical works, such as *Vi&VA* tools and

frameworks, tend to focus on a specific aspect instead of the broader understanding of knowledge. For instance, Vis4ML [195] is an *ontology* which aids in modeling the machine-learning aspect of *Vi&VA* but does not focus on other aspects. That is, although Vis4ML includes pre-processing and data mining as steps of machine learning, it does not tackle automatic processes or user feedback within their model, leaving some of the MACHINE aspects unattended.

Automation of Machine Learning

Among the MACHINE aspects of *Vi&VA*, automation (Figure 2.1[A]) is a recent trend in the community. AutoML methods, for instance, replace machine learning engineers with automatic and exhaustive procedures to build models [102, 101, 17, 167, 76]. Explainable AI (*XAI*) [2] is another trending concept that has been used to automate parts of the machine learning pipeline [11]. Similarly, automatic pattern identification [219, 66] attempts to extract insights from data automatically.

Automation of Visualization

As seen in Figure 2.1, the image [I] sent to the user at any given point comes from the Visualization [V], which is generated based on the schema [S] developed by that *Vi&VA* tool. Similar to AutoML and *XAI*, the automation of defining or deciding the schema [S], which itself is the design of the visualization used by the tool, has been another trending topic within *Vi&VA*. For instance, there is a recent proliferation of ways to facilitate and automate parts of the visualization. For instance, Deepeye [145] automatically generates visualizations from keywords and VizByWiki [140] from news articles. Another is the automatic generation of info-graphs by Text-to-Viz [59], which aims to use existing insights in the form of text to generate visualizations to reemphasize that insight in a visual format. The storytelling of processes, such as temporal summaries [31], has also been attempted. In general, many works have advanced the ways visualizations can be understood and generated, yet few have touched the broader context of knowledge as defined by the knowledge model theory. Instead, each work seems to be in its own goal-specific bubble, not expanding and investigating how automating the generation of visualizations impacts the user's knowledge discovery process.

Search Query/Engines: Automation of Information Retrieval

To better understand the existing knowledge within data, Information Retrieval is a broad area that attempts to extract "information" from data. Of course, one of the main ways to query for and extract data is by using database queries [80]. There has been significant research on ways to automate such data retrievals, such as through query auto-completion [254] and other similar techniques [89]. Within such research, significant effort has been made to propose better ways to use *Search Engines (SE)* [58] during data analysis for this purpose. Indeed, it is undeniable that advancements in *SE* represent a revolution without precedent in human history for knowledge dissemination. Due to the large amount of readily available data, knowing how to construct *Search Queries (SQs)* [95] has become a fundamental skill. From Google [142] to Wikipedia [224], most SEs require users to type a text-based search query to retrieve relevant information.

Although users are required to write these queries manually, the machine can only retrieve relevant information by using a large set of intelligent mechanisms. Of course, the retrieved information is dependent on what data it is fetched from. For instance, when fetching information from Wikipedia [224], one must expect the possibility of bias or wrong data due to the intrinsic nature of how Wikipedia works. Nonetheless, *SE* can be considered as a large automation process to convert queries into *explicit knowledge*. Of course, this approach has its limitations. Though much data is available through *SE*s, most of the structured datasets, such as the UNData [161] time-series world indicators or other datasets within Kaggle [220, 217], live outside the scope of a *SE*. On the other hand, the information contained in structured datasets is typically available to probe through visualization tools or interfaces. Additionally, there is a lack of attempts to automate the generation of queries, and since *Vi&VA* is normally focused on visualizations and visual metaphors, this reduces the potential applicability of *SE*s as part of *Vi&VA* tools.

Indeed, there is a gap in the literature when considering ways to query for explicit knowledge from visual interaction. During the research of this dissertation, no mechanism that uses pieces of visual data, such as user selections in a line chart, to search for related information in existing knowledge repositories, such as Wikipedia and other general-purpose *SE*s, was found. Instead, current solutions

either focus on the generation of visualizations given an insight [115, 15, 255] or focus on how to provide a visual interface for writing the query [130, 263, 26, 27]. Even though such visualizations are shown to aid the information-gathering processes, in these works, the use of visual selections as *visual search queries* to fetch relevant information was an unexplored concept. Also, various examples of search query suggestions for data analysis [168, 254] have been proposed, but the use of such techniques applied to *visual search queries* is similarly unexplored.

Intelligent Agents

New advances in *AI* through the use of Intelligent Agents has been another trending topic within *Vi&VA* [22, 68]. Though arguably this concept is not captured within the model of Figure 2.1, the scientific community already has proposed ways to capture the influence of an external Intelligent Agent (*IA*) within the *Vi&VA* model. The work by El-Assady and Moruzzi [68], for instance, argues that a *Vi&VA* tool can, itself, be a form of interface between an Intelligent Agent and a Human Agent. Yet, Large Language Model (*LLM*)s, such as ChatGPT [169], have already shown great impact when being used in cooperation with humans. From information retrieval [5], analysis [104], and explanations [215], Intelligent Agent are impacting many fields and will certainly be central for Visual Analytic Democratization.

2.2.4 User Knowledge-guided Visual Analytics

Similar to how *explicit knowledge* represents the knowledge contained within any of the MACHINE aspects, such as datasets, machine learning, and visualizations, *tacit knowledge* represents the information that is part of the user. Significant work has been done to demonstrate the breadth and depth of *tacit knowledge* within *Vi&VA*. The model of Sacha et al. [193] describes the *Vi&VA* process as a series of user interactions that lead to new findings. A collection of findings may be related in some way, providing the user with a new insight. Several insights associated with other information, such as external sources or experience, can coalesce into a new *tacit knowledge*.

The *theoretic system* proposed by Federico et al. [72] expands on the model of

Sacha et al. [193] by describing *tacit knowledge* as a dynamic knowledge that may exist prior to the user ever using the *Vi&VA* tool, but it can be modified by the user's perception (Figure 2.1[B.X]) of the visualizations generated by the tool. In short, Federico et al. [72] demonstrates and exemplifies how the subsequent interactions and *feedbacks* between the HUMAN and MACHINE are related. They also describe how automatic processes in data mining, visualizations, and machine learning can influence the user's *tacit knowledge*. Nevertheless, although the works listed and described by Federico et al. [72] may differ, *Vi&VA*'s purpose can be summarised as the creation of findings, insight, and *tacit knowledge* through an iterative workflow of visual interactions between the user and computer [72, 193, 43], or, in other words, a user *storyline*.

Exploratory Data Analysis

Among the several possible ways to perform such workflow, a notable one is Exploratory Data Analysis (*EDA*) [212], where extraction of knowledge from data has been at its core. *Vi&VA* tools like Gapminder [185] help educate people to confront pre-conceived knowledge with data, showing how wrong their premises are about the world due to bias or outdated information [190]. Tools such as these transform raw datasets into visual tools, allowing users to search for relevant information given their own desires. These datasets may come from closed sources, general-purpose repositories, such as Kaggle [220], domain-specific forums, and social media. However, the translation of *explicit knowledge* within these datasets into *tacit knowledge* through exploratory means has a significant gap between how the knowledge theory views the issue and how *Vi&VA* tools solve it, which limits the direct utility of such theoretical work in practice. For instance, *Vi&VA* tools naturally have a limited amount of data that can be displayed, which can limit the amount of knowledge attainable by the user [108]. Too much data can cause issues in performance or worsen the usefulness and understandability of the visualizations themselves [56]. Indeed, Federico et al. [72] also argues that since this process is conceptual, it is "often inconsistently used" by the literature, which shows a missed opportunity to define a consistent formulation to apply such theory in practice. This inconsistency has another consequence: *Vi&VA* tools are unable to

consistently communicate their users' *tacit knowledge* amongst themselves.

User Tracking and Provenance

When considering how users can respond back to the *Vi&VA* tool with what new knowledge they acquired and how it was attained, Federico et al. [72] defines two possible methods in Figure 2.1[B]: the direct externalization X of the *tacit knowledge*, such as writing it onto a text file, or exploration E within the *Vi&VA* tool, which establishes that the user's action, such as visual interactivity, is directly influenced by the users' *tacit knowledge*. To understand this dynamic over time, the concept of Provenance (see chapter 1) is brought forth.

Definition 2.2.1 (Provenance)**.** *Tracking and using data collected from a process, such as a Vi&VA tool being used by a user.*

Provenance Application in Graphs When applying provenance, some researchers focus on collecting and analyzing user behavior [62, 177, 241, 152, 107]. Each of these tools and techniques aims to collect and structure some part of the user's knowledge discovery process and analyze it. von Landesberger et al. [241], for instance, perform *behavior provenance* by modeling user behavior as a graph network, which is then used for analysis.

To perform *provenance*, however, one must first collect information from the user. Existing works have collected changes in datasets [62], updates in visualizations [20, 251], and other similar events in order to analyze and recall user behavior. On the other hand, tracking user's *tacit knowledge* is either done by manual feedback systems [21, 152], by manual annotations over visualizations [208], or by inference methods that attempt to extract users' insights by recording the users' screens, video or logs and extract from them interactivity patterns as a post-mortem task [20, 99]. Among these *Vi&VA* systems, InsideInsights [152] is an approach to recording insights through annotations during the user's analytical process. It has demonstrated that collecting user annotations is a legitimate way to extract and store users' tacit knowledge.

Provenance Types Provenance itself can be split into different areas: *Data Provenance* [204, 62] focus on tracking changes of data over time; *insight provenance* aims to track any discoveries by the user over time [90]; *Analytic Provenance* attempts to broaden the scope of provenance into anything that can constitute an analysis, such as the process of *EDA* [250, 148]; *behavior provenance* attempts to track any and all interaction events by the user; and *knowledge provenance* attempts to track the acquisition of new knowledge [63, 82]. Notice that these concepts are not mutually independent. For instance, researchers argue that *knowledge provenance*, where the *Vi&VA* tool will track the user *storyline*s, can be done by recording any change in the available datasets [62] (e.g., data pre-processing), which is also part of *data provenance*, or updates in visualizations [20, 251, 241], which is also part of *behavior provenance*. Of note among such works is the one of Chang et al. [41], which attempts to use visual analysis within a Knowledge Base system by storing tacit knowledge extracted from experts into a "compressed" format. Works such as these show examples of applying provenance to understand users' knowledge gathering.

Provenance Critique Many of the related works listed consider *knowledge provenance* as a subset of *data provenance*, assuming that all knowledge-related changes can be extracted from changes within the data itself. However, this does not match with the knowledge definition of *Vi&VA*'s knowledge models [193, 72] where certain concepts, such as how user tacit knowledge interacts with the *Vi&VA* tool's explicit knowledge, are overlooked within the literature of *knowledge provenance* and *data provenance*. Also, most related works do not tackle how to interpret multi-user *Vi&VA* workflows [20] where multiple different tacit knowledge sources can be compared or combined, nor allow to analyze how much of the explicit knowledge is attainable by the user [251].

Sharing *Storyline*s through Storytelling and Storyboarding

One potential application to the provenance of multiple users is the recall and retelling of their *storyline*s. Info-graphs are one of the most used forms to transform provenance data into a storytelling visualization [128]. The automation

of storytelling has also interested the *Vi&VA* community. Zhu et al. [264] details how info-graphs can be automatically generated through machine learning [46] or pre-defined rules [84]. When applied to storytelling, displaying insights as visual or textual annotations on top of visualizations has been the aim of several works [84, 47, 46].

Although these proposals and results are very relevant for storytelling, they only list the insights found but not the process of how a user might have reached them. Instead, other formats, such as slide decks, are shown to be better suited to tell a linear story discussing the questions or intentions users had when using a tool, their interactions while searching for an answer, and the answer itself. Other applications little explored in existing literature are ways to list or to contrast multiple users' gathered knowledge [200]. In other words, no approach found during the research of this dissertation evaluates how to use the collection of aggregated knowledge from multiple users extracted from *provenance* events (e.g., user interaction or new insight harnessed) to (semi)automatically generate shareable media like a slide deck.

2.2.5 Temporal Events versus Concept Matching (atemporal)

Another aspect of the modeling of *Vi&VA* is how time is interpreted and understood. That is, the *provenance* of the user's knowledge and behavior can be stored and analyzed as a temporal sequence of events or as a *state space* set, which denotes the set of all possible states of the *Vi&VA* tool, interaction possibilities, and aggregation of knowledge independent of time. In other words, although a *storyline* can be defined as a linear sequence of new knowledge over time by using the temporal relationship between events as what relates the events to each other, it can also be defined as a time-independent set of all gathered knowledge, where the similarity, correlation or repetition of events is the information used to relate the events to each other.

To better understand the distinction of temporal events versus atemporal *state space*, let us discuss it further. *Vi&VA* literature in knowledge modeling has shown that users' interactivity with *Vi&VA* tools can be understood in its temporal aspects, such as an iterative workflow of user intentions, behaviors, and insights [193, 194,

72]. In other words, *Vi&VA* can be modeled as a sequence of events, which can be classified as a manual event, such as user interactivity, or as an automatic event, such as certain data mining processes or *ML*. This event-based modeling of *Vi&VA* is what allows researchers to apply *behavior provenance* to *Vi&VA* since *behavior provenance* attempts to do something very similar: tracking certain types of behavior events over time. Still, although these works describe ways to apply *behavior provenance* in *Vi&VA*, their own limitation of how to correlate events independent of time becomes more apparent. In other words, how can one know if the event (e.g., click of a certain button) is the same as what happened by another user in the past? And how does the recurrence of such events (or lack thereof) influence the knowledge-gathering process of each user?

The differentiation between the temporal events and this *state space* association between events based on similarity is a concept not much investigated in the *Vi&VA* literature. Instead, the following two concepts are either merged or ambiguous when related works used *provenance*: the *temporal* aspect, which indicates what and when users executed *Vi&VA* tasks, and the *atemporal* aspect, which indicates what the possible *Vi&VA* workflow states (*state space*) and the possible transitions among states. Instead, such works either store the temporal sequences of events without indicating whether they occurred previously or only focus on the space state without recording the temporal sequence of events.

2.3 Knowledge Ontology and Structure

So far, I have discussed the current state of knowledge modeling research in *Vi&VA* and the relevance of *provenance* as a way to collect and track events associated with the users' *storyline*s. Now, once we have collected such data, we must structure it in some way to be able to store it. *Vi&VA* has proposed several *ontologies* usually through the Web Ontology Language format [160]. Authors of Shu et al. [202] describe *ontologies* as a structure that defines domain concepts and the relationships between them, providing a domain language that is meaningful to both humans and machines. *Vi&VA* researchers have applied *ontologies* to formally define the theoretical structure that best denotes the existing data and the workflow of said data [193, 241, 195, 45, 43, 175, 242]. Vis4ML [195], for instance, describes an

ontology for machine learning in *Vi&VA* so that other researchers can better model and understand their *ML* and data mining design process. Although such *ontologies* are relevant to the general theoretical definitions in *Vi&VA*, they by themselves are, by definition, only theoretical. That is, an *ontology* by itself does not tackle how it can be applied to structure any data generated from *provenance*, nor discusses how or if such data can be collected and used for downstream tasks, such as data analysis. In other words, when compared to the other concepts in Section 2.2.2, ontologies by design do not provide an overarching *theoretical system* to which can be used to apply *Vi&VA* theory in practice [44].

When considering other related works, especially works from the domain of Semantic Web, *Knowledge Graphs* (*KG*s) [74, 45] has aimed to be the best way to structure knowledge-related data. *KG* is a technique that defines a network graph schema following a specific *ontology*. For instance, works like DBPedia [12] show how using an *ontology* to design a *KG* to store knowledge-related data can be used in practice. Similarly, others have used *ontology* to create "knowledge databases" in the form of *KG*s [92, 75, 116]. Moreover, compared to typical databases, the structure of *KG*s focuses less on the usual row-based structure [40] but uses the relationships between taxonomies as the foundation of knowledge, which allows works like KGTK [116] to analyze and appraise the data within it in innovative ways, such as creating data science pipelines as part of the KG itself.

***KG*s Extensions** *KG*s are not the only technique that structures knowledge as a network graph. Event Knowledge Graphs [98] expands on the definition of *KG*s by enforcing the concept of event-based relationships between nodes of the graph. Another notable sub-type of KGs is the Temporal Knowledge Graph, which enforces the concept of temporal relationship between graph nodes. That is, Temporal *KG*s [92] encode relationships like "order of events" or "time difference between events" as the relationship between the graph's nodes.

By including event or temporal data in the network graph, *KG*s have been shown to be a promising technique to model the user *storyline*s. For instance, with the temporal and event concepts, *KG*s are able to structure the temporally-based event data of *provenance*. In addition, *KG*s can also structure *explicit knowledge* [12]

and *behavior provenance* [116]. As of yet, however, other works have only used *KG*s to solve one of the problems at a time.

***KG*s Techniques** Naturally, *KG*s and their sibling concepts are all considered network graphs that aim to structure knowledge-related data. Therefore, any network graph database, such as neo4j [162], and graph network algorithm and technique, such as Graph Neural Networks (GNNs) [118], graph visualizations [42, 106] and graph operations [116] like page rank and traveling salesman, can also be applied to them. *KG*s are therefore not limited to only providing a way to structure knowledge-related data but also allow for the storage, analysis, and appraisal of such data through a vast array of existing techniques.

2.4 Conclusions

The objective of this dissertation is to investigate and propose ways to Democratize *Vi&VA*, and the way I focus is by the automation of *Knowledge provenance* and *Behavior provenance* on *Vi&VA*, which includes proposing (semi)automatic methods to model, structure, store, query, and appraise the users' knowledge generation process. *KG*s have shown to be in a uniquely favorable position as the foundation to allow for the proposal of this dissertation.

This chapter discussed the role of Knowledge in *Vi&VA*, compared how the concept of knowledge is interpreted in theoretical and practical related works, and listed several downstream research topics, such as Search Engine, *provenance*, and Knowledge Graphs. From all these examples, the importance of the overarching concept of knowledge in *Vi&VA* cannot be understated. Yet, the existing literature shows gaps when connecting theoretical concepts and models to tools and systems. This dissertation provides a bridge where *Vi&VA* tools can use existing *theoretical methodology* to model and structure the user knowledge generation and use the existing practical techniques to store, query, and share the modeled knowledge. The advantages are many, such as the ability to automate the collection and analysis of user behavior, compare user's experiences and insights, compare *Vi&VA* tools' ability to provide users with valuable insights, use autonomous agents for aid during the knowledge gathering process, and search among existing knowledge

graphs for knowledge which was previously discovered.

Chapter 3

Q4EDA - From Visual Selection to Insightful Queries [1] [2]

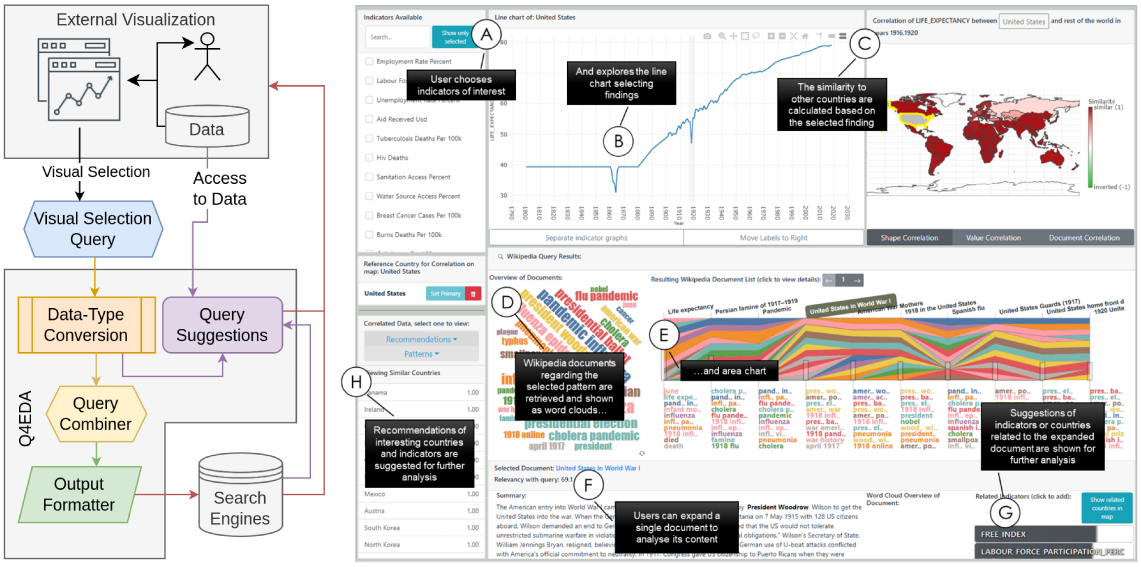


Figure 3.1: Q4EDA, a framework for converting user visual selections to search queries (Left), and LINKED, a sample implementation of the Q4EDA framework for visualizing world indicators and querying related information from Wikipedia (Right).

In order to understand how users pursue knowledge, the first contribution of this dissertation, which is summarised in Figure 3.1 presents an approach where users find visual insights within a VA tool and, through visual interaction, inquire and probe for any existing knowledge of said insight within an existing knowledge base: Wikipedia.

[1] This chapter was based on *Christino, L., Ferreira, M. D., & Paulovich, F. V. (2022). Q4EDA: A novel strategy for textual information retrieval based on user interactions with visual representations of time series.* Published in Information, 13(8), 368.

[2] Example of Use-Case: `https://www.youtube.com/watch?v=9JOmJ8kmANk`

3.1 Overview

Knowing how to construct text-based *Search Queries (SQs)* for use in *Search Engines (SEs)* like Google Search [142] or Wikipedia's [224] search-box function has become a fundamental skill. Though much data is available through such SEs, most structured datasets live outside their scope. Visualization tools aid in this limitation, but no such tools come close to the sheer amount of information available through general-purpose SEs. To fill this gap, this chapter presents *Q4EDA*, a novel framework that converts users' visual selection queries executed on top of time series visual representations, providing valid and stable SQs to be used in general-purpose SEs and suggestions of related information. The usefulness of *Q4EDA* is presented and validated by users through an application linking a Gapminder's line-chart replica with a *SE* populated with Wikipedia documents, showing how *Q4EDA* supports and enhances exploratory analysis of United Nations world indicators. Despite some limitations, *Q4EDA* is unique in its proposal and represents a real advance toward providing solutions for querying textual information based on user interactions with visual representations.

3.2 Introduction

Advancements in *Search Engines SEs* [58] represent a revolution without precedent for information dissemination. As mentioned in chapter 2, the ability to write *Search Queries (SQs)* [95] to request information from *SE*s. Although much data is available through such *SE*s, most of the structured datasets, such as the UNData [161] time-series world indicators, live outside their scope and cannot be incorporated into search queries. On the other hand, visualization tools are able to display the information contained in structured datasets. One example is Gapminder [186], where users can visualize animated charts displaying the UN-Data dataset. Despite their popularity, no individual visualization tool comes close to the sheer amount of information available through Wikipedia's hyperlinked text documents. Therefore, even if one uses, for instance, Gapminder to discover misconceptions [196], they will undeniably be required to search elsewhere for extra information regarding the underlying findings.

The task of generating text-based *SQ*s for *SE*s is, broadly speaking, done by the users themselves. However, this process can be challenging since the translation between user intent and keywords is sometimes not trivial, causing users not to find the information being looked for [120]. Beyond regular keyword lists, other search strategies are available to address such limitations. For instance, it is also possible to "search by image" [182], where users can find images similar to a given image, or "search by natural language" [34, 120], where users can search using descriptive texts.

However, when considering structured datasets there was no mechanism that uses pieces of visual data, such as user selections in a line chart, to search for related information using general-purpose *SE*s found during the writing of this chapter. Instead of using visual selections, there are solutions that either use visualizations as outputs [115, 15, 255] or as a visual interface to manually construct what is essentially a text-based query [130, 263, 26, 27]. Even though such visualizations are shown to aid the information-gathering processes, in these works, the use of selections as *Visual Selection Query* to fetch relevant information from general-purpose *SE*s is an unexplored concept. Also, various examples of search query suggestions for data analysis [168, 254] have been proposed, but the use of such techniques applied to *Visual Selection Query* during data analysis is similarly unexplored.

This chapter proposes *QuEry for visual Data Analysis (Q4EDA)*, a novel framework that converts user visual selections to relevant search queries to be used for general purposes in search engines. Using query expansion and suggestion techniques, one of the promising applications for *Q4EDA* is to allow users to perform an enhanced visual analysis of time-series dataset collections. To use *Q4EDA* in this context, a visualization tool, such as Gapminder or Tableau [261] captures and forwards user *Visual Selection Query (VQs)* to *Q4EDA*, then *Q4EDA* processes the selection and outputs an *SQ* which can then be used on general-purpose *SE*s. *Q4EDA* also suggests other potential time series related to the selected event within the dataset collection to be subsequently investigated. In summary, the main contributions of this chapter are:

- A conversion technique to transform visual selection queries into valid and stable search queries usable in general-purpose search engines;

- A strategy to expand the converted search query to better retrieve related text documents and provide suggestion ranked lists with data related to the Visual Selection Query;
- An exploratory data analysis application example that uses *Q4EDA* in practice to provide means to find more (textual) information related to an observed pattern compared to the standard manual keyword-based queries.

The remainder of the chapter is structured as follows. Section 3.3, discusses related work involving techniques that seek to interpret visual interactivity as queries and how they execute and use the results of their query. Section 3.4 formalizes the problem and outline the *Q4EDA* solution. Section 3.6 presents use-case examples of how visualization tools can use *Q4EDA*. Evaluations through user survey and query stability performance are then presented in Section 3.7, indicating a good degree of stability and reproducibility of the search queries and overall success of *Q4EDA* in enhancing users' ability to probe for relevant information of patterns or events found during visual data analysis. Finally, Section 3.8 discusses *Q4EDA* limitations, and Section 3.9 draws conclusions.

3.3 Related Work

In order to support users in visual data analysis, some tools and techniques allow for a non-obstructive exploratory approach through visual interactivity [211, 130], among the most relevant is NorthStar [130], which goes in-depth into the difficulties of providing an exploratory system that follows responsive and real-time guidelines for intuitive and engaging user exploratory analysis while at the same time utilizing automatic problem detectors throughout the entire workflow to reduce the amount of potential bias or incorrect insights generated through the exploration. However, although a certain level of "query conversion" is done through these tools or techniques, they still expect what is essentially a text-based query to be constructed manually through their visual interface. Furthermore, these tools use the query to retrieve data from domain-specific databases, which differs from the proposed query conversion method for search engines. Moreover,

although research has identified other ways to encode visual findings and hypotheses beyond visual selection query constructors [213], no existing work, as far as the authors know, has demonstrated ways to use such methodology to convert a visual selection into a search query as *Q4EDA* proposes.

Considering the perspective of Visual Selection Query and their related information from external datasets, some approaches focus on extracting or generating automatic visualizations and annotations. Despite representing significant progress, the amount of analysis enhancement they provide is limited to the generation of visual metaphors [59, 140, 31] and of visual feature extractions [31, 219, 66] produced from a single dataset used for analysis. Other works focus on generating enhanced visualizations by displaying one dataset as a visual feature of the other. These visual features can be annotations [115, 133], can involve textual queries to generate visualizations [255, 145, 154, 112], can define a question-answer interface to visualizations [127, 119, 257], can use textual datasets to help the understandability of structured datasets [15, 126, 255, 210], or can automatic link text to images for use within visualizations [156, 256, 65]. The research has shown approachable and engaging ways to optimize visualizations and analysis by using text-based datasets and *NLP* techniques. Of them, usage of heterogeneous data [65] and cross-modal methods [256, 257] has shown that utilizing multimodal datasets, such as time-series and text, provides significant advantages to the user's analysis. Although many of them use some query to retrieve information from text, no one uses external search engines as the target of such queries. Instead, they focus on specific text datasets. These efforts show a focus on analyzing a single dataset with the aid of another, which provides *NLP* capabilities during analysis. However, such approaches differ from the proposal of *Q4EDA* which does not just convert Visual Selection Query and provide valid search queries to be used in existing search engines but also provide query suggestions, aiding the analysis in many ways.

Visual Analytics has also recently started to use heterogeneous datasets for analysis, and one such work done by Zhou et al. [263] proposes a method of using users' interactions to quantify their attention and, from it, decide which medical documents to present to the user. Arguably similar to *Q4EDA*, Zhou

et al. [263] uses a textual database and a query system from their previous work Cadence [26, 27]. Although their text database does not significantly differ from existing search engines, hence approaching *Q4EDA*'s proposal, this work does not provide an actual query conversion process, nor propose methods of query suggestions, nor allow for visual selections as queries. Instead, queries are again constructed manually using the Cadence system through a drag-and-drop interface to perform the search. This procedure is arguably most similar to the interface of NorthStar [130] as opposed to *Q4EDA*.

Significant effort has been made by the information retrieval community to propose better ways to use search engines during data analysis. The practice of retrieving information [58] describes ways to expand a query, so its effectiveness is more relevant to users when executed in a *SE* [260, 39, 14, 64] and how to provide suggestions for future queries [168]. Although their techniques are very relevant to *Q4EDA*, the vast majority are non-visual approaches that describe advances in how to perform query expansion of a keyword or natural language search query as opposed to *Q4EDA*'s *visual selection queries*. Among the ones that use visualizations, they either use it only to display information, lacking interactivity [111], or use it as a query builder interface, where there is no concept of visual selection queries [125], focuses on visualizing a manually or interactively constructed query [198, 192]. Indeed, the authors are not aware of any research within the information retrieval community which transforms or converts visual selections into *SQ* automatically, that is, without any user intervention to manually, even if through visual interfaces, construct the search query. *Q4EDA* aims to fill this gap where users are not required to interpret the available data and manually construct the search query. Instead, users can visually select portions of existing visualizations, and the selection is automatically converted into a valid search query for use in search engines.

Both information retrieval and data management communities have had great strides in connecting heterogeneous data, such as time-series datasets and text documents. Currently, one of the major advances in this direction is the concept of dataspaces [60, 80] and knowledge graphs [64, 18, 151, 12, 254]. Manual, semi-automatic, and automatic procedures for matching and linking these have been studied [89, 158, 156, 10, 53, 96, 191]. Though some contributions links structured

datasets within visualizations to external data and others link data to text documents, no related work performs the full path from visual selections to search engine responses as far as the authors are aware. Though *Q4EDA* uses many of the concepts shared among these valuable contributions, *Q4EDA* blurs the line between visual interaction, database query, and information retrieval to deliver what it proposes.

In summary, existing works provide many relevant works which use textual datasets to enhance data analysis, but they provide no *visual selection query* conversion capabilities. Many do not support search engines nor provide query suggestions from text documents extracted from said search engines. Some newer work exemplifies the benefits of linking structured datasets to textual counterparts for data analysis enhancement. However, they still fail to provide a way for users to analyze datasets through visual selection queries, such as a box selection within a line chart, and simultaneously be provided suggestions of said visual selection. Instead, existing work either fails to provide the visual selection query, the query conversion, the support for *SE*s, or the query suggestions. This is the novelty of this framework: to provide a visual analytic workspace where external VA tools provide enhanced data analysis of time-series datasets by converting visual selection queries to search queries to be used within well-known *SE*s and supplying query suggestions to further enhance the analysis. To promote this query conversion, *Q4EDA* devises a novel approach using existing pattern analysis and natural language process strategies to convert user-selected findings and elements of interest to valid *SQ*s while also providing suggestions based on the selected finding, allowing users to navigate the data and build up knowledge.

3.4 Methodology

Here is presented *QuEry for visual Data Analysis (Q4EDA)*, a novel query conversion framework designed to enhance data exploration tasks by ingesting user's findings through visual selections and returning *search queries (SQs)* which can retrieve relevant information from general-purpose *search engines (SEs)*. By identifying user-driven visual selections of findings (or patterns) of interest, *Q4EDA* allows visualization tools to request the conversion of said selections into *SQ*s to

be used in keyword-based *SE*s to retrieve textual information related to the finding. Additionally, *Q4EDA* provides query suggestions with suggestions based on the similarity or correlation of the selected finding to other parts of the dataset under analysis. *Q4EDA* is specifically designed for time-series dataset collections, hence it expects the visualization tools to similarly include time-series data as the input visual selection query.

Definition 3.4.1 (Search Engine *SE*)**.** *A software responsible for performing information retrieval from a database given a text-based query. Usually refers to text-based engines, like Google Search or Wikipedia's search-box function.*

Definition 3.4.2 (Search Query *SQ*)**.** *The text-based input which is given to a Search engine to request information.*

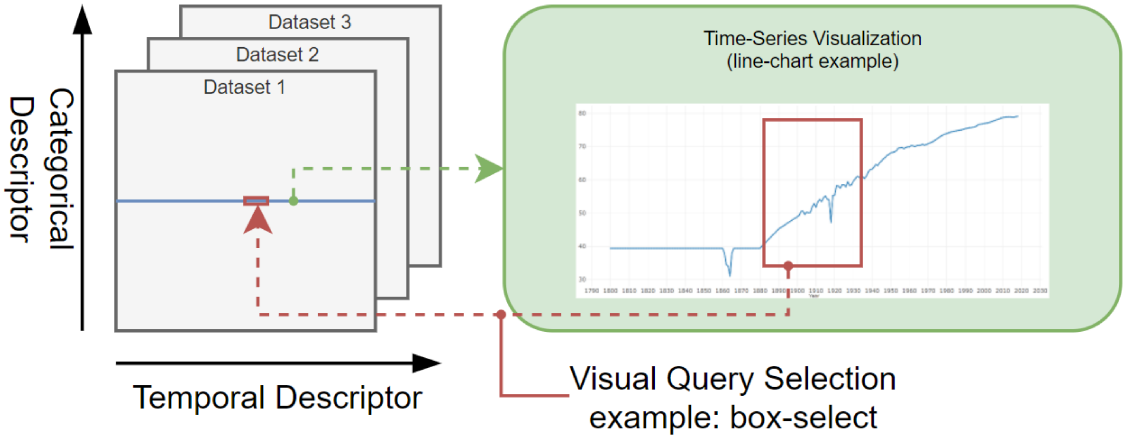
3.4.1 Overview

Q4EDA is structured as summarized in Figure 3.2. After a setup phase, a visualization may forward a *Visual Selection Query (VQ)* to *Q4EDA* to be processed and converted into a *Search Query (SQ)*. *Q4EDA* can be divided into three distinct steps: conversion, output, and suggestion. *Q4EDA* conversion uses the *VQ*'s *data-types*, such as numeric or categorical values. Similarly, each of *Q4EDA*'s outputs implements the *SQ* specification of a target *SE*. Finally, *Q4EDA* provides query suggestions by either correlating text documents retrieved from the *SE* to the dataset's available data or correlating the selected time-series pattern to other available time-series within the dataset. Although the presented implementation of *Q4EDA* attempts to emulate the data available within Gapminder, the framework is purposely flexible to allow for other applications, such as will be discussed.

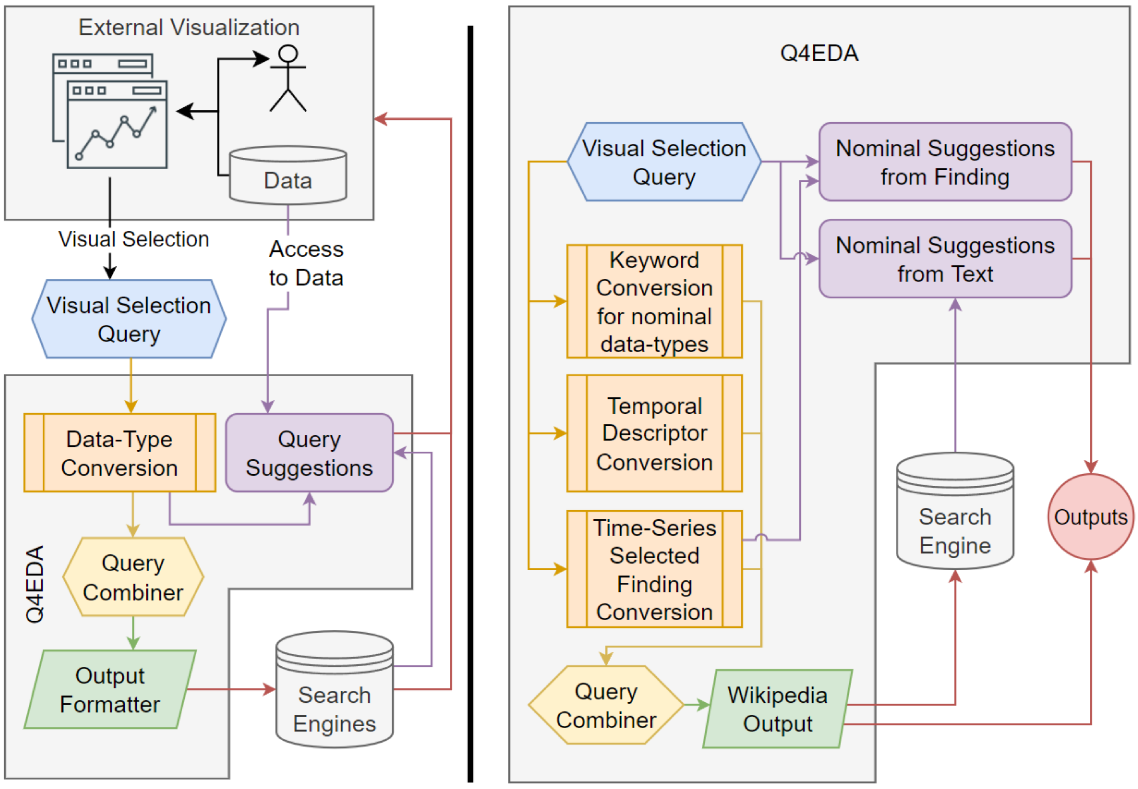
Definition 3.4.3 (Visual Selection Query *VQ*)**.** *A search query that is made based on visual interactivity, such as mouse hovering or selecting.*

3.4.2 Design Requirements

Q4EDA focuses on providing a *VQ* conversion framework for time-series dataset exploratory analysis. For example, if a user asks *Q4EDA* to generate a *SQ*



(a) Time-series dataset collection structure expected by *Q4EDA* (left) is composed of multiple time-series datasets. Each row of the many datasets are time-series which themselves are visualized through tools like Gapminder [189], and a selection within such visualization (in red) comprises the *Visual Selection Query (VQ)* converted by *Q4EDA*.



(b) Overview of *Q4EDA* conversion and suggestion processes. *Q4EDA* expects to have access to a dataset collection and a given user's *VQ* (left), and the inner operations convert each data-type into output queries which are combined as a valid *SQ* (right). After executing the *SQ* in a *SE* and retrieving the text documents, *Q4EDA* also provides query suggestions.

Figure 3.2: Overall summary of *Q4EDA*.

which represents a pattern of type "peak" within a time-series, *Q4EDA* should use the available information from the time-series data and the visual information of the *VA* to generate a text-based *SQ*. By combining elements of information retrieval, the gaps in the literature in regard to *VQ* conversion, and the goal stated above, a compiled list of requirements is made:

R1 – Visual Selection Query Conversion. *Q4EDA* must provide a valid *SQ* given a *VQ* that provides: (a) **output correctness** – the information returned from the targeted *SE*s should be related to the user's *VQ* [58]; and (b) **output stability** – the results should not vary too much given slight variations on the *VQ* since slight variations of user interaction (selection) are expected to happen in practice [39];

R2 – Information Retrieval. *Q4EDA* should follow existing approaches to enhance the retrieved results, including: (a) **query expansion** – the converted *SQ* should be expanded [14] with terms to broaden the query results to better allow for related information to be found; and (b) **query suggestion** – suggestions [168] should be provided to provide users with information of interest.

3.4.3 Q4EDA Definitions

As a first step, it is essential to define what an *SQ* is in the context of this dissertation. Many online resources define *SQ* as "A search query or search term is the actual word or string of words that a search engine user types into the search box" [83]. Beyond that, a plethora of similar concepts is seen when applied to different modes of communication. To simplify and better contextualize *Q4EDA*, the definition used for a *Search Query (SQ)* is: a term-based format of querying for information. Similarly, a *Search Engine (SE)* is defined as any system which can receive an *SQ* and retrieve information relevant or related to the query from its database(s), where among the most famous examples are Google Search, Wikidata (Wikipedia's Knowledge Base), Apache Lucene and Elasticsearch [70].

As discussed, *Q4EDA* is designed to be used by a data analysis visualization tool to convert a *visual selection query (VQ)* into a *search query (SQ)*, which can then

be used to retrieve information from a *search engine (SE)*. For this, *Q4EDA* expects to have access to the *time-series dataset collection* being analyzed. *Q4EDA* also expects the data to be organized as such: the dataset collection is made up of multiple datasets wherein lives the many time-series, as is exemplified in Figure 3.2(a). For instance, UNData [161] is a dataset collection where multiple datasets of so-called indicators (e.g. "life expectancy" or "child mortality") contain a per-country time-series. With this, the four main *data-types* of *Q4EDA* conversion are defined: *dataset name*, *categorical descriptor* such as country name, *temporal descriptor* and *time-series value*. With this, the definition of a *Visual Selection Query (VQ)* as a subset of a visualization's data which was selected through user interaction as exemplified by the red selection of Figure 3.2(a). For instance, Gapminder's line-chart [189] could theoretically allow for a box-selection or lasso-selection of a part of the displayed data with which the user can indicate a visual selection query. This *VQ* would include the selection years as the x-axis's temporal descriptor of the selection, the y-axis's time-series numeric values, and the corresponding dataset name and categorical descriptor of that individual line chart.

With these definitions in hand, *Q4EDA* can be described as a framework that receives a *VQ* and, by utilizing the available dataset, it converts the *VQ* into one of the output *SQ* formats. After executing said *SQ*, *Q4EDA* also provides query suggestions among the available dataset names and categorical descriptors. For that, the workflow to use *Q4EDA* follows the representation of Figure 3.2 where after a user performs a *VQ*, its data is given to *Q4EDA* for the query conversion and suggestion process.

3.4.4 Query Conversion

Q4EDA's framework leans on the three aspects: *query conversion*, *query combiner*, and *query suggestion*, as is seen in Figure 3.2. While the query combiner defines the supported *SE*s, the query conversions define the process used to convert each of *VQ*'s individual data-types to generate a relevant output *SQ* using query expansion techniques (*R2.a*). For this, *Q4EDA* first converts each data-type separately, and then the results are *combined* and *formatted* into the output *SQ*.

The inner mechanism used by *Q4EDA* to process and store the individual

conversion step follows a grammatical convention based on existing *SQ* grammars, the closest of which is Elasticsearch's Simple Query format [221]. The formal grammar described in EBNF [77] is:

⟨*sub-expression*⟩ ::= ⟨*or*⟩ | ⟨*and*⟩ | ⟨*required*⟩ | ⟨*term*⟩
⟨*or*⟩ ::= '(' ⟨*sub-expression*⟩ { '|' ⟨*sub-expression*⟩ }+ ')'
⟨*and*⟩ ::= '(' ⟨*sub-expression*⟩ { '&' ⟨*sub-expression*⟩ }+ ')'
⟨*required*⟩ ::= '"' ⟨*sub-expression*⟩ '"'
⟨*term*⟩ ::= [⟨*negative-factor*⟩] ⟨*inner-term*⟩ [⟨*weight-factor*⟩]
⟨*inner-term*⟩ ::= '(' ⟨*spaced-term*⟩ ')' | ⟨*word*⟩
⟨*spaced-term*⟩ ::= ⟨*spaced-term*⟩ ' ' ⟨*word*⟩ | ⟨*word*⟩
⟨*word*⟩ ::= { ⟨*lower-case-letter*⟩ }+
⟨*weight-factor*⟩ ::= superscript ? ⟨*weight-factor*⟩
⟨*negative-factor*⟩ ::= '−'

where the conversion's *sub-expression* output consists of multiple *terms* which can represent the input *positively* or *negatively*, and with the added *weights* and logical operations, the output is able to be descriptive enough for formatting to target many *SE*s. In order to exemplify the grammar, by assuming an input requesting *Q4EDA* to convert the question "population of the United States" into a *SQ*, a plausible output could be $\{(\text{united states} \mid \text{usa} \mid \text{america} \mid (\text{north america})^{0.5})\&(\text{population} \mid \text{habitants} \mid \text{people}^{0.5} \mid (-\text{death})^{0.5})\}$, where it includes perfect match terms (e.g. "united states"), terms with lower weights for less exact matches (e.g. "north america") and negative terms to indicate opposite meaning or antonyms (e.g. "death").

Keyword Conversion

The first conversion tackles the keyword data-type, such as the dataset name or categorical descriptors. By acquiring the dataset name(s) from a given *VQ* or the categorical descriptor(s), *Q4EDA* uses natural language processing to generate a set of related terms for inclusion in its output. *Q4EDA* first applies a text mining approach to assign a set of related terms $T_d^D = \{t_1^d, t_2^d, \ldots\}$ for every keyword d received. For instance, the keyword $d =$ "life expectancy", which is a dataset name from UNData [161], besides being represented by the terms

"life" and "expectancy", can also be represented by terms like "longevity" and "lifetime". Similarly, the same keyword is negatively represented by terms like "mortality" or "death". Such an example results in the terms $T^{D}_{life\_expectancy} = \{life, expectancy, longevity, lifetime, -mortality, -death\}$ where positive terms are related to the keyword, and negative terms are negatively related, a concept similar to antonyms. In this way, any such nominal descriptors will have their semantic meaning defined by their associated terms. Finally, the set of terms is formatted into an expression following *Q4EDA*'s grammar: $(life \| expectancy \| longevity \| lifetime \| - mortality \| - death)$.

This operation is done through a pre-trained GloVe model [174], which generates related tags for a given keyword. First, the GloVe model is loaded using the Gensim library [181], and then every unique keyword d is tokenized and lemmatized with NLTK [143] and WordNet [73, 260] and passed to Gensim's "similar_by_vector" method.

Country Keyword Conversion

Although the simple method described so far works well for dataset names and categorical descriptors with common nouns, specialized versions of keyword conversion are used to dataset names or categorical descriptors with proper nouns, such as geo-locations such as street, building name, city, state, or country and events such as "the second world war", for better query expansion results [23]. Therefore, *Q4EDA* provides one of such specialized keyword conversions to properly process a country keyword.

Since geo-information also encodes geographical concepts like continents, distances, area, and population, among others, this specialized conversion can output a more relevant set of terms for a given country. That said, this module design could expand indefinitely due to its data complexity, therefore, I limited the processing to grammatical variations, naming conventions, synonyms, and regionality. To create the set of terms T^{C}_{c} of words for a country keyword, the country's name is used along with other terms related to the country, such as adjectives or nouns (e.g. "United States" adds "America", "American" and "USA"). Although a VQ may be specific for a single country, the VQ's relevant information within the

target *SE* may include other similar or neighboring countries as well. Therefore, terms for the collection of countries near the selected country were added, such as the continent name or sub-area, with a smaller weight (e.g., "United States" adds "North America" with lower weights). This generalization has shown to be essential because some textual information may only contain more general references to the geographical location where a specific event occurred. To complement T_c^C with said extra data, data from Gapminder's geography dataset [187] was extracted and used as a reference for any country *categorical descriptor*, such as is the case with UNData. Therefore, such an example results in the following output:

$$\begin{aligned} T^C_{united\_states} = &(\text{united states} \mid \text{united states of america} \mid \\ &\text{american} \mid \text{america} \mid \text{usa} \mid (\text{north america})^{0.5}) \end{aligned}$$

Finally, the list of terms is converted to *Q4EDA* inner grammar definition.

Temporal Descriptor Conversion

Unlike previous processors, the temporal descriptor conversion is unique since it tackles a continuous temporal value and has some inner context that can be used to enhance search queries. Similar to the country keyword conversion, this step is required to be specialized to a given temporal variables, such as date, time, month, weekday, year, or any combination of them. Therefore, *Q4EDA* provides one such implementation which focuses on converting a year range into a valid query output to be included in *Q4EDA*'s output *SQ*.

The term set generated has the form of $T^E_{y_a,y_b} = \{t^e_1, t^e_2, \ldots\}$ where y_a and y_b identify the limits of the range of years. The natural terms associated with a range of years are the years themselves. However, this conversion process goes beyond and allows the VA Tool to define a weight distribution to better describe the selection interest within the range of years. For instance, while the default weight profile will give the same weight to all years within a range, if the VA Tool includes a Gaussian weight distribution, the processor will give higher weights to the center of the range and lower to the others.

Additionally, the year conversion compares the year range to a predefined year

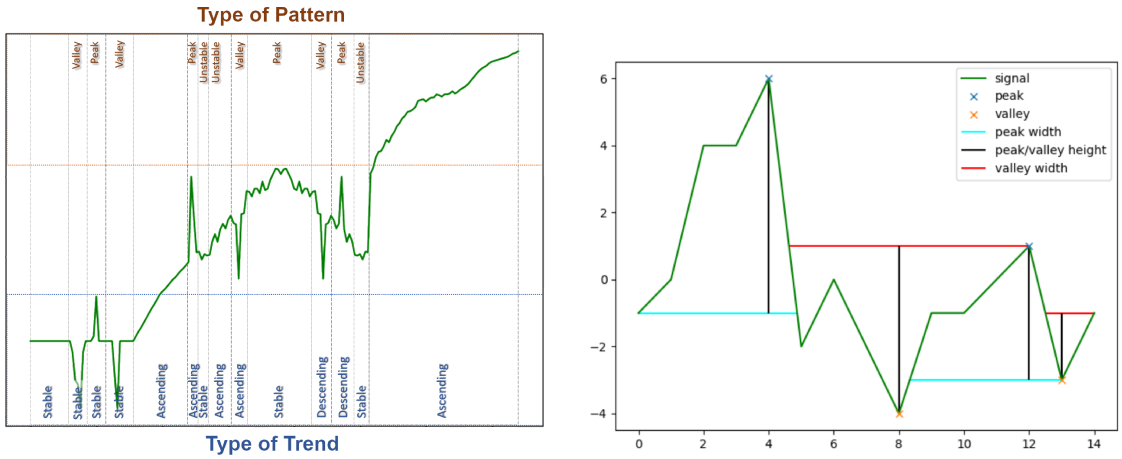
range term. For example, if the year range is $1950 - -1960$, an additional term of $1950s$ is included to represent the decade selected. If the year range does not match the exact decade, lower weights are given to the decade terms. The weight used in decade terms is $w = 1 - 2 * (|ds_a^b| + |ds_b^e|)$ where ds_a^b is the distance from the first year to the beginning of the decade and ds_b^e is the distance in years from the last year to the end of the decade, and weights equal or below 0 causes the decade term to be discarded. The same process has been applied for centuries. Therefore, such an example results in the following output:

$$T^E_{1851--1859,gaussian} = (1851^{0.2} \mid 1852^{0.3} \mid 1853^{0.5} \mid 1854^{0.8} \mid 1855 \mid 1856^{0.8} \mid 1857^{0.5} \mid 1858^{0.3} \mid 1859^{0.2} \mid 1850s^{0.6})$$

Time-Series' Selected Finding Conversion

Finally, *Q4EDA* analyzes the numerical values selected by the user $F = I_c^d(y_a, y_b) = \{e_{y_a}, \ldots, e_{y_b}\}$ and converts the underlying pattern into a set of terms $\{t_1^p, t_2^p, \ldots\}$. Its implementation categorizes the selected values, which is here called *finding*, by its trend tr, which can be either *ascending*, *descending* or *stable*, and by its pattern tp, which can either be *peak*, *valley*, and *neutral*. This results in 9 different possible trend/pattern combinations, as exemplified by Figure 3.3(a). These two identifiers are applied to a finding by using existing trend/pattern statistical methods, and the resulting keyword is then converted to the desired query output using the same GloVe model [174] *NLP* process used in the Keyword Conversion (see Section 3.4.4). That said, if the input has only one value (e.g., $a == b$) or no value, this conversion step will output nothing at all.

The type of a selected trend tr is defined by applying the Moving Averages (MA) method, a well-known technique in time-series analysis to define whether a series is stationary or not, providing us with a trend estimation [30]. MA first transforms the selection $F = I_c^d(y_a, y_b) = \{e_{y_a}, \ldots, e_{y_b}\}$ into a new series $F' = I_c'^d(y_a, y_b) = \{e'_{y_a}, \ldots, e'_{y_b}\}$ setting its values $e' \in I_c'^d(y_a, y_b)$ to the average value in the time interval $e'_p(w) = \mu(e_{y-w}, \ldots, e_y, \ldots, e_{y+w})$. Here, I empirically define the window with the default value $w = 2$. However, it is possible to modify this parameter during setup. The discrete derivative of F' and the sum of the normalized values



(a) Combinations of trends and pattern types in visual selections.

(b) Peaks and valleys detection.

Figure 3.3: Examples of trends and patterns combinations and peaks and valleys detected by *Q4EDA*.

are computed as follows

$$tr = \sum_{e'_y \in F'} \frac{e'_y - e'_{y-1}}{|e'_y - e'_{y-1}|}. \tag{3.1}$$

Based on that, the trend type is defined as

$$trend = \begin{cases} \text{ascending}, & tr > 0 \\ \text{descending}, & tr < 0 \\ \text{neutral}, & \text{otherwise} \end{cases}. \tag{3.2}$$

To define the pattern type tp of the selected finding, *Q4EDA* uses peak detection methods that attempt to identify a local maximum by comparing neighboring values [240, 252]. In this process, all potential peaks k_p within the selection were identified, and similarly, all potential valleys k_v are identified by inverting the selection data points as $F_v = -F = -I_c^d(y_a, y_b)$. Using this method, a pattern factor pf was assigned to the selection F, computing the following equation to determine whether it contains a peak, a valley, or neither.

$$
\begin{aligned}
pf &= |F| \times \tfrac{(w^+ - w^-)}{\sigma(F)}, \\
w^+ &= \textstyle\sum_{k_p} W(k_p).Pr(k_p), k_p \in peaks, \\
w^- &= \textstyle\sum_{k_v} W(k_n).Pr(k_v), k_v \in valleys,
\end{aligned}
\tag{3.3}
$$

In Equation 3.3, W represents the width and Pr the prominence, or absolute height, of the peak or valley. Therefore, w^+ is equivalent to a probability of the selection to be a peak and w^- a similar probability of being a valley. Note that $\sigma(F)$ is the standard deviation of $F = I_c^d(y_a, y_b)$. Based on the pattern factor pf, the type of pattern is defined as

$$
pattern = \begin{cases}
\text{stable}, & \sigma(F_c^r) < \lambda_1 \\
\text{peak}, & \sigma(F) > \lambda_1 \text{ and } pf > \lambda_2 \\
\text{valley}, & \sigma(F) > \lambda_1 \text{ and } pf < -\lambda_2 \\
\text{unstable}, & \sigma(F) > \lambda_1 \text{ and } \lambda_2 \geq pf \geq -\lambda_2
\end{cases},
\tag{3.4}
$$

where λ_1 is a threshold of whether to consider if a selection contains a pattern or not, and, if a pattern is detected, λ_2 is a threshold to define whether the selection is a peak, a valley, or an unstable oscillation. The parameters $\lambda_1 = 0.5$ and $\lambda_2 = 1.5$ were set empirically, but these can be changed by users of *Q4EDA*.

The result of this process is a pair of identifiers (*trend* and *pattern*), which describes the selection and is exemplified in Figure 3.3(b). The conversion's output is defined by using the identifier pair as keywords to be converted with the same GloVe model presented in Section 3.4.4, in which *Q4EDA* expands a keyword into a valid query output to be used within the output *SQ*.

3.4.5 Query Combiner and Output Formatter

So far, the conversion process was exemplified with a singular input to each. Each conversion outputs a single query output in the sub-expression format of Section 3.4.4, with a valid sub-expression representing that specific. However, *Q4EDA* is not limited to only one input per data-type. Instead, each conversion process is done for **each occurrence of its input**, therefore if the input has multiple datasets $\{D1, D2, \eta\}$, the keyword conversion will be executed for each individual dataset name and return one sub-expression per dataset T^{D1}, T^{D2}, η. All other

processors would similarly be executed multiple times if the *VQ* includes multiple occurrences of the input metadata-types.

Q4EDA then combines every sub-expression into a full expression by first creating a full combinatory permutation within each sub-expression output, excluding the time-series pattern conversion, since its output will already be associated with the combination of the others. For instance, if the *VQ* contains two countries, one dataset name, and two ranges of years, then *Q4EDA* would also expect four finding sub-expressions, one for each of the country/name/years combinations. With this, each of the two countries would be converted individually T^C_{c1} and T^C_{c2}, the dataset name would be converted to T^D_d, the two-year ranges of the selection would output $T^E_{y_a,y_b}$ and $T^E_{y_c,y_d}$, and the four findings would output four distinct T^P. *Q4EDA* then expands the combinatory permutations as four inner queries by the combinatory permutations one by one and concatenates each with an "and" & operation, resulting in the following four sets of sub-expressions: $(T^C_{c1}\&T^D_d\&T^E_{y_a,y_b}\&T^P_{c1,d,y_a,y_b})$, $(T^C_{c1}\&T^D_d\&T^E_{y_c,y_d}\&T^P_{c1,d,y_c,y_d})$, $(T^C_{c2}\&T^D_d\&T^E_{y_a,y_b}\&T^P_{c2,d,y_a,y_b})$ and $(T^C_{c2}\&T^D_d\&T^E_{y_c,y_d}\&T^P_{c2,d,y_d,y_c})$.

Then, to unite the sub-expression sets, two expressions are calculated: a full expression intersection T_I by concatenating all sub-expressions sets with "and" & operands and a complete expression union T_U by concatenating all sub-expression sets with "or" $\|$ operands. These two are then united as $T = (T_I)^2 \| T_U$, where the full intersection is given higher weight over the full union. Note that $(T_I)^2$ applies a weight operation to a sub-expression set instead of an inner-term, as defined by *Q4EDA*'s grammar. Therefore, every term within the complete expression intersection should have its weight multiplied by 2. Finally, the two full expressions are summarized through a Boolean Algebra [134] which includes Exponentiation Algebra to also solve for the weight-factor calculation.

Finally, by using an output formatter to reformat the final query calculated above into a valid *SQ* of a given *SE*. *Q4EDA* provides one such formatter which converts the query to the Elasticsearch simple query format [221] by replacing the following aspects of the query:

Other than the direct equivalents listed above, which are simply replaced, the *negative-factor* is unavailable within Elasticsearch and, therefore, removed. Note that the term itself with a *negative-factor* is kept as regular *operands* (e.g.,

Table 3.1: *Q4EDA*'s output formatter replaces bnf the required output tokens based on the targeted Search Engine (*SE*)

bnf	inner example	Elasticsearch equivalent
weight-factor	superscript	$\wedge$
negative-factor	$-$	N/A
and	&	$+$
required	"spaced term"	+(spaced term)
spaced-term	some words	"some words"

"$(-mexico^{0.5})$" becomes "$mexico^{0.5}$") since querying for antonym terms of the finding can also represent information about it [260].

3.4.6 Query Suggestion

In parallel to the *SQ* conversion, *Q4EDA* also provides query suggestions by utilizing and analyzing the available dataset collection, the *VQ* and the final text documents results from the conversion process, as is shown in Figure 3.2. In other words, *Q4EDA* not only implements query expansion (*R2.a*) by aggregating relevant terms to the *SQ* through the conversion processes but also implements query suggestions (*R2.b*) in terms of the visual selection query by suggesting other partitions of the dataset collection which may potentially be related. By loosely following information retrieval techniques [168], *Q4EDA* tackles two suggestion approaches where users are suggested related dataset names and categorical descriptors. First it provides suggestion given the presence of said dataset names or categorical descriptors within the text documents which were retrieved from the *SE* after executing the final *SQ*. Second, *Q4EDA* provides suggestions based on the numerical or pattern similarity of the *VQ* finding to other possible VQs among all the datasets and categorical descriptors.

Nominal Suggestions from Text

The first suggestion approach attempts to search the available nominal data's presence, which includes both the dataset name and the categorical descriptor, in a given text document. For that, a *VQ* is required to be converted and its output *SQ* is executed within the target *SE*. The resulting text document(s) are used by this suggestion process to rank the nominal data present on each text

document by the number of occurrences. The result provides the suggestion of other datasets or categorical descriptors which are related to the text document being read. In order to broaden the use of this suggestion approach, *Q4EDA* provides three implementations: direct counting, indirect counting, and natural language processing (*NLP*), and the result is a score list that associates each nominal data to each of the texts. That is, following the previous sections' examples, *Q4EDA* suggests the most relevant datasets and countries within UNData that relates to any given Wikipedia text document which was retrieved from Elasticsearch after executing the result of a *VQ* conversion.

In the direct mode, *Q4EDA* uses a simple case-insensitive number of occurrences of all possible nominal data from the dataset, or, in other words, a bag-of-words technique is applied. That is, *Q4EDA* suggests a ranked list indicating the number of occurrences of every dataset name and another ranked list for the categorical descriptor. In the indirect mode, however, *Q4EDA* first uses the same keyword conversion approach of Section 3.4.4 to expand the available keywords from the dataset name and categorical descriptor, and their query output's terms are then searched within the text document using bag-of-words. Finally, with the *NLP* mode, *Q4EDA* compares the keyword conversion outputs to the extracted keyword list from the text document through another *NLP* method called Gensim Keywords [181], which transforms text documents into keywords. Finally, *Q4EDA* once again uses NLTK with the GloVe model [174] to transform both sets of terms, namely the keyword conversion and from the text keyword extraction, into two vector embeddings which are then compared with cosine similarity. The result is, once again, one ranked list per nominal data which indicates suggestions indicating the similarity between the text contents and the available data within the dataset collection under analysis.

By assuming *Q4EDA* is configured with the Wikipedia Elasticsearch Dump [223], UNData time-series data [161], and Gapminder line-charts [189], and that the user performs at some point a visual selection query which, after converted and executed, returns a Wikipedia document about USA's life expectancy, then *Q4EDA* scans the whole UNData dataset collection across the two nominal data, namely dataset name and country, and search for all their keywords (e.g., United States or

Life Expectancy) within the text document. In the direct mode, *Q4EDA* would provide two lists containing the number of occurrences of each country and indicator within the text documents. In indirect mode, *Q4EDA* converts each dataset name and country using the keyword conversion process and searches the resulting terms (e.g., United States, USA, American for the united states country or Life, Expectancy, Death for the life expectancy indicator) within the text document, and the occurrences of each term are aggregated as its respective dataset name or country as an average. The results are also two lists indicating the average occurrences of related terms of each country and dataset name. Finally, with the *NLP* mode, *Q4EDA* also converts the text document itself into a keyword list and by converting both this list and the keyword conversion query output terms to a vector embedding format, *Q4EDA* calculates their cosine similarity. Once again, the results are two lists indicating a similarity score between the text's generated keywords and the related terms of each country and dataset name. Independent of the mode used, the resulting lists are a country ranking list of the most related countries to the text and an indicator list of the most related indicators to the text documents.

While the direct mode provides suggestions of other nominal data present in the text and the indirect mode provides suggestions of other nominal data whose concept is present in the text, the *NLP* mode provides suggestions of nominal metadata whose concept matches the most prominent concepts within the text. In other words, the direct mode is better to find texts literally talking about "life expectancy", and the indirect mode is better to find texts with information that talks about the concepts surrounding "life expectancy" of a country (e.g. including "death" and "mortality"), and the *NLP* mode is better to find texts whose overall context is regarding the concepts surrounding "life expectancy" of a country. In the case of suggesting UNData nominal data from Wikipedia text documents, through preliminary tests, it was gathered that the direct mode provided better ways to suggest nominal data based on the presence of specific words within the text, such as "child" of the child mortality indicator, the indirect mode provided better ways to suggest similar countries of a given *VQ*, and the *NLP* mode provided better ways to suggest similar datasets of a given *VQ*.

Nominal Suggestions from Pattern

The other query suggestion approach focuses on analyzing the selected finding of Section 3.4.4 and providing related nominal data, such as dataset name or categorical descriptor, as suggestions for further analysis due to their similarity or dissimilarity. By comparing the time-series numeric values of the finding $F = I_c^d(y_a, y_b) = \{e_{y_a}, \ldots, e_{y_b}\}$ to all other equivalent findings among the nominal data present within the dataset collection, *Q4EDA* is able to calculate a similarity list for each nominal data using statistical correlation techniques, as is exemplified in Figure 3.4. For instance, if considering the UNData, *Q4EDA* would output one list indicating the similarity of the finding of all other countries $c' \in C, c' \notin c$ if all other inputs are the same, which includes the year range and dataset name, and another similarity list of all other datasets $d' \in D, d' \notin d$ if all other inputs remain the same, this time including the year range and country. Among the two lists, the most similar suggestions as related to the *VQ* are shown, while the most dissimilar (e.g. lowest similarity score) are shown as suggestions of an "opposite" or "inverted" pattern to the *VQ*.

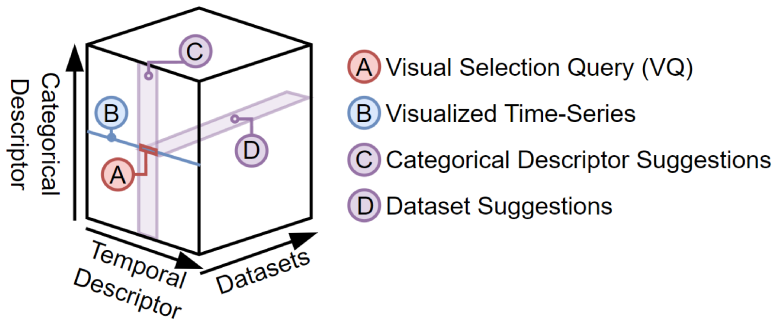


Figure 3.4: Example of the two suggestion lists given the dataset collection and the user's *VQ* (A) of the visualized time-series (B). *Q4EDA* calculates and outputs two suggestion lists: a dataset list indicating similar datasets to the *VQ* given the same time-range and categorical descriptor, and another list indicating similar categorical descriptors (e.g. country) to the *VQ* given the same time-range and dataset.

For this, *Q4EDA* provides two different strategies to compute the aforementioned similarity: Pearson correlation and Dynamic Time-Warping (DTW) [124, 147]. These two techniques are provided because each one solves the limitations of the other. For instance, Pearson Correlation is the choice if one prefers to compare

findings purely according to their general visual pattern similarity while not focusing on the distance between the selection's individual values. However, if one wishes to compare findings according to differences in amplitude or consider their raw value distances, DTW is the option. The Pearson correlation is calculated as follows

$$corr(F, F') = \frac{1}{N^2} \sum_{i}^{N} \frac{(x_i - \mu(F))(x'_i - \mu(F'))}{\sigma(F)\sigma(F')}, \tag{3.5}$$

where $\mu(\cdot)$ is the average value, $\sigma(\cdot)$ is the standard deviation, N is the number of values in the patterns, and x and x' are the values in from two functions F and F' respectively. Correlation $corr(F, F')$ between the two functions F and F' ranges in $[-1, 1]$. Positive values indicate linear related series, negative inversely related series, no relationship otherwise.

The second option, the DTW, is a robust dissimilarity measure that finds the non-linear alignment that has the lowest accumulative Euclidean distances between points, resulting in an optimal shape match preserving magnitude [124, 147]. Since the correlation is a similarity and the DTW is an unbounded dissimilarity, the DTW dissimilarity is transformed into similarity to keep consistency as follows

$$dtw_{sim}(F, F') = \frac{1}{1 + dtw(F, F')}, \tag{3.6}$$

where $dtw(F, F')$ is the DTW distance between two patterns, and the resulting similarity $dtw_{sim}(F, F')$ ranges in $[0, 1]$.

The resulting suggestion lists contain the correlation score of all other possible variations for each of the respective nominal data, as seen in Figure 3.4. This list represents suggestions of similar or dissimilar findings extracted from the dataset collection given the user's VQ. With these suggestion lists, users can, for instance, directly analyze other related countries or datasets of the UNData or even visualize the lists as a similarity heat-map, or, in the case of geographic information, choropleth geography maps, for example.

3.5 LINKED - Example use of Q4EDA as an Integrated VA tool

This section presents *LINKing hEterogenous Data (LINKED)*, a VA tool powered by *Q4EDA* that automatically links correspondences between UNData time-series sets and Wikipedia textual excerpts during exploratory analysis based on user interaction by using *Q4EDA*. LINKED uses a novel approach merging natural language processing, information retrieval, and signal processing strategies to decompose findings (user selections) into queries to unravel potential descriptions of the observed patterns inside large textual databases to provide added certainty to the user's analysis results. It also offers the possibility of navigating from textual descriptions back to related time-series through automatic suggestions, providing a bi-directional connection between the two data sources. LINKED uses UNData world demographic indicators provided by the Gapminder initiative and a custom-built Wikipedia Search-Engine as a textual source of information in the presented implementation. Through a set of user tests, LINKED was attested to enhance users' analytical capabilities by loosely coupling these heterogeneous datasets in a single tool, which when analyzed statistically results in considerably more information found that otherwise would be ignored or not discovered when using the datasets independently. In summary, the main contributions of LINKED are:

- A baseline back-end architecture to integrate at two datasets of the same domain: one structured in a tabular time-series structure, such as UNData, and another unstructured textual dataset, such as Wikipedia;
- An application of *Q4EDA* as the query-generation system which serves to translate the selection of a visually identified pattern from a line-chart time-series into a query and retrieves relevant data from the Wikipedia unstructured textual dataset;
- A reverse query-generation strategy that summarises a textual entry to suggest related data from the time-series dataset;
- A single visual analytics dashboard to show the time-series and the textual datasets providing related analytical functionalities.

LINKED is designed as a dashboard with well-established visual metaphors employed to visualize time-series as line charts, geographical data as maps, and text data as wordclouds and area charts (see Figure 3.5). LINKED is implemented using a web-model architecture based on back-end and front-end services, where the front-end uses React, and the back-end uses Python[3]. Although the presented example focuses on using UNData world indicators and Wikipedia historical textual information, LINKED can be extended to be data agnostic, allowing it to be extended for other scenarios and domains by replacing the datasets used throughout this work with a different textual and time-series datasets.

LINKED Data and Design Requirements

As discussed, the focus of LINKED is to link time-series dataset(s) with textual descriptions during the execution of exploratory tasks. LINKED uses multiple world demographic indicators provided by the Gapminder initiative [186] where each dataset holds a single indicator's data per country, and Wikipedia [222] as the textual source of information. The bidirectional process of linking these two data sources is accomplished through a comprehensive list of design requirements that need to be supported, involving:

R1 – Visual Selection Query Between Datasets. Allow users to select a pattern of interest in a time-series, which is hereby called "finding", and search for related textual information through *Q4EDA*, providing potential explanations for the observed finding;

R2 – Document Summary. Display the retrieved documents with visual summaries for fast results analysis, but not hindering users if they wish to read the original document itself;

R3 – Suggestions/Recommendations. Suggest indicators (time-series) and countries for further analysis considering the selected finding and/or each retrieved document;

[3]LINKED can be accessed at `http://expatt.vav.aknakos.com`

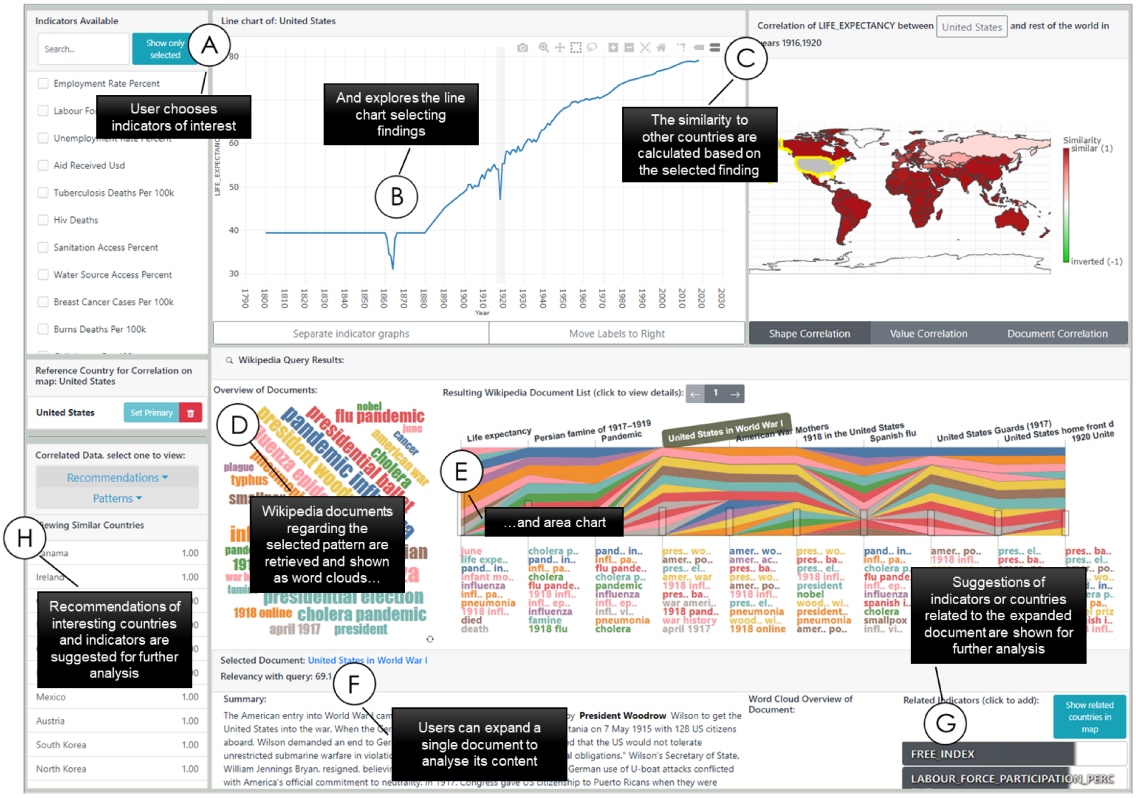


Figure 3.5: LINKED interface setup to explore world demographic indicators, a time-series structured dataset. Using the line chart visualization (A), users can select findings (B) and further analyze the indicators through a correlation world map, which shows whether other places in the globe present similar behavior to the selected finding (C). LINKED then links the time-series dataset to a separate Wikipedia database, an unstructured textual dataset, by transforming the user selection onto a textual query and showing its result as document overview visualizations (D, E), per-document summaries (F), and per-document suggestions of other countries or indicators related to the document (G). Also, suggestions of other countries and indicators of interest related to the selected finding are shown (H) for further analysis.

R4 – Correlation. Provide comparison between the selected country and indicator and other countries and indicators, allowing users to find similar and dissimilar countries or indicators;

R5 – Custom Query Allow users to run a custom search query to directly fetch documents and start the analysis from the suggested countries and indices.

These requirements were compiled by multiple prototyping phases with informal feedback from other students within the Visual Analytics Lab I work. Next, I discuss how LINKED implements these requirements.

3.5.1 LINKED Overview

To set notation, consider the whole group of world demographic indicators I where I can be partitioned through C countries and D unique indicators per country. Each unique indicator I_c^d is composed by a set of time-series defined as $I_c^d = \{e_{t_1}, \ldots, e_{t_n}\}$, where e is each data-point entry measured among $\{y_1, \ldots, y_n\}$ years. This notation is intentionally equivalent as the one used by *Q4EDA*.

In LINKED, the user initially selects one or more indicators d (Figure 3.5(A)) and a line-chart is shown presenting I_c^d (Figure 3.5(B)), which can be used to select a time-frame of interest. This "time-frame of interest" will be referenced as *finding* F for the remainder of the section, and is defined as $F = I_c^d(y_a, y_b) = \{e_{y_a}, \ldots, e_{y_b}\}$ where $y_1 <= y_a < y_b <= y_n$ are the time-frame selection boundaries in the time-series. The selected finding and all its related data is then forwarded to *Q4EDA* in order to build a search query. By using the generated search query on the Wikipedia Search-Engine, LINKED retrieves potential explanations of the finding to be displayed to the user. Figure 3.5(A) shows an example of selecting a finding, which is displayed as a gray area. The visualization shows a massive decrease in the "Life Expectancy" of "United States" in the period between 1917 and 1919 (**R1**).

Users may use the map in Figure 3.5(C) to define which country or countries c to be displayed on the previously described line-chart and when a finding is selected, the map is colored based on the similarity of the selected finding $I_c^d(y_a, y_b)$ with all other countries C as in $\bigcup_{c' \in C} I_{c'}^d(y_a, y_b)$ considering the same indicator. Users can use the map to investigate whether there are other interesting findings F' similar to the

selected finding in other countries, whether the selected country is different from other regions of the world, or even to get an overview of the finding's similarity distribution around the globe. For example, by starting with Figure 3.5(B), it is possible to use LINKED to analyse similar behavior to the selected finding and see that the same pattern is also observed in multiple countries (e.g., United States, Canada, and Venezuela), which is in contrast to the behavior of other countries such as Mexico and the United Kingdom.

LINKED uses *Q4EDA* to translate the finding into a search query q, which is sent to the search engine holding the second dataset: the Wikipedia documents. The retrieved documents are then summarized (**R2**) using *Q4EDA*'s *Nominal Suggestions* into a keyword list, which is then used to generate the *overview wordcloud* (Figure 3.5(D)) and the interactive *explanation river overview* (Figure 3.5(E)). In this overview, the documents are positioned in the x-axis, and the rivers' width represents the probability (or frequency) of keyword terms in each document. These keyword terms are also listed below the chart to facilitate navigation with an added translucent bar, indicating the search engine score of each resulting document in relationship with the search query. The same color-scheme is used within the WordCloud and the Stacked Area Chart in order to identify the appearance of the same word or phrase among both visualizations.

On user interaction with the *explanation river overview*, each text document can be expanded on-demand to display four per-document elements: the text summary (Figure 3.5(F)), a per-document wordcloud, the related list of indicators (Figure 3.5(G)), and a map of mentioned countries, all of which are used to propose next steps to the user's exploratory analysis. According to the example presented in Figure 3.5, the finding observed in Figure 3.5(B) is probably related to the "Influenza Pandemic" and "World War I", which can be extracted from investigating the document list (Figure 3.5(E)).

At the same time, *Q4EDA*'s *Nominal Suggestions from Pattern* provides suggested countries and indicators of interest for the same selected time-frame of finding F, which populates the lists in Figure 3.5(H). The **Suggestion Lists** includes a set of lists containing other suggested indicators and countries related to the selected finding (**R3**). For example, if a user wants to discover the countries

with the most similar or dissimilar shape to the selected finding, they can use the "Similar/Dissimilar Countries" lists, which show the same or the inverse information of the correlation map using a ranked list format (**R4**). Another example is when the user wants to discover indicators with the most similar or dissimilar shape to the selected finding's shape while considering the same country, which can be done using the "Similar/Dissimilar Datasets" lists. Finally, "Prominent Peaks/Valleys" provides lists to aid the analysis of the finding's shape.

In addition to the analysis starting with a linechart analysis, users may also wish to start their analysis through the Wikipedia documents. LINKED provides a "manual query mode" (**R5**) for users to provide a manual search query to directly retrieve documents from the Wikipedia search engine. Then, users can begin their analysis from the retrieved documents and the corresponding suggestion lists. For example, a user can start their analysis by querying a term (e.g., "influenza"), and from each document's textual summaries, visualizations and recommendations, they may add some of the recommended indicators and countries to their linechart time-series.

3.5.2 Search Engine

In order to search for data within Wikipedia, LINKED uses an open-source No-SQL database, and search engine based on Apache Lucene called Elasticsearch [129] since it is optimized for fast text indexing, processing, and searching even with large datasets.

The setup of the database is comprised of multiple technical steps outlined in [70]. Elasticsearch requires a JSON-based query configuration of fields, per field weights, and other similar configurations to search text documents within the database. The database uses this to return the documents with the highest score while matching the specified query and boosting the per-field weights. After performing internal tests with students of my lab, the query settings of LINKED were configured to use a weighted-based of $W_t = 2$ for title weight and $W_b = 1$ for body weight, which boosts the score of matched documents with related titles. Finally, LINKED uses *Q4EDA*'s formatter where its generated search query is in the Elasticsearch simple query format [221] (Section 3.4.5).

3.5.3 Visualizing the Explanations

Once the documents are retrieved, the last step in transforming findings into explanations is to present the fetched information (**R2**). LINKED presents this information as an overview visualization outlining the top k retrieved documents (k is a user parameter, which by default is set to 10) and visualizations of the overview of results and a summary of each document on demand. Users can also use buttons to fetch the subsequent k documents, not hindering or forcing users to only use the first k documents for their analysis.

3.5.4 LINKED Discussions and Limitations

Given the design requirement of using well-known and straightforward visualizations to build LINKED, such as maps, line charts, area charts, and wordclouds, some visual representations have inherent limitations regarding visual scalability, for instance, when displaying too many time-series at once as pointed by some participants in the qualitative feedback. However, it was opted to use simple and popular visual representations that are more familiar to the general public, as was confirmed through interviews, user testing, and user evaluation feedback. Although other more scalable visual representations could be used, for instance Visception [132] with map-based layouts, the added complexity and the prolonged user learning curve would reduce the reach of LINKED.

3.5.5 LINKED Conclusion

This section presented LINKED, a system to enhance the exploratory analysis of world demographics by linking potential patterns of interest, called *findings*, to textual documents from Wikipedia through a novel visual selection query mechanism. This is done by the visual analysis of both datasets simultaneously, where one aids the understanding of the other.

3.6 Use-cases and Results

3.6.1 UNData Line charts and Wikipedia

This section presents a hypothetical usage scenario of *Q4EDA*, showing how it can assist external VA tool users, such as LINKED, in understanding patterns of interest and building up extra knowledge during exploratory data analysis. LINKED is used here to implement a replica of Gapminder's line chart visualization [189] to allow for the exploration of the UNData dataset [161] as a VA tool coupled with an Elasticsearch [70] *SE* loaded with Wikipedia text documents [223].

Here, I introduce Justin, a fictional American High School student. He wants to investigate how the average lifespan has changed over the years in the United States. With that in mind, he visualizes the "Life Expectancy" indicator of the "United States" country (Figure 3.6). The resulting line-chart shows a positive trend toward increasing the overall American lifespan over the years. However, he notices two interesting patterns, one valley between 1860 and 1866 and another between 1917 and 1919 with some instability between 1901 and 1930.

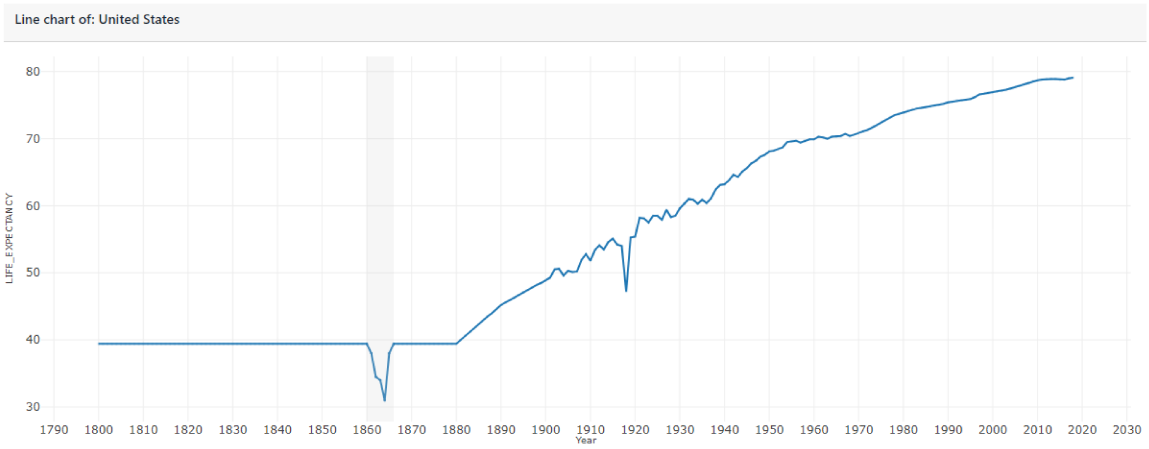


Figure 3.6: Life expectancy line chart of the United States. Two noticeable valleys are observed between 1860 and 1866 and between 1917 and 1919. The gray area represents the user selection.

To further inspect these patterns and understand what is happening, Justin first selects the left-most valley (between 1860 and 1866). The enhanced VA tool uses *Q4EDA* to automatically generate a search query, retrieve related Wikipedia documents, and get a list of suggestions of related countries and datasets considering the

selection (Section 3.4.6). Based on that, Justin adds the two top-ranked suggested countries to the line chart (Sweden and the United Kingdom) and realizes that the drop in life expectancy is probably an American effect since even the most related countries do not present similar valley in the same period (omitted due to space constraints). With that in mind, Justin checks the retrieved documents and observes the prevalence of the terms "civil war" in the returned snippets (Figure 3.7(a)) and concludes that the lower life expectancy rating may result from a civil war. By reading some of the retrieved documents, he learns that the trigger of the civil war was the result of Abraham Lincoln's election and the United States southern states feeling unrepresented and/or challenged due to slavery, being able to discover one important piece for the storytelling of the United States lifespan variations, including its trigger.

Justin follows up to investigate the second dip in life expectancy and selects the period between 1917 and 1919. The retrieved documents' contents are, however, less homogeneous with different causes for the drop, with a span of documents discussing different topics (Figure 3.7(b)). As in the previous example, he adds other top-ranked suggested countries to the line chart, including countries from different continents. Differently from the previous valley, all the included countries present a similar drop in life expectancy in the same period (Figure 3.8), apparently suggesting that a global event took place. Rechecking the retrieved documents, he infers that some potential reasons for the changes in life expectancy may be the World War I and the Spanish Flu, also called Influenza Pandemic.

Still using the countries suggestion list, Justin discovers that Russia is amongst the lowest-ranked countries and decides to investigate – here is discussed the country suggestion as a ranked list. Adding Russia to the line chart, Justin finds out that the reason for the similarity score being low in this period is that Russia's valley is much wider than the United States (image omitted due to space constraints). By selecting the valley in Russia's life expectancy and checking the suggested countries, he discovers that Belarus, Ukraine, Turkmenistan, Uzbekistan, Tajikistan, Kazakhstan, and Uzbekistan are the most similar to Russia. Also, by checking the retrieved documents (Figure 3.9(a)), Justin discovers that during 1917 and 1919, Russia was facing the abolition of its monarchy in 1917, a civil war, and the

(a) Selection between 1860 and 1866.

(b) Selection between 1917 and 1919.

Figure 3.7: List of documents retrieved from Wikipedia related to the drop in the United States life expectancy.

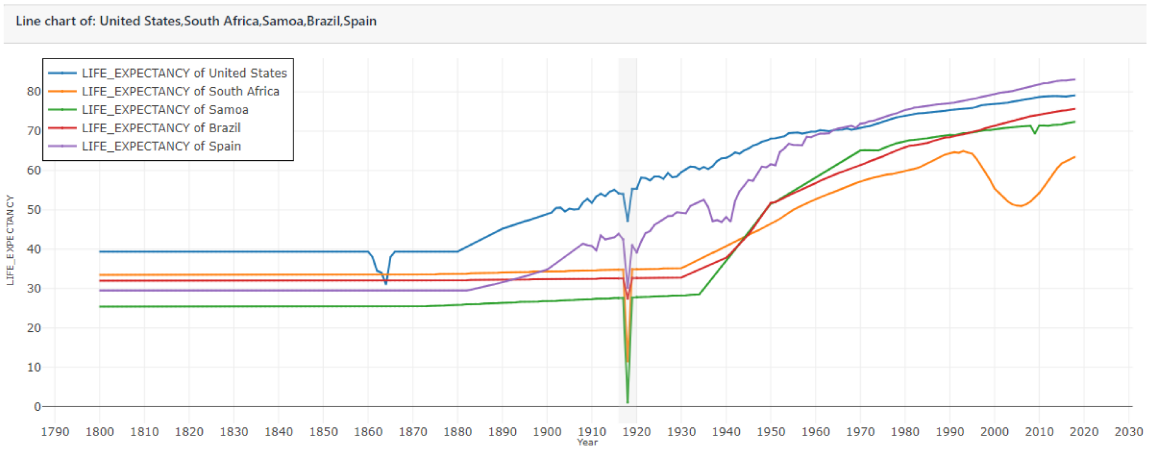


Figure 3.8: Life expectancy line chart multiple countries. All countries present the same valley between 1917 and 1919, indicating a global reason for the drop in life expectancy.

beginning of the Soviet Union, besides likely facing World War I and the Influenza Pandemic as he saw with the rest of the world.

Besides suggesting countries, *Q4EDA* also suggests other datasets with correlations given a selection (Section 3.4.6). Justin notices that the "Democracy Index" dataset is suggested, so he creates a line-chart displaying Russia's life expectancy and democracy index together (Figure 3.10). In the resulting visual representation, he notices an inverted pattern (a valley in the life expectancy and a peak in the democracy index) between 1917 and 1923. Justin then selects such a time period and, considering the list of returned Wikipedia documents (Figure 3.9(b)), he discovers that the sudden change in the democracy index of Russia can be attributed in some part to Russia's constitution of 1918, "Russian Famine", "Russia's Civil War" and the "Russian Revolution", which tells about how the monarchy was abolished in 1917 and the Soviet Union was established in 1923, matching the steep fall of Russia's democracy index in 1923. Justin concludes that the political landscape variation in Russia definitively impacted the lives of the Russian population, including their life expectancy.

Although an elementary analysis, Justin discovered pieces for the life-expectancy variations in the United States, including its trigger. Two different global situations, World War I and the Spanish Flu. Why Russia was differently affected than the rest of the world at the time and much of Russia's history on revolutions and economic

(a) Selection of the valley of Russia's life expectancy.

(b) Selection of the drop of Russia's democracy index score.

Figure 3.9: List of documents retrieved from Wikipedia related to the pattern in Russia's dataset between 1917 and 1919, indicating several potential events that may have affected it.

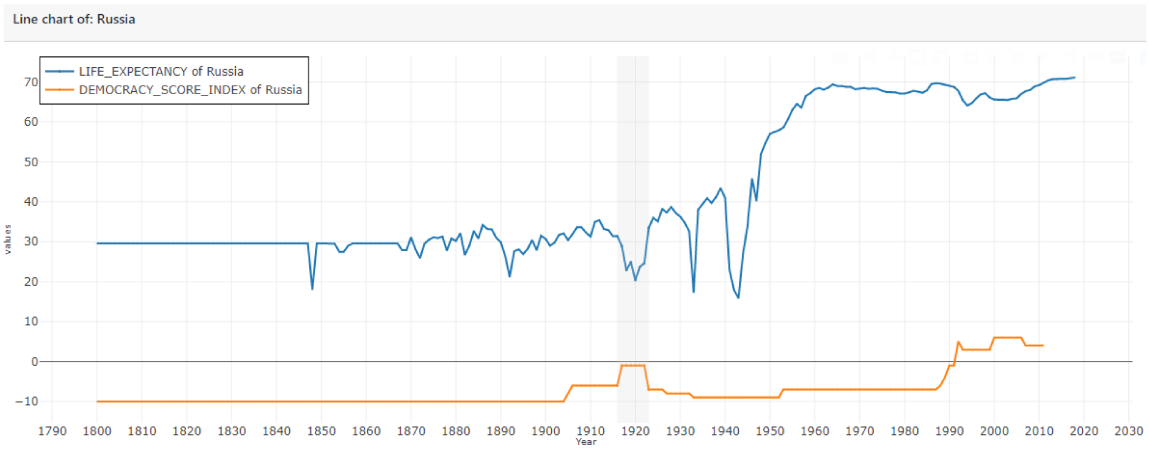


Figure 3.10: Russia's life expectancy and democracy index present interesting similarities.

crises.

3.6.2 Inner Query Stability Evaluation

Although so far it was assumed that users had provided precise visual queries, user selection is not expected to be exact every time due to the inherent limitations of visual interfaces and means of visual selection. In order to evaluate this, an extensive automatic evaluation was done to verify how much the Wikipedia results change when slight variations of the visual selection query occur. This also aims to evaluate how well *Q4EDA* solves goal *R1.a*. This evaluation follows Memon et al. [153] and creates an oracle that emulates the user behavior, allowing us to test this aspect of *Q4EDA* exhaustively. Notice that this evaluation is not validating the search for results themselves since no labeled dataset that would enable us to evaluate the connection between time-series visual patterns and the text documents of a search engine like Elasticsearch was found.

The oracle implementation is straightforward. Given a time series, the oracle iterates over it, automatically extracting the top patterns classified by the height of their valleys and peaks; then, the oracle varies a window around each pattern to generate similar selections that emulate users' selection inaccuracies. The window size of the experiments is set to 3, resulting in 9 queries per pattern (one original and eight derived). This value was defined by running a test with my lab members asking them to manually select patterns in several different time series, considering

the average variation among them to set the window. The stability of *Q4EDA*'s query process is then measured by comparing the intersection of the set of documents retrieved using the original query q and the sets of documents fetched using the derived queries q'. Given D the set documents retrieved using the original query q and D_i the list of documents returned using one of the derived queries q'_i, the stability is computed as

$$\frac{1}{|D| \cdot |\Delta|} \sum_{D_i \in \Delta} |(D \cap D_i)|, \tag{3.7}$$

where Δ is the set containing the lists of documents produced by all derived queries q'. Notice that the number of documents in D and D_i is the same and defined by the number of documents displayed in the interface, which was empirically set as 10.

To execute a comprehensive test, 960 time series were selected from the UNData dataset [161], and then the query stability was measured for each one. The evaluation automatically detected 5,286 relevant patterns, resulting in 47,574 queries submitted to the search engine. Figure 3.11 summarizes the results. Overall, on average, the stability is 0.5121, meaning that slight variations in the selection return 51% of the documents returned by the original query. More specifically, peaks and valleys are more stable, with an average of around 64% and a standard deviation of 0.22. At the same time, 'unstable patterns' have less document stability with an average of 39% and a standard deviation of 0.27. This suggests that the peaks and valleys are meaningful within *Q4EDA*, indicating that it appropriately translates well-defined visual patterns into coherent groups of documents. Although this cannot quantify the quality of the retrieved documents, it indicates that users with similar behavior are expected to receive similar documents when attempting to select the same pattern, indicating a good degree of stability and reproducibility.

3.7 User Evaluation

Using the same workflow as Justin's usage scenario (see Section 3.6.1), a user study was conducted to evaluate: (1) whether *Q4EDA* conversion method allows users to more accurately find textual information related to a specific time series

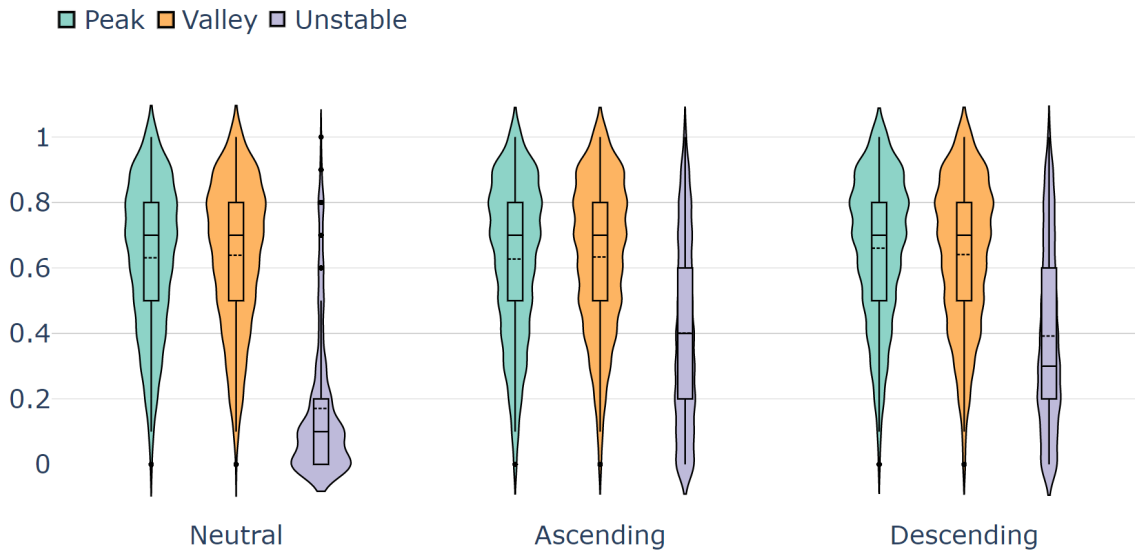


Figure 3.11: Query stability analysis. On average, slight variations in the selection return 39% to 64% of the documents returned by the original query depending on the pattern type, indicating a good degree of stability and reproducibility, especially for patterns with peaks and valleys.

pattern if compared to manually searching (*R1* and *R2.a*); and (2) whether users are more accurate in their query results using *Q4EDA* even when confident in their findings (*R1.a*). LINKED was used as a replica of Gapminder [186, 189] to represent the times-series and the Wikipedia search engine. This setup was used given the Gapminder popularity and to avoid search results variations that may occur between different user profiles if, for instance, the Google search engine is employed. Through this study, I aim to check the following null hypotheses:

> $\mathbf{H_0^P}$: There is no difference in the amount of correct information related to a given pattern of interest a user can find using the proposed visual selection query conversion and manually querying the target search engine.
>
> $\mathbf{H_0^C}$: There is no difference in the amount of correct information related to a given pattern of interest a user who was **confident** in their answer can find using the proposed visual selection query conversion and manually querying the target search engine.

Overall, the evaluation recruited 21 participants of ages 16 and up. All but two

are from computer sciences, one is from social sciences, and the other claim to not have a primary area of study. Due to COVID-19, it was not possible to conduct an in-person study. Instead, it was conducted through a self-guided online survey. The experiment first conducts a demographic questionnaire and introduces the Gapminder replica with a video tutorial. Participants are then guided through an interactive tutorial using the system and finally are given a list of tasks to be performed. The study was executed in an automated and non-obstructive manner through an online survey system. Participants joined the study through their machines and had unrestricted time to complete it. The study took on average 50 minutes, but due to the self-guided nature of online surveys, a third of the participants did not complete the study in one continuous session.

To evaluate *Q4EDA*'s hypotheses, I have randomly split the participants into two groups, one containing 11 participants, called *Cohort 1* (*C1*), and another containing 10 participants, called *Cohort 2* (*C2*). Both groups execute two sets of tasks. For the first task, *C2* is the experimental group while executing tasks using *Q4EDA*, and *C1* is the control group executing tasks without visual queries but instead manually searching Wikipedia to find related information to the observed patterns. Therefore no search query conversion capabilities are provided to *C1* as opposed to *C2*. For the second task, the roles of *C1* and *C2* are swapped.

The benchmark of this user study was done by manually searching both Wikipedia and through visual selection query for the expected text documents and phrases and confirming that all tasks can be done with both tools. It is important to note that both tools use and provide interfaces to the same time-series dataset and the same textual dataset. The study does not attempt to capture all possible historical events and facts related to a pattern but only to capture some equally available to be queried in both formats.

For the first set of tasks, using the "United States life expectancy" dataset, the participants were asked to search for probable causes for the drop between 1860 and 1866. The participants were given 5 alternatives and asked which one is related to the observed patterns (four correct and one incorrect). Answers from the experimental group using Q4EAD were better in finding more texts related to the time-series pattern. On average, participants manually searching Wikipedia

(*C1*) answered correctly 31% of the alternatives presented regarding related causes against 48% using *Q4EDA* (*C2*). For this task, all participants but one from *C2* said they now understand better *Q4EDA*'s usefulness for this kind of task and, by using the *Q4EDA*'s suggestions (see Section 3.4.6 and *R2.b*), 7 participants from the experimental group were also able to find other related indicators that may further enhance the exploratory analysis.

In the second task, the two groups were swapped so that *C1* is the experimental group using *Q4EDA*'s query conversion and suggestions and *C2* is the control group directly querying Wikipedia. In this task, the users were asked to investigate another drop in the United States life expectancy between 1916 and 1919. Similar to the previous set of tasks, participants using *Q4EDA* had overall higher performance. Participants manually searching Wikipedia (*C2*) answered correctly 38% of the alternatives presented of related causes against 68% using *Q4EDA* (*C1*).

To verify the null hypothesis $\mathbf{H_0^P}$ across the two sets of tasks, the control group was compared directly using only Wikipedia and the experimental group using *Q4EDA* across **all** tasks. The calculated p-value is $P_v^{H_0} = 0.0006 < 0.05$ and t-value is $T_v^{H_0^P} = -4.33$. Therefore, $\mathbf{H_0^P}$ can be rejected. In more specific terms, when comparing the number of answers from each group that matched the expected answers, the control group had, on average, identified 4.33 fewer pieces of information compared to the experimental group, matching what was expected.

The participants were also asked how confident they were with their answers. The answers where the participant was at least $Conf \geq Agree$ in their confidence level had a p-value of $P_v = 0.011 < 0.05$, rejecting the hypothesis H_0^C, indicating, therefore, that when only considering the participants who claim to be confident in their answers, the confident participants of the control group had fewer answers matching what was expected when compared to the confident participants of the experimental group. When considering that these two groups were similarly confident in their answers, it is possible to conclude that the confidence of the control group does correlate to a correct answer as well as experimental group, or in other words, given a pattern of interest, confident *Q4EDA* users have more accurate textual findings than users who directly queried Wikipedia.

In conclusion, participants of the experimental group significantly outperformed participants of the control group, showing that *Q4EDA* is effective in providing means for users to better and more correctly discover information in regards to a visual pattern and also to provide an effective way to convert a visual selection query into useful Wikipedia queries when compared to constructing the query manually.

3.7.1 Historian (Expert) Interview

In order to evaluate *Q4EDA*'s impact in real visualization-based data analysis applications, I interviewed Catherine, an MA student in History, while using the replica of Gapminder's line-chart from Section 3.6.1 to validate and align *Q4EDA*'s goal with real use-cases. For her undergraduate honors dissertation, Catherine researched the Second World War's effects on specific aspects of her city, omitted for confidentiality reasons. In her dissertation, her research workflow involves querying open portals through keywords about the research topic and visiting local libraries. Tasks are classified by her as long and time-consuming when considering the amount of time needed to retrieve information successfully.

In this interview, I initially introduced *Q4EDA* and its LINKED implementation. I described the data contained within as "World Demographics Indicators extracted from the United Nations repository". Her initial reaction was of great interest. She demonstrated enthusiasm with the breath of potential uses of *Q4EDA* once she saw it working. After being shown *Q4EDA*'s query conversion mechanism in action to retrieve relevant Wikipedia documents related to a finding of interest, Catherine explained that this simple action of visual interaction is great to empower lay users to find potential explanations for their findings. She also described that it would be of great help within her research if *Q4EDA* supported regional time-series data, such as provincial or municipal level, and regional text data, such as local libraries and news data, potentially reducing the effort involved in the preliminary information retrieval phase of her research. Catherine was asked how she would use *Q4EDA*, and she displayed significant interest in analyzing the History of Russia due to its very eventful History over the past 100 years. Finally, she concluded the interview by saying: "Overall, I think this is great! It is a really

cool program, and I am sure a lot of people will really benefit from using it."

3.8 Discussions and Limitations

This chapter has explored a method to convert a visual selection query into a search query usable within search engines. The technique was evaluated by converting patterns within world indicator dataset collections and enhancing users' analysis through Wikipedia documents. Indeed, the framework is open and easily usable in other domains, which may also benefit from interpreting time-series visual selections as search queries. Different input datasets, such as news, stock market, financial data, IoT sensor data, social media, and natural disasters dataset collections could also be used in a similar fashion. Although some of such dataset collections may be directly supported by the processes implemented so far, others may contain new data-types, which would also require new conversion processes to be implemented, requiring modification to *Q4EDA*'s structure. All authors of the published paper referent to *Q4EDA* expect future applications of *Q4EDA* in domains other than the one exemplified by LINKED.

Alternatively, in line with the expert interview, a more fine-grained demographics analysis, considering cities or neighborhood levels and local newspaper press datasets could be supported as well, among many other potential usages. That said, the main challenge in all examples is mostly designing good visual metaphors for domain-specific problems and discovering the best *SE* and textual document collections to be used for each specific domain, all of which is external to *Q4EDA*'s proposed framework. In any case, a fundamental observation has to be made: the textual snippets are not necessarily intended to find an exact cause for a particular visual finding since spurious correlations may happen.

Most of the exploratory tests executed in this chapter resulted in valuable retrieved textual information. However, it failed to bring meaningful documents for some specific demography indicators during the evaluation, though manually searching the same search engine gave similar poor results. Indeed, the usefulness of the query conversion depends on whether the *SE* has relevant information about the selected patterns and how well both the user and the *SE* can parse the data. For instance, if Wikipedia has no information on some subject for a given

pattern, *Q4EDA* may still return documents that may aggregate next to nothing for the analysis, which is not surprising. Therefore, *Q4EDA* by design forgoes how to match other time-series datasets to potential *SE*s, expecting instead this to be decided by whoever sets up and uses the *Q4EDA* framework (e.g., external VA tools).

Another limitation pertains to the suggestion approaches. *Q4EDA* suggests relevant nominal data given the visual selection query and the text documents by counting the number of the textual elements shared among the two either directly or through *NLP*. Updates to the suggestion processes are planned so that *Q4EDA* can also include the temporal descriptor (e.g., year) within the search or include semantic analysis over the text. The suggestion and keyword conversion processes could also benefit from other *NLP* techniques. Important to note that other *NLP* techniques were tested as part of the development process, namely Latent Dirichlet Allocation (LDA) [24], and DistilBert [183]. The ones here presented were used due to their speed in processing hundreds of text documents in less than a minute.

The usage scenario and user evaluation discussed were also tested using Google as a search engine, and the preliminary results were arguably more informative than Wikipedia. Indeed, Google API was the first choice for most of the examples discussed. However, its connection to the Gapminder replica was not ideal due to Google's imposed limitations, such as the limited number of API calls and limited information parsing. Also, other challenges were encountered, such as the processing of unstructured web pages as opposed to well-defined Wikipedia documents and the significant variations of results given the user's profile.

Therefore, even though both output modules were available, Wikipedia was preferred to be used to exemplify and evaluate *Q4EDA*. Also, since no labeled dataset where patterns within time-series data are linked to textual information within a search engine was found during the writing of this chapter, *Q4EDA* was unable to use common metrics, such as accuracy or f1-score, to evaluate its resulting query. Compiling a ground-truth dataset is possible through cooperation with experts and other researchers. Indeed, it is hoped that *Q4EDA* will be used once such a labeled dataset is compiled in the future.

Finally, although the main usage scenario focuses on a line-chart view, *Q4EDA*

does not impose such requirements on the visual metaphor. For instance, if the Gapminder's bubble chart [188] is used, a *VQ* over it would consider the two datasets displayed as the chart's axis. Hypothetically, such a VA tool could expect a box selection within the main bubble chart visualization, which would make the *VQ* include one year and a range of countries. However, the VA tool could also expect a selection within the animation timeline of the bubble chart, which would make the *VQ* include a range of years and all available countries or even a combination of the two selection modes. In summary, *Q4EDA* is agnostic in terms of which visual representation was chosen.

3.9 Conclusions

In this chapter, I presented *Q4EDA*, a framework that converts a *visual selection query* into a *search query* format to be used in existing *search engines* and, from its result, suggests other potential aspects of the data to be analyzed, all of which providing a novel strategy to utilize user input for textual information retrieval. I also presented LINKED, a use-case implementation of *Q4EDA* with a VA dashboard that displays UNData as visualizations, collects visual selections from users, and, from *Q4EDA*'s results, queries a Wikipedia database and displays its results. The usefulness of *Q4EDA* is brought by an application linking a Gapminder's line-chart replica with Wikipedia documents to support exploratory analysis of world indicators. The improvement in users' exploratory analysis capability is then confirmed through a user test showing that users can find more information using *Q4EDA* compared to the standard manual keyword-based queries, especially when confident in their findings. The stability of the conversion process given slight variations of its inputs was also evaluated in order to verify the applicability of *Q4EDA* with inaccurate visual selection interfaces. Despite its limitations, *Q4EDA* is unique in its proposal, representing an advance toward providing solutions for querying textual information from general-purpose search engines based on user interaction with visual representations.

Chapter 4

ChatKG - Visualizing Time-Series Patterns Aided by Intelligent Agents and a Knowledge Graph [1] [2]

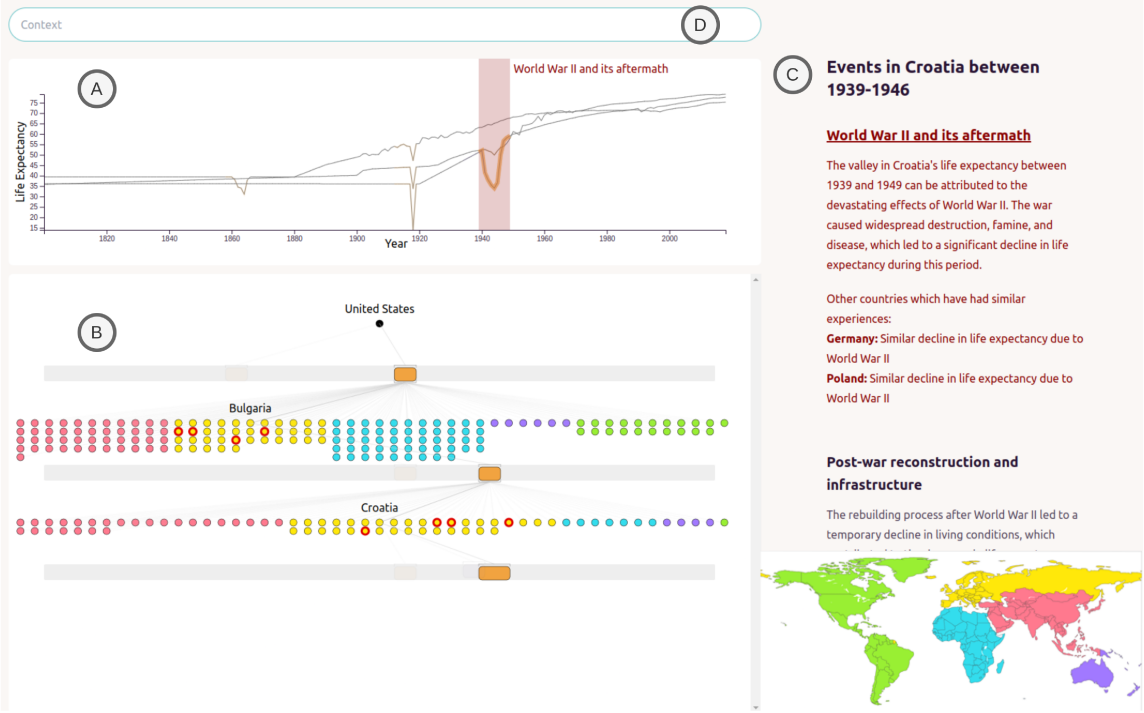


Figure 4.1: ChatKG is being used to investigate the life expectancy dataset. USA's valley between 1912-1924 shows a large number of other countries with patterns in the same time period. Of them, Bulgaria was selected. By selecting Bulgaria's right-most pattern, it was found that Croatia has two overlapping patterns with Bulgaria, and when hovering the patterns in ChatKG, chatGPT says WWII impacted both countries at that time.

THE line-chart visualizations of temporal data exemplified by *Q4EDA* in chapter 3 are not the only ones that can enable users to identify interesting patterns for the user to inquire about. Intelligent agents equipped with information retrieval

[1] This chapter was based on *Christino, L., & Paulovich, F. V. (2023). ChatKG: Visualizing Temporal Patterns as Knowledge Graph. Published in Eurographics Association*, which was then invited for extension, leading it to be submitted to a Special Issue of Computer & Graphics in Nov 2023.

[2] Demonstration Video: `https://www.youtube.com/watch?v=-InwIXbE86s`

capabilities have recently been shown to significantly improve the pursuit of knowledge (chapter 1), even after considering its limitations. The second contribution of this dissertation expands on the ground-work of *Q4EDA* by using oracles, such as chat AIs, as part of a Visual Analytic tool to automatically uncover *explicit knowledge* related information to said patterns. This approach entitled *ChatKG* proposes a novel visualization strategy that visualizes the structure of a Knowledge Graph which encodes the relationship between a dataset of temporal sequences, the patterns found in each sequence, the temporal overlap between patterns, and related *explicit knowledge* from an intelligent agent to each given pattern.

4.1 Overview

Line-chart visualizations of temporal data enable users to identify interesting patterns for the user to inquire about. Using Intelligent Agents (*IA*), Visual Analytic tools can automatically uncover *explicit knowledge* related information to said patterns. Yet, visualizing the association of data, patterns, and knowledge is not straightforward. In this chapter, I present *ChatKG*, a novel visual analytics strategy that allows exploratory data analysis of a Knowledge Graph that associates temporal sequences, the patterns found in each sequence, the temporal overlap between patterns, the related knowledge of each given pattern gathered from a multi-agent *IA*, and the *IA*'s suggestions of related datasets for further analysis visualized as annotations. I exemplify and informally evaluate ChatKG by analyzing the world's life expectancy. For this, I implement an oracle that automatically extracts relevant or interesting patterns, populates the Knowledge Graph to be visualized, and, during user interaction, inquires the multi-agent *IA* for related information and suggests related datasets to be displayed as visual annotations. Tests and an interview conducted showed that ChatKG is well suited for temporal analysis of temporal patterns and their related knowledge when applied to history studies.

4.2 Introduction

An underlying presumption in most Visual Analytics (*VA*) tools is that users can collect new knowledge through interactivity with data and visualizations [193].

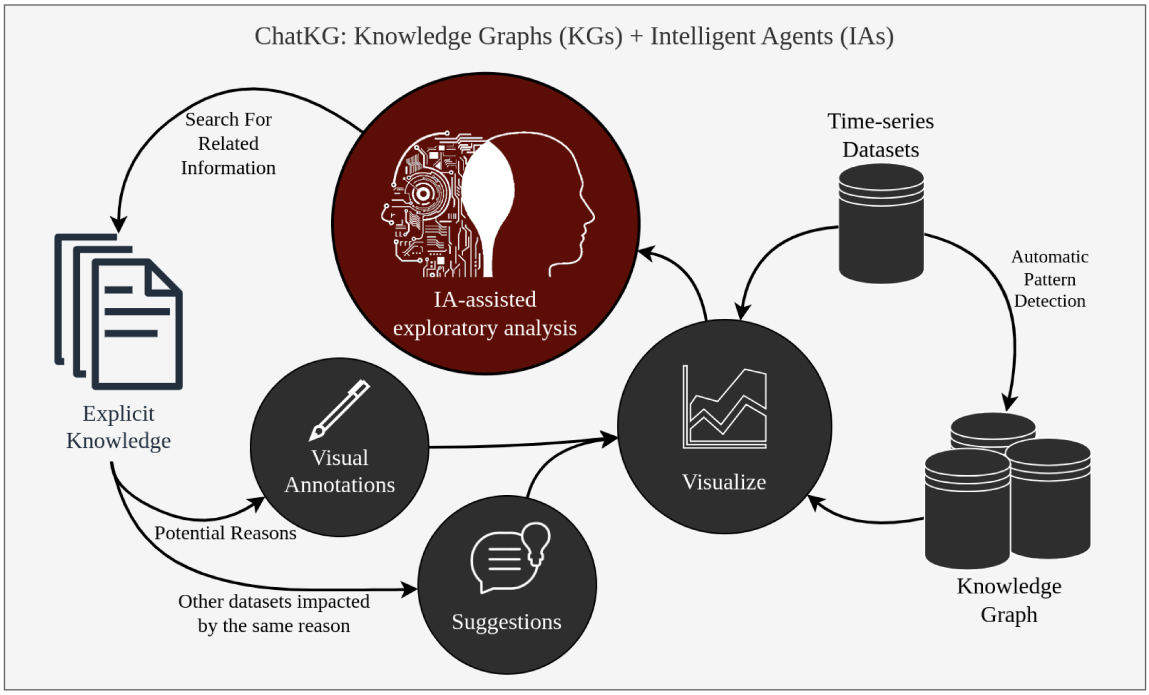


Figure 4.2: ChatKG Overview. The user and Intelligent Agent (*IA*) share responsibilities in the exploratory analysis. A temporal dataset is processed into a Knowledge Graph for the user to visualize. Patterns of interest are sent to the *IA* to fetch related information, which is used to annotate the visualization and to add suggestions of other available datasets to which the information retrieved has also had an impact.

One instance is how line-chart visualizations of time-series data can tell whether a particular variable is trending upwards or downwards or if there are any interesting patterns for the user to inquire about. Despite their popularity as data democratization tools, in the absence of an oracle with the knowledge to explain the visible patterns, like Hans Rosling for GapMinder [186], it is challenging to extract from line charts anything beyond the existence of patterns. This concept of using external sources of information has already been explored, mostly using information retrieval strategies to fetch information related to visual patterns [31, 219, 66], which is then overlayed as annotations on top of visualizations [59, 214, 140, 31]. Despite the relative success in addressing the problem of helping understand data patterns, the usual output of information retrieval engines is an ordered list of relevant web pages, which does not have a straightforward way to be summarized and displayed to the user as part of a *VA* tool (see chapter 3). On the other hand, with the advent of Intelligent Agents (*IA*s), such as chat AIs, users can now inquire

with natural language for related information about patterns found in data. Indeed, the *explicit knowledge* retrieved from chat AIs can potentially aid users by providing new and relevant information during data analysis [144]. In this context, *IA*s like chatGPT [169] represent a step further. Unlike usual databases, *IA*s accept a more flexible input and output through natural language and a novel reasoning engine that is well positioned to not only retrieve information but to reason and argue about it [19]. Recent developments have also allowed *IA*s to access compute resources to execute code, read files, and search the web. Indeed, *IA*s are able to use information retrieval engines to enhance their own capabilities [246] and to contextualize information. *IA*s can also handle complex or multi-step tasks using a multi-agent environment [248], where individual specialized *IA*s communicate to establish a plan of action and cooperate to achieve an arguably better output. This way, data analysis is a task within reach to be semi or fully automated by *IA*s [104].

Yet, when considering the use of *IA* usage within a *VA* tool, several challenges appear. Since the information retrieval engine uses a formal query and returns a structured output, the use of such engines has so far been favored in *VA* when compared to *IA*s, which use natural language. Due to this, *VA* tools can generally retrieve and display information out of patterns found in data, but more advanced abilities are still beyond the available techniques. For instance, *IA*s can retrieve information from within its vast model, which includes much of the information available online [179]. *IA*s can summarize information and reinterpret it in light of the pattern found in the data [104]. *IA*s can also discuss any conclusions reached due to the information and contextualize the conclusions based on specific instructions, such as relating the results to another completely different subject or rephrasing the conclusions to remove technical terms. Although a single *IA* has known issues with hallucinations [19], a multi-agent environment access to coding has shown to lessen this issue [248] by providing references as evidence for the veracity of the conclusions. Finally, *IA*s can extend their scope by ingesting other datasets beyond the initial one and suggesting further avenues for research. Yet, *VA* tools have not yet bridged the gap of how to collect data from a dataset and related information from *IA*s and use them in conjunction as a means to aid in exploratory

data analysis through visual metaphors and, therefore, provide contextualized reasons and suggestions for the existence of detected patterns in the dataset.

In this chapter, I present *ChatKG*, a novel visual analytics strategy that allows exploratory data analysis of the multi-modal task that associates temporal sequences, the patterns found in each sequence, the temporal overlap between patterns, the related knowledge to each given pattern gathered from a multi-agent *IA*, and the *IA*'s suggestions of related datasets for further analysis visualized as annotations. ChatKG is proposed as a way to visualize a cooperative environment between a Knowledge Graph (*KG*) that comprise of a time-series dataset and the patterns found within, and an Intelligent Agent (*IA*) which can retrieve and interpret contextualized knowledge about the patterns, which includes visual annotations of related information and suggestions for further research, as is exemplified by Figure 4.2. That is, the *KG*&*IA* intersection serves as a general knowledge repository and a connective layer between the temporal data, their patterns, and the contextualized *explicit knowledge*. I also exemplify ChatKG as a *VA* tool. For this, an oracle automatically extracts relevant or interesting patterns, such as *valleys* and *peaks*, from a time-series dataset and formulates natural language prompts to retrieve related information to each pattern. The data is structured as a *KG*, which populates ChatKG. Then ChatKG uses a multi-agent *IA* environment, which allows it to retrieve information from its model and execute code to search the web for up-to-date information that can be cited, referenced, reasoned, and summarized as visual annotations. With it, users can explore the temporal sequence, its patterns, and the *explicit knowledge* collected from the *IA*. ChatKG also enables users to contextualize their analysis through prompt customization and analyze the temporal overlaps between the patterns among other temporal sequences. I demonstrate ChatKG by reproducing GapMinder's use case of world life expectancy analysis [186]. I informally evaluated ChatKG with students and staff in my lab, confirming that ChatKG achieved its goals. I also interviewed a historian and verified that ChatKG is well suited for the analysis when applied to world history data, given that the historian can contextualize the *IA* to their research goal.

In summary, the main contributions are:

- Modeling the association of *explicit knowledge* from an intelligent agent to temporal patterns of line charts as a *KG*;
- Visualization of KGs with multi-modal data: temporal sequences, categories, contextualized text, and suggested content from the intelligent agent.

The remainder of this chapter is structured as follows. In Section 6.3, I discuss the related work and the limitations of the current literature in collecting patterns of temporal data, knowledge graph modeling, and *IA*s. From the limitation of how to allow *VA* to utilize *IA*s in exploratory data analysis of time-series data, I present in Section 4.4 ChatKG and in Section 4.5 I discuss use cases demonstrating and exemplifying how the interoperability of *VA* and *IA* through visual metaphors provide an interactive way for users to explore data in a cooperative environment with a multi-agent *IA*, not being required to type questions and visualizing the results as both natural languages and visual annotations. In Section 4.6, I discuss the limitations of my approach and the required improvements necessary for the broader use of *IA*s in *VA*, and in Section 4.7, I discuss the conclusions.

4.3 Related Works

Information Retrieval is a vast field of research dealing with the general problem of obtaining resources relevant to a piece of information. Among the different research directions in information retrieval to improve the typical query, strategies around how to generate and expand queries [58] so the retrieved information is more relevant to users [260, 39, 14, 64] and how to provide suggestions for future queries [168] had received considerable attention. Beyond the typical queries, it is also possible to retrieve information "by image" [182] to find images similar to a given image or "by natural language" [34, 120] using descriptive texts. Significant efforts have been made in visual analytics to aid users in allowing for non-obstructive exploratory analysis of information retrieval [211, 130]. Among the most relevant is NorthStar [130], a system that provides an exploratory system that follows responsive and real-time methods to display an intuitive visualization interface while at the same time automatically performing statistic operations to reduce the amount of potential bias or incorrect insights, and TimeSearcher [110]

which is a visualization tool which proposes various forms to perform visual queries in time-series data. Another relevant tool is Q4EDA (see chapter 3), which proposes a framework that converts time-series patterns selected by users to relevant search queries to be used in engines like Google or Wikipedia. Yet, the detection of patterns of interest is still expected to be done by the user. The inevitable comparison of these approaches to Intelligent Agents (*IA*) like chat AIs, which use a natural-language approach to information retrieval, has shown that the *IA*s are more naturally able to cope with complex or non-uniform queries [3] due to their capability to process and understand natural language input. With ChatKG, I propose that detecting visual patterns and requesting information from *IA*s is more valuable for the user exploratory task due to its natural language capabilities than related works' approach.

Definition 4.3.1 (Intelligent Agent *IA*)**.** *A program that can make decisions or perform actions similar to a human in the context of a human-computer interaction system. Chat AIs, like chatGPT [169], are examples of IAs.*

Definition 4.3.2 (Information)**.** *A concept, fact, or circumstance that of which, by being understood, has value to oneself.*

Another aspect of information retrieval is the auto-discovery of patterns or insights. QuickInsights [66] proposes a way to quantify the "interestigness" of a visualization, wherein users might find it interesting to analyze. On the other hand, Tang et al. [219] propose a scoring system to systematically identify which insights or patterns will be most interesting for users. Although I do not present a novel way of auto-identifying interesting patterns, ChatKG does so by focusing on peaks and valleys of a temporal sequence, as opposed to the multi-varied approach of [219] or the comparison between visualizations of QuickInsights [66]. ChatKG aims to utilize this concept to auto-detect patterns of interest on the user's behalf and, from them, request related information from *IA*s.

The use of *IA*s has been taking up interest [22, 68]. Text-based techniques where a *VA* tool provides a question-answer interface to interface users with visualizations have shown to be effective [127, 119, 257]. The work by El-Assady, M. and Moruzzi, C. [68], for instance, argues that a *VA* tool can be a form of interface between an *IA*

and a user. *IA*s, like ChatGPT [169], have already shown great impact when used in cooperation with humans. Recently, Chat2Vis [149] proposed auto-generating visualizations based on chat input. From analysis [104] and explanations [215], *IA*s are impacting many fields. Yet, since the advances of *IA* are very recent, not many successful usages have been shown in Visualization and Visual Analytics methods. In this way, ChatKG differs in related work by using the *IA* to extract the *explicit knowledge* encoded in its Large Language Model (LLM) related to a specific temporal dataset's pattern and use the result to provide a visual metaphor which includes the explanation and suggestion of the detected patterns.

The advances of Intelligent Agents (*IA*s) have also caused a large impact in recent information retrieval methods [5, 246]. *IA*s are already capable of searching and exploring existing literature given a prompt through retrieval augmented generation (RAG) [248], generate code by allowing it to have access to compute resources [215], reason using the collected information through its large language model [19], and contextualize the answer based on specific user requirements [85]. At first, information retrieval may seem straightforward for *IA*s because it only requires the *IA* to search for information already available, but several practical difficulties may arise [13]. For instance, there may be controversies about the information retrieved, requiring the *IA* to judge the veracity or reliability of the information. Another issue is the concept of hallucinations, where agents with no access to data can produce unfactual results and wrongly express a high degree of confidence about the veracity of the results [19]. Using multiple agents to coordinate a collaborative effort while separating the responsibilities of fact-checking and code execution to solve such complex issues was shown to be successful by AutoGen [248]. AutoGen utilizes multiple agents with specialized tasks in different aspects of the problem work, and by allowing them to cooperate, they can provide better output compared to single agents [248]. This concept has rarely been applied to Visual Analytics, and with ChatKG, I aim to expand the capabilities of its information retrieval through a multi-agent *IA*. This way, ChatKG is able not just to retrieve information but can search for citations on the web, verify if the information retrieval is of good quality, and contextualize the output to answer specific questions the user may have.

Although a certain level of automated information retrieval is done through existing tools and techniques [246, 91], the information retrieved is presented to users without any modeling strategy. In visual analytics, *Knowledge modeling* defines a workflow where user interaction leads to knowledge generation [193, 72]. In this process, all information retrieved from tools like Q4EDA and NorthStar [130] is categorized as *explicit knowledge*. Once the user discovers new insights, it becomes a *tactic knowledge*, which refers to knowledge encoded as experience on the user's part. With this taxonomy in mind, although these information retrieval tools retrieve and display information catered to user queries, they do not model the collected information as a knowledge-centric structure. ChatKG bridges this gap through Knowledge Graphs (*KG*).

On the other hand, the use of knowledge graphs (KGs) in Visual Analytics is usually limited to a database, but not to a visual metaphor [241, 45, 43, 175]. KG4Vis [136], for instance, uses a knowledge graph structure to provide visualization recommendations but does not express the knowledge graph visually. CAVA [40] and ExeKG [262] are some examples of systems that enable the exploration and analysis of KGs through visual interaction. Indeed, the number of works using KGs for visualization or analysis purposes has been slowly increasing, many of which say that KGs are suitable for knowledge analysis and extraction [137]. Yet, these works focus on visualizing an existing *KG* instead of modeling and visualizing one that they themselves have modeled and populated. ChatKG is novel in this space since it models and stores the information retrieved about patterns and proposes a novel visualization strategy to visualize the *KG* for exploratory time series analysis.

Considering the perspective of pattern detection through visual selection, some approaches focus on using the pattern to extract or generate automatic visualizations and annotations. For instance, *VA* has attempted to process a given dataset and include external information related to the dataset in question as annotations [115, 133]. Some *VA* tools generate new visual metaphors from text [59, 31, 115, 133]. Others fetch existing visualizations from the web to be included as part of the system [140]. Despite representing significant progress, the amount of analysis enhancement they provide is limited to the generation of visual

metaphors [59, 140, 31] and of visual feature extractions [31, 219, 66] produced from a single dataset used for analysis. ChatKG is inspired by such works, yet overlaying ChatKG is novel in its proposal to generate annotations from the *IA*s results. ChatKG relegates the *IA* knowledge from what would be a large text annotation to a small reference annotation possible by *IA* summarization [146, 239] and a separate text panel containing additional information.

4.4 ChatKG

The ChatKG method consists of two main components: a Knowledge Graph (*KG*) that connects temporal visual patterns to *explicit knowledge* from an Intelligent Agent (*IA*) and ChatKG to visualize it. The ChatKG knowledge graph uses an oracle, which scans a time-series dataset for all patterns, such as valleys and peaks, from within. These patterns are used to model and populate a knowledge graph, which is then displayed to the user as a line chart and a graph diagram. The ChatKG *IA* is triggered once the user wishes to investigate potential reasons for a pattern, and its results are shown as text and included in the visualizations as annotations. The schematic of this workflow is shown in Figure 4.3. In this section, I describe the modeling process taken to generate the *KG*, the visualization strategy to display the *KG* for exploratory analysis, and the use of *IA*s to retrieve related information on patterns found in the data. In the following sections, I discuss the design and implementation of ChatKG and its example as a *VA* interface.

4.4.1 Connective Knowledge Graph Generation

I aim to allow users to explore knowledge in the form of textual data through visualizations and visual findings through textual explanations. This is possible by an associative *KG* that links patterns found in visual representations of time-series data and *explicit knowledge* that holds relevant information about the said pattern. To populate this *KG*, three main steps are taken: pattern detection/extraction, automated retrieval of *explicit knowledge* from an *IA*, and modeling a *KG ontology* that relates the time-series datasets, patterns, and the *IA*'s *explicit knowledge*.

In summary, by providing the method with a time-series dataset, such as one from UNData [161], it interprets the data as line-chart visualizations of the datasets

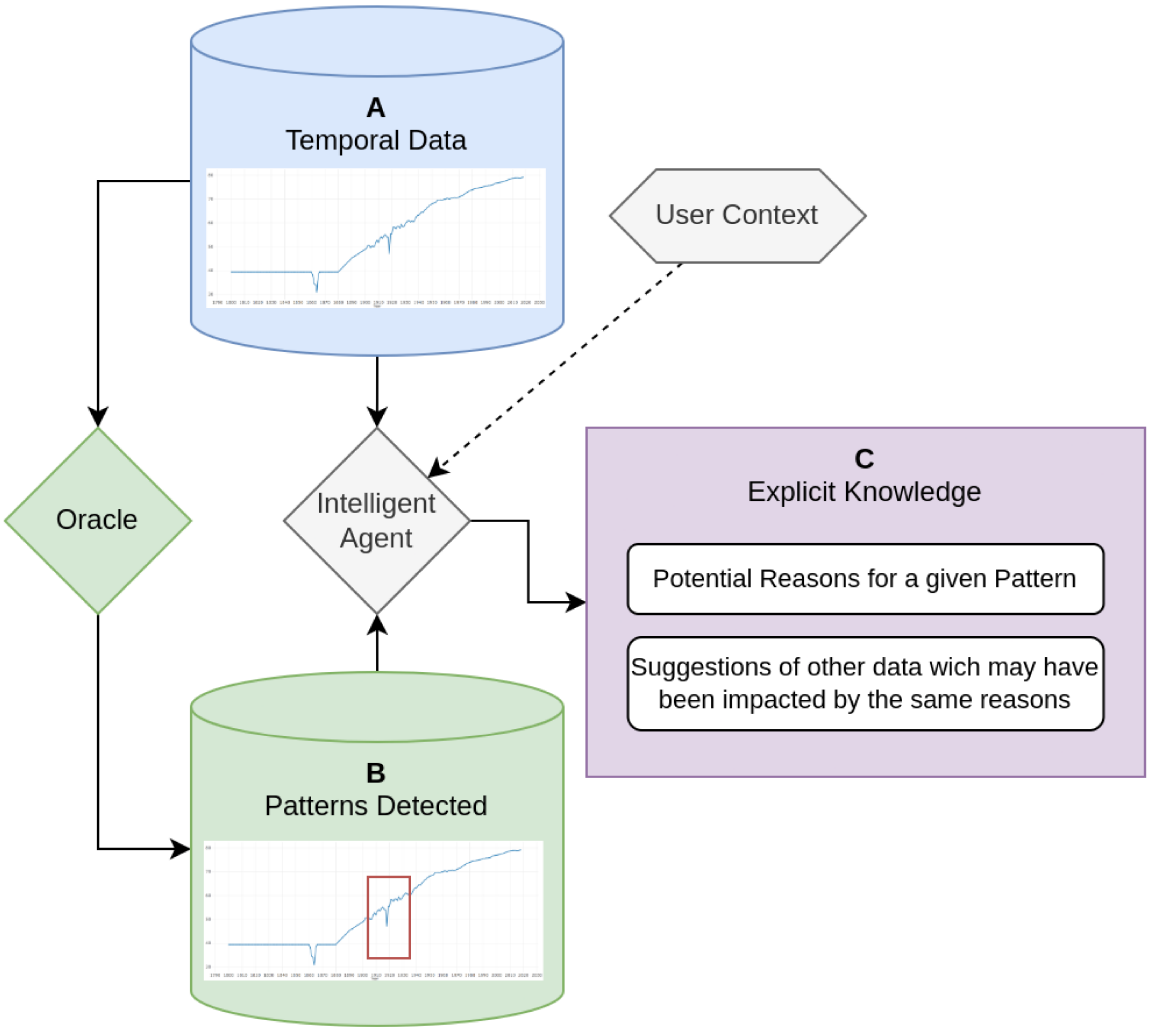


Figure 4.3: ChatKG schematic. Given a time-series dataset (A), an oracle detects all patterns from within the data (B). If a user wishes to investigate the potential reasons for a given pattern, an ChatKG's Intelligent Agent (*IA*) uses the information to generate an *explicit knowledge* structure (C), which includes potential reasons for the pattern and suggestions of other data that the same reason may have also impacted.

and runs a human-like oracle that auto-discovers potential *patterns* throughout the data. First, Each pattern generates a unique question for the *IA*, which returns some *explicit knowledge* in the form of a text, which is recorded. Then, a connective layer is modeled as a *KG* to associate the dataset, the patterns, and the *explicit knowledge* from the *IA*. Finally, the resulting *KG* is visualized by ChatKG where users can explore the time-series dataset, its patterns, and the associated *explicit knowledge*.

The standard practice to design a *KG* is to define its classes and relationships through the Web Ontology Language (OWL) [8]. In the method, the classes used are: *Dataset*, which defines the time-series dataset being analyzed; *Pattern*, defining any pattern that was found; *TimeSpan*, representing any range of time (e.g., years between 1800 and 1900); *Time*, defining a singular time entry (e.g., the year of 1800); and *Knowledge*, defining any *explicit knowledge* collected. The relationships of ChatKG's model are: *dataset-pattern* associates a dataset to any pattern found within it; *pattern-timespan* associates the timespan of a given pattern; *timespan-time* associates timespans to its related time entries; and *pattern-knowledge* associates any *explicit knowledge* related to a given pattern. The *ontology* described is shown in Figure 4.4. Namely, the *ontology* centers around a *Pattern* class, which identifies which dataset it came from, the time span of the pattern, and what knowledge was extracted from the *IA* regarding the pattern. This way, ChatKG is able to query the *KG*, for instance, to check other datasets with similar patterns due to the same or overlapping time spans.

The first step to populate the *KG* is selecting a time-series dataset. This dataset must contain a continuous value that changes over time. Datasets from UN-Data [161] are great examples. A reference to the datasets is recorded into the *KG* as nodes of type *Dataset*. With the data in hand, ChatKG runs an oracle to extract all findings from the data to populate the *KG*. The implemented oracle is based on the method described in Q4EDA, which itself is inspired by prior work [124]. Q4EDA discusses two types of visual patterns. Of them, patterns related to trends, such as the overall increase or decrease of life expectancy (see UNData [161]), were shown to retrieve inconsistent related information from Wikipedia. Therefore, I limited the extraction of patterns to "peaks" and "valleys", since they were shown

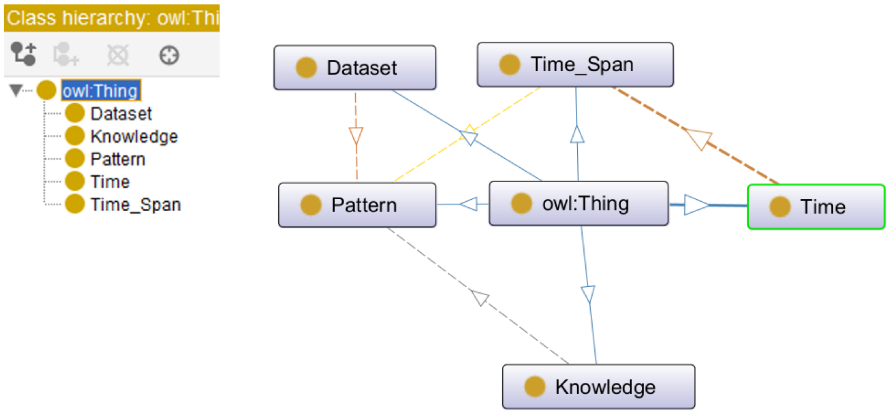


Figure 4.4: Design of ChatKG's Knowledge Graph following the Web Ontology Language [8]. It defines the classes and their relationships that will be visualized by ChatKG.

to be more informative [199].

The oracle then use a peak/valley strategy to detect all peaks and valleys throughout the data. This is done by comparing neighboring values of the time-series dataset [240, 252]:

$$P = \sum_{k} W(k).Pr(k), k \in S \tag{4.1}$$

In Eq. 4.1, we see that for a given time-series S, every sub-section of the time-series $k \in S$ is analyzed using its width W and its prominence or absolute height Pr. With this, I calculate the probability P of whether a given sub-section $k \in S$ is a peak. The same method is used to estimate a valley probability with negated k as in Eq. 4.2:

$$k_v = -k, k \in S \tag{4.2}$$

Finally, a threshold is used to define whether the time-series S is either a pattern (peak or valley). This pattern detector is run over all time-series datasets to extract a list of all potential pattern findings from the data. Each pattern is recorded in the *KG* as a *Pattern* and is associated with its related sub-section $k \in S$, which is recorded as a *TimeSpan*. The *Pattern* node is also associated with its related *Dataset*. Note that all these variables are set based on a case-by-case basis as is exemplified

at Section 4.5. Finally, I go through all patterns extracted and query an *IA* through a templated prompt query, which is exemplified in Section 4.5, and record this query as the *Knowledge* related to each *Pattern*.

4.4.2 Visualization

ChatKG is the proposed visualization that allows exploring the *KG* defined above by visually encoding the associations of time-series sequences, the patterns, the time-series overlap between patterns, and the *IA*'s *explicit knowledge* of each pattern. In its essence, this visualization is designed to be exploratory. Therefore, it is designed to thrive on user interactivity. By discussing with potential users, the design requirements found were:

R1: Given a dataset, display the patterns for visualization;

R2: Given a pattern and an optionally defined context, display the *explicit knowledge* from an *IA* which can be interacted with;

R3: Given a pattern, display other datasets which contain similar patterns;

By evaluating potential existing visualizations for each requirement, I define that the visualization of line charts and text containers would be suitable for displaying the time-series data and the *IA*'s *explicit knowledge* as goals. Yet, developing a visual metaphor that displays the relationship of patterns, datasets, time, context, and knowledge proved to be less straightforward.

Here, I propose a novel exploratory-focused visualization called *ChatKG*, which displays the relationships of the *KG* as links between elements and is exemplified in Figure 4.5. Considering the *KG* design of Figure 4.4, ChatKG displays an interactive unraveling of the graph in the y-axis. That is, by starting with a user-selected dataset, ChatKG displays the relationship dataset-pattern and pattern-timespan as a bar-visualization (Figure 4.5), from which users can interact by selecting a pattern-timestamp pair to display all related datasets, which are represented as square packing circles and, optionally, colored according to a categorical value. In the example of Figure 4.5, each circle represents a country colored by its continent, following the example provided by GapMinder. The sequential exploration and clear exposition of the user's exploration path in the y-axis have enabled users to

easily backtrack their exploration behavior through mouse hovering to understand better and remember their findings [194].

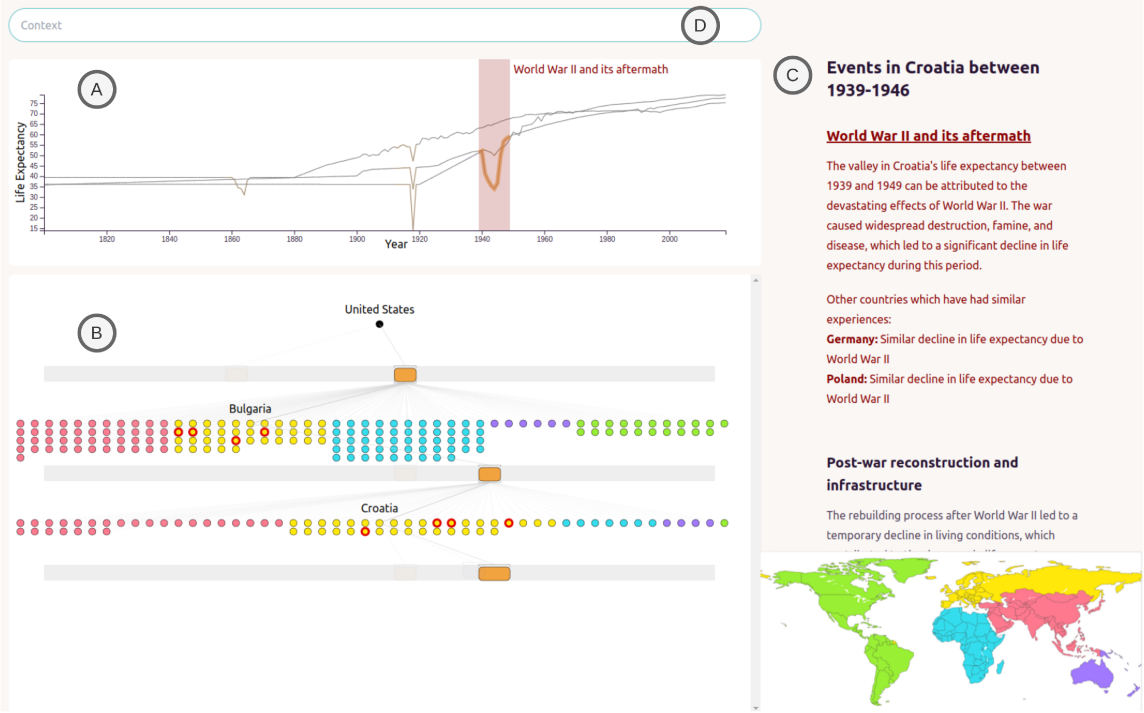


Figure 4.5: Example of ChatKG being used to investigate the life expectancy dataset. The line chart visualization (A) displays the time series of three selected countries: USA, Bulgaria, and Croatia. USA's valley between 1912 – 1924 shows a large number of other countries with patterns in the same period. Of them, Bulgaria was selected in the graph diagram (B). By selecting Bulgaria's right-most pattern, it was found that Croatia has two overlapping patterns with Bulgaria, and when hovering the patterns in ChatKG, the *IA* displays its text. The description of potential reasons (C) says WWII impacted Croatia's life expectancy, and related countries like Germany and Poland were similarly affected due to the same reason. In (D) the user is able to define a context for the query, which in this example is empty.

To design ChatKG, I informally tested different visualization options with users from my research lab. First, I attempted to use a network graph visualization through cystoscope.js [207], but the resulting node-link diagram did not provide a good visual metaphor for the patterns and time spans. I considered using glyphs and other icons, but using nodes in a graph as the visual metaphor of time spans was shown to be lacking by showing to and collecting feedback from others. Therefore, I considered a Gantt chart to display the pattern's time spans

visually, but its structure did not allow the user to visualize and select the datasets related to each pattern (*R3*). Therefore, the devised visualization overlaps a node-based diagram, which displays categorical data and the relationships between the category and a given pattern, and a Gantt chart visualization to display the occurrences of patterns within a given dataset.

Although the visualization discussed so far can be used by itself, ChatKG was also designed as a way to link a line-chart visualization (Figure 4.5) of the time-series datasets and a text container that displays the collected knowledge from the *IA*. Indeed, I exemplify ChatKG by using it as part of a *VA* tool composed of three main visualizations as is shown in Figure 4.5: (A) a line chart that displays the raw time-series data of all datasets and each of the patterns found in them; (B) the ChatKG visualization of the time-series datasets and extract patterns; and (C) a text display of the *explicit knowledge* extracted from the *IA* given a pattern being hovered by the mouse.

To best analyze and explore the patterns (R2 and R3), ChatKG's bar visualization is aligned to the line chart in its x-axis. Users choose which datasets they want to analyze by selecting in ChatKG, which causes the line chart to display the time-series data and the detected patterns of the selected datasets. Also, hovering a pattern-timespan in any bar visualization will cause the related knowledge text from the *IA* to be displayed to the right. Finally, mouse hover displays all visual elements related to what is being hovered. For instance, by hovering a pattern in the line chart, the related pattern-timespan is highlighted, and by hovering any element of ChatKG, such as a dataset or a pattern-timespan, ChatKG highlights all elements and links between the hovered element until the original dataset, providing the map of how the user's exploration got to that element. Optionally, the dataset can be grouped and color-coded by some metadata, such as how the datasets from the example of Figure 4.5 are categorized by countries, which are then grouped by and colored by continent following the dataset of [131].

4.4.3 Knowledge - Intelligent Agent (IA)

In every user interaction, ChatKG is tasked with retrieving related information to a given pattern. The knowledge repository I use in this work uses LMStudio [69]

and AutoGen [248], allowing ChatKG to utilize most of the state-of-the-art chat *IA*s available. I exemplify ChatKG with Mistral Openorca 7B, which has demonstrated exceptional capability to uncover potential reasons why certain patterns are seen in time-series data despite many limitations of potentially producing incorrect information.

The ChatKG presented so far can aid users in single-task analysis where their questions were strictly related to data contained within the time-series data. For instance, when presented with life expectancy data (see UNData [161]), users can discover potential reasons for changes in life expectancy. However, the *explicit knowledge* shown is a static piece of text that came directly from the *IA*. This means that users cannot ask for more information, have no access to references or citations of where this information came from, and cannot relate the collected information back to the visualizations. Additionally, preliminary tests with domain experts also showed that limiting the analysis to specific prompts caused the outputs from the *IA* to be too generic to be valuable. For instance, when ChatKG is asked for related information on the decrease in life expectancy, results usually describe wars, diseases, and other similar causes. Although they said this was a good start, it was not enough to allow the user to focus or contextualize the *IA* to describe the pattern under a specific concern. A specific example from one of the users was that they wanted to contextualize the answers to discuss whether or not it impacted University attendance.

Therefore, I expanded on the prior knowledge extraction methods through multi-agents [248], which allows the *IA* to not only depend on the data within itself but also to search the web for up-to-date information that can be cited and referenced. Of course, for this, I are required to assume that *IA* and, consequently, the internet have information about the time-series patterns in question. The schematic of Figure 4.6 describes the flow of information, which shows that the pattern detection (see Section 4.4.1) is used to generate a prompt to the *IA* group.

The ChatKG *IA* is a chat-based interface configuring a chat AI, such as chatGPT, to perform tasks to enhance all three requirements from Section 4.4.2. Given a pattern, ChatKG asks three things to its *IA*: *explicit knowledge* that may potentially explain the pattern (R2), other datasets that may be of interest to be analyzed (R3),

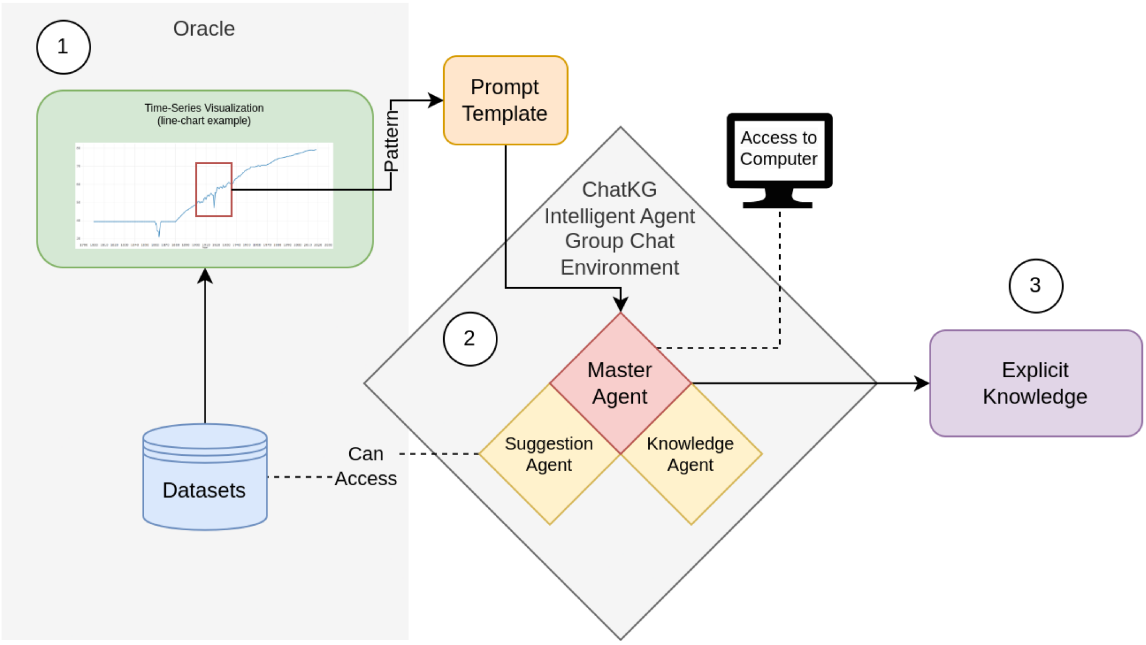


Figure 4.6: ChatKG's Intelligent Agent (*IA*) schematic. A prompt is generated following a prompt template from the patterns detected using the Oracle (1). The *IA* chatgroup (2) consists of three agents: a knowledge agent that does reasoning, a suggestion agent with access to the data, and a master agent that coordinates the group. The output (3) is an *explicit knowledge* property formatted for use in ChatKG.

and summarized annotations to be added to the visualization to better understand or visualize the knowledge extracted (R1). Finally, to answer these questions using an *IA*, I propose creating a multi-agent environment through AutoGen [248] with three agents. To design each agent, I are required to specify their abilities, such as code-generation, image-generation, reasoning, web-search, and others. This is done by defining the type of agent for each agent, the large-language-model model being used, the model parameters, the system prompt that tailors the agent to a specific type of task, and the user prompt that indicates what needs to be done. Let us describe each agent individually and then discuss how they cooperate.

The *knowledge agents* focuses on collecting knowledge. As part of its responsibility, it is tasked not only to propose potential explanations of a given pattern but also to find sources on the internet to be used as references. This way, the *explicit knowledge* shown in ChatKG will have a target for scrutiny if the user wishes to verify its veracity and credibility.

```
knowledge_agent = AssistantAgent(
    name="reasoning",
    llm_config=llm_config,
    max_consecutive_auto_reply=10,
    system_message="You are a helpful assistant. You use your knowledge
    skills to solve tasks. Reply TERMINATE if the task has been
    solved to full satisfaction. Otherwise, reply CONTINUE, or the
    reason why the task is not solved yet.",
)
```

The *suggestion agent* focuses on collecting other potential datasets of interest. ChatKG uses the output of this agent to enhance the knowledge graph visualization by displaying which of the known related datasets have a high probability of being of interest. This means that this agent requires the context of which datasets are available in ChatKG, what was the *explicit knowledge* found by the knowledge agent, and whether or not the available datasets have any information mentioned by the *explicit knowledge*. Therefore, the suggestion agent needs the ability to do retrieval augmented generation (RAG) [248], which allows the agent to retrieve information

from within a file. The example of Figure 4.10 shows how some countries related to the pattern selected are suggested to be investigated. The suggestion agent is defined as:

```
suggestion_agent = RetrieveUserProxyAgent(
    name="ragproxyagent",
    retrieve_config={
        "task": "qa",
        "docs_path": "../datasets",
    },
)
"""
```

The knowledge and suggestion agents are put into a chat group [3] where a master agent receives the prompt and gives commands to the agents. The master agent is defined as:

```
master_agent = UserProxyAgent(
    name="user_proxy",
    human_input_mode="NEVER",
    is_termination_msg=termination_msg,
    code_execution_config=None,
    llm_config=llm_config,
    system_message="Reply TERMINATE if the task has been solved to full
    satisfaction.",
)
"""
```

The master agent is triggered through a prompt that includes all information about a given pattern. The output must follow a JSON structure used by ChatKG to process the results. The prompt follows this template:

[3] See https://github.com/microsoft/autogen/blob/main/notebook/agentchat_groupchat_RAG.ipynb

```
QUESTION: Give me reasons for the <valley or peak> in <dataset name>
between <pattern year range>.
<if there exists a context given by the user> Contextualize the answer
focusing on <any context being given by the user>
<end if>

PLAN OF ACTION:
    * Summarize the answers into a json structure.
    * For each answer in the json structure, include a small header, a larger
    description for each answer, a citation that discusses the answer further, and
    the year range that the answer applies to.
    * For each answer, include a list of similar countries that were impacted by
    the same reasons. Include the name of the suggested country, a reason why the
    country was suggested, the year range that the country was impacted, and a
    citation that discusses the country further.
    * Include at least three possibilities and three country suggestions for each.

ANSWER - The answer should follow this
example:
[
    {header: "Some header", description: "a longer description",
    date_range: {from: integer, to: integer},
    citation: "URL", similar: [
        {suggestion: "other suggested country",
        reason: "reason why this country was suggested",
        date_range: {from: date_begin_integer, to: date_end_integer},
        citation: "URL"},
        {suggestion: "other suggested country",
        reason: "reason why this country was suggested",
        date_range: {from: date_begin_integer, to: date_end_integer},
        citation: "URL"},
        ...
```

```
    ]},
    {header: "Some header", ...},
    ...
]
"""
```

By filling the templated prompt with the required information, ChatKG requests for *explicit knowledge* from the *IA*. The *explicit knowledge* returned describes a list of potential reasons for the pattern, including a small header, a larger description, and the date range in which that reason was known to happen. The result also includes other countries which potentially have had a similar experience. This information is used to populate the ChatKG explanation text panel (Figure 4.5[C]), and if the user hovers any of the answers, the line-chart visualization of ChatKG is then enhanced with a visual annotation of each potential reason by adding a highlight to the line chart with the hovered answer. The result is also used to highlight ChatKG's graph diagram to indicate which other countries may have also been impacted by similar circumstances. The example of Figure 4.10 shows how if the user hovers the first answer, an annotation of the hovered answer is shown in the line chart, and the related countries are highlighted in the graph diagram.

4.5 Use Case: Reasoning Life Expectancy Fluctuation

Inspired by GapMinder [186], I implemented one main use-case to explore the world's life expectancy fluctuation. In this example, the datasets are temporal sequences of a specific country's life expectancy. Therefore, each *Dataset* node in the *KG* is a country, each *Time* is a year, and *TimeSpans* are measured in "span of years". I ran the pattern extraction algorithm, setting the algorithm threshold to $P > 1.5$ following Q4EDA (see chapter 3). Yet I noticed that the number of patterns was too big, causing the visualizations to become cluttered. Additionally, detecting all patterns sometimes returns the whole temporal dataset as a pattern and several patterns with too many overlaps.

To reduce the number of patterns to a more reasonable size, I defined empirically that patterns should have at least 3 years and at most 8 of time span, and if two

patterns with the same peak or valley were detected or if two patterns overlapped for more than two-thirds of their time-span, only the one with largest time-span was kept. With this, the number of patterns kept was 661. Although an adjustable parameter can regulate this, such values showed good results in the informal evaluation conducted.

Then, the Intelligent Agent (*IA*) was set up for the collection of *explicit knowledge*. For this, I configured an AutoGen [248] multi-agent environment following Section 4.4.3. I configured it to call LMStudio [69] configured with Mistral Openorca 7B since it gave the best results out of the ones tested. However, users can also use competing models from LMStudio or the GPT-4 model [169] though the OpenAI API.

User prompts for each agent were then created from each pattern following the template of Section 4.4.3. This prompt triggers the *IA* to generate the *explicit knowledge* to be displayed as a text to be shown to the user and to be used to annotate the visualizations with suggestions of other countries that may be related to the one in question and annotations that visually explain to the user the *explicit knowledge* collected.

The result of the combination of the *KG* populated with life expectancy pattern detections and the ChatKG *IA* can be seen in the example of Figure 4.5, which displays a visualization for users to explore each country's life expectancy. This visualization is given to the user as an interface between the user, *KG*, and *IA*. The line chart displays fluctuations in life expectancy and all patterns found for each country being investigated, and the graph visualization displays all other countries that have had overlapping patterns with the selected one. Additionally, the *explicit knowledge* retrieved from the ChatKG *IA* is displayed in the right panel. The *IA* output is comprised of a list of potential reasons for the existence of a given pattern. This output is parsed and the headers and descriptions of each reason are shown as text. The citation URL of each reason is made into a link which the user can visit if they click in the respective reason, and by hovering with the mouse over one of the reasons, the visual annotations are shown. In the example of Figure 4.5, the first reason is highlighted in red, and the corresponding suggestion is shown as a transparent red rectangle and red text header which describes the time-range

relative to that specific reason. Additionally, the related countries of the hovered reason are highlighted in the graph visualization. Finally, the user has access to a context text field to specify a context to be given to the ChatKG *IA*.

4.5.1 Usage Scenarios

I executed three usage scenarios within this setting to exemplify how ChatKG can be used and performed an expert interview. The first example describes an investigation where the user wants to discover potential reasons for two patterns found. The second example describes another user searching for similarities between different countries. The third introduces a context, where the user not only wants information about life expectancy, but wants ChatKG to contextualize the reasons for variations in life expectancy when compared to university attendance and history.

Specific Usage - United States Investigation: Justin, a fictional high school student, was asked to investigate the USA's life expectancy history. Therefore, by using ChatKG, he identified two main patterns of interest. By hovering the right-most pattern, he discovered that at the time, the USA was impacted by WW1 and infectious diseases, like Influenza and Tuberculosis (Figure 4.8), which might have caused the life expectancy disturbance he saw. By clicking on the pattern, he was shown that more than a hundred countries had some event that impacted their life expectancy in the same period. He did the same to the left-most pattern but discovered that that event seems to have been caused by the American Civil War (Figure 4.7), and although not many other countries had drastic life expectancy changes at that time, France and Russia were impacted by Wars in periods just before (Russia Crimean War 1853-1856) or after (French-Prussian War 1870-1871). Here, Justin was able to learn potential reasons why the USA's life expectancy was impacted and what other countries might have been similarly impacted.

In this example, Justin queried for the life expectancy of the USA, which involved a query to the *KG* and time-series dataset and the population of the visualizations. When Justing hovered over each pattern, the natural language prompt was fired following the prompt template of Section 4.4.3, and the three *IA* agents coordinated to retrieve potential reasons for the patterns to have occurred. The

knowledge agent provided reasons from its own model and generated code to search the web to confirm the reasons. The suggestion agent had the ability to read from the existing data and provide this information back to the knowledge model, which, in turn, added to the output the suggestions of which other countries were impacted by each of the reasons (e.g., Russia and France also impacted by Wars).

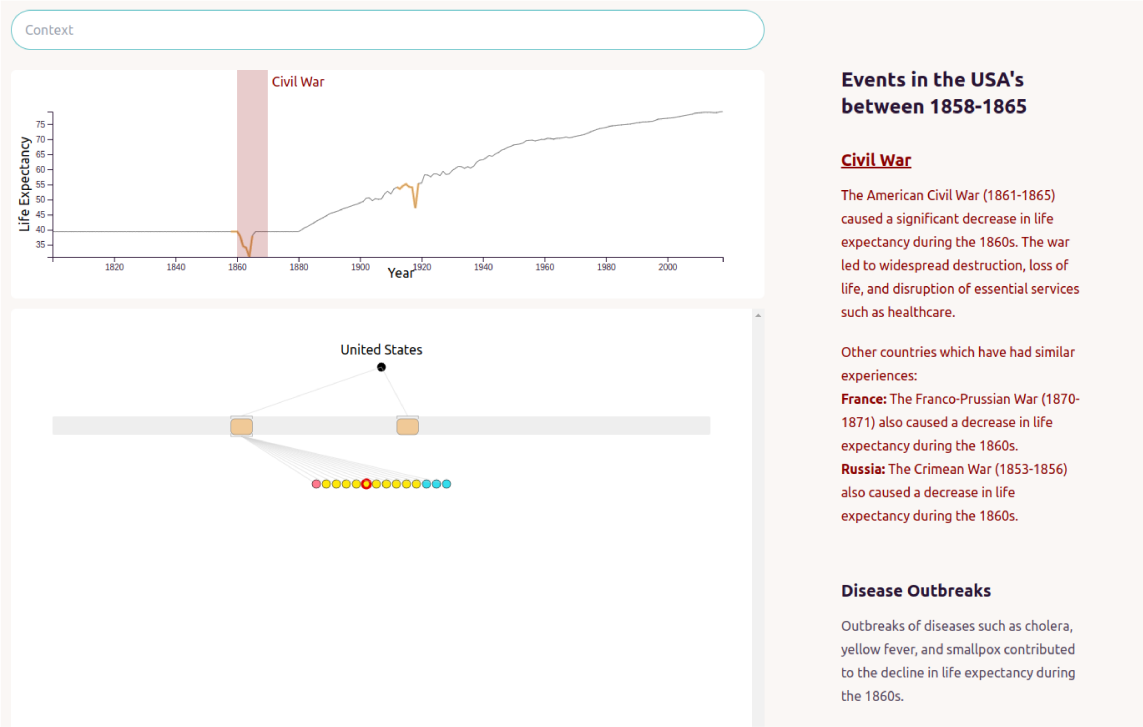


Figure 4.7: Example of ChatKG being used to investigate the life expectancy in the USA. The first valley from 1858-1868 was likely caused by the American Civil War, as per ChatGPT.

Open-ended Usage - Search for unexpected similarities: Angela, a fictional history major, wishes to explore the history of the world's life expectancy and see if there are any unexpected similarities between seemingly unrelated countries. She investigates Algeria and notices several unexpected patterns (Figure 4.9). Hovering the left-most pattern shows that, at the time, Algeria was not a proper country but a French colony, which makes her suspicious of how reliable the life expectancy value from UNData [161] is. By clicking on that pattern, Angela investigates all countries listed as related and sees that some may also have questionable data, such as Mauritius and Tunisia. She finds it interesting that similar patterns were found between countries that, at the time, should have been colonies and wonders

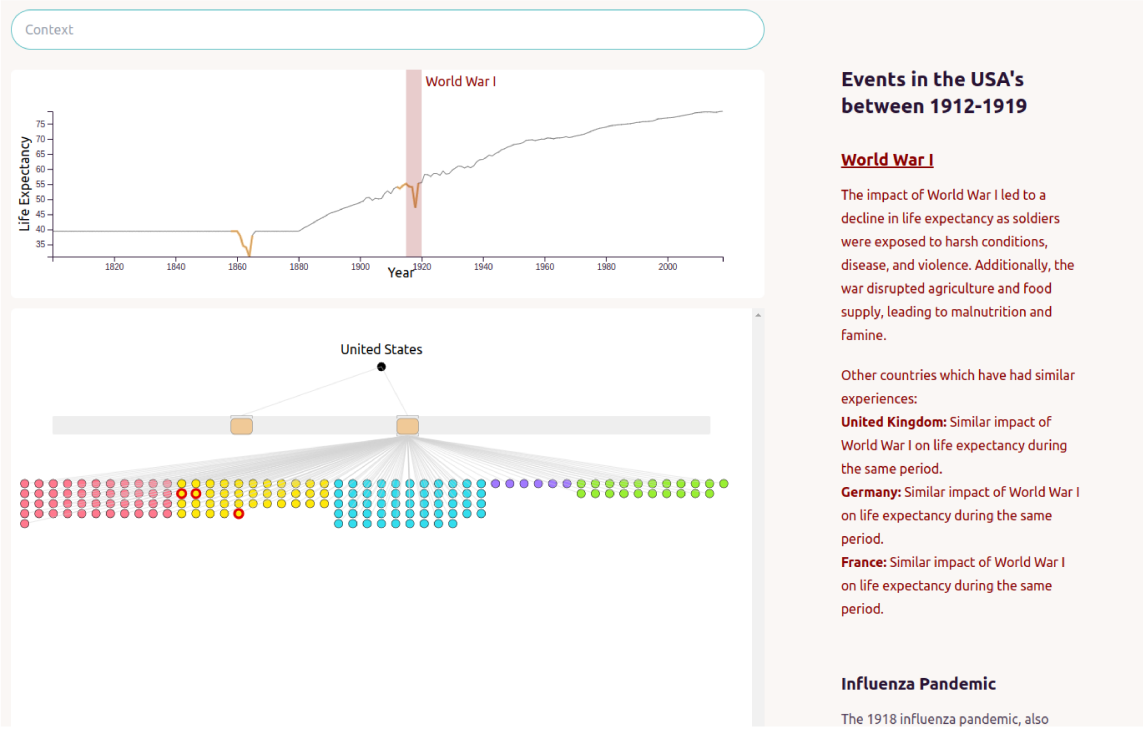


Figure 4.8: Example of ChatKG being used to investigate the life expectancy in the USA. The second valley from 1912-1924 was likely caused by World War I, Influenza, and other factors, as per ChatGPT. Additionally, from the number of circles in ChatKG I conclude that many countries have also had some impact on their life expectancy during the same period.

if the following Algerian pattern would have a different result. Here, Angela could explore the data and discover the unexpected, which prompts further data exploration.

In this example, Angela queried the *KG* multiple times due to her interactivity with the visualization. She focused less on the output of the *IA* and more on the reliability of the time-series data and the way the *KG* was modeled. However, in order to verify whether the data from UNData [161] is reliable, she used the outputs from ChatKG's *IA*, which told her the information needed for her to decide whether or not to consider the time-series data reliable. From here, Angela can also follow the citations given by the *IA* to verify how reliable all this information is.

Contextualized Usage - Relationship between life expectancy and university attendance: Karl, a fictional university student, wishes to investigate if there might be any relationship between events that impact the population's life expectancy and

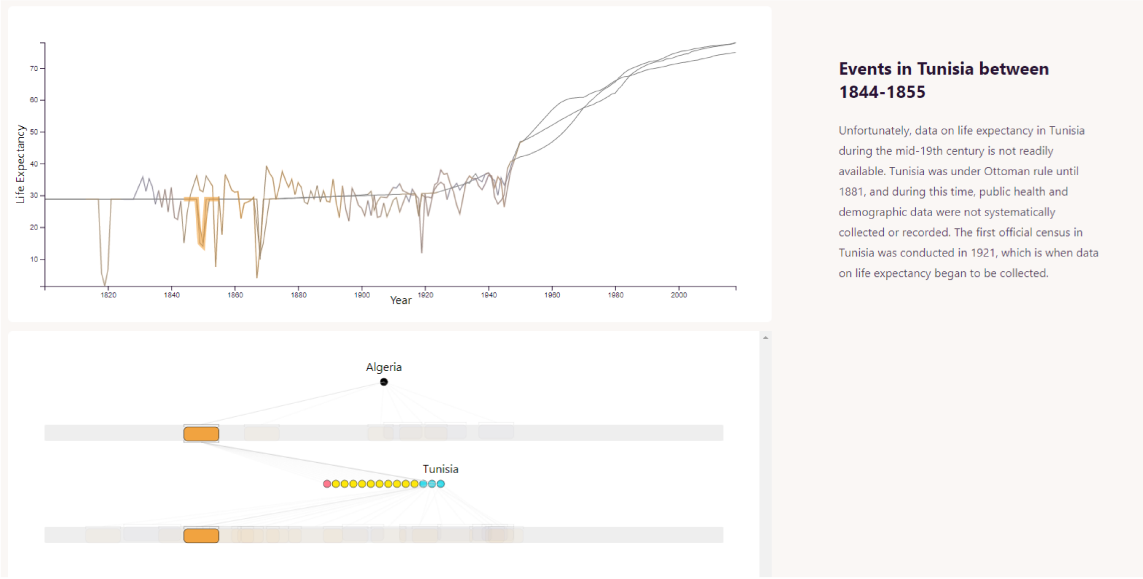


Figure 4.9: Example of ChatKG being used to investigate the life expectancy in Algeria and Tunisia. The left-most pattern (1844-1855) has 3 African countries (in cyan) with patterns in the same period: Tunisia, Mauritius, and Algeria. The results from ChatGPT show that Algeria and Mauritius were colonies at the time of France and Great Britain, respectively, and Tunisia was under Ottoman Rule.

the attendance or overall history of universities. He started investigating the United Kingdom since he knew of the rich history of Cambridge and Oxford universities. By investigating the largest valley of the line chart (Figure 4.10 1912-1919), he found that not only WW1 impacted the British life expectancy, but that due to the war, many universities were closed or had their operations significantly reduced. Also, fewer resources were available, and many young men were conscripted, which led to fewer students. Interested to see how the USA's universities were impacted in the same WW1 period, he found that the USA had a significant increase in the number of universities and students, which improved the overall education level (see Figure 4.11). This way, Karl found out that even though both British and Americans had lower life expectancies in this period, the way universities were impacted in each country was vastly different.

In this example, Karl queried the *IA* using a context. By specifying the context as "university education, such as the founding of universities, attendance in universities, number of students, Nobel prizes, and so on", Karl was able to retrieve information from the *IA*, which discussed not only the potential reasons

for the variation in life expectancy, but how did this variation impact universities in the given country. This way, Karl was able to discover that even though similar patterns were found in different countries, due to the fact that each country had different reasons for the change in life expectancy, the impact it had on universities in each country was vastly different.

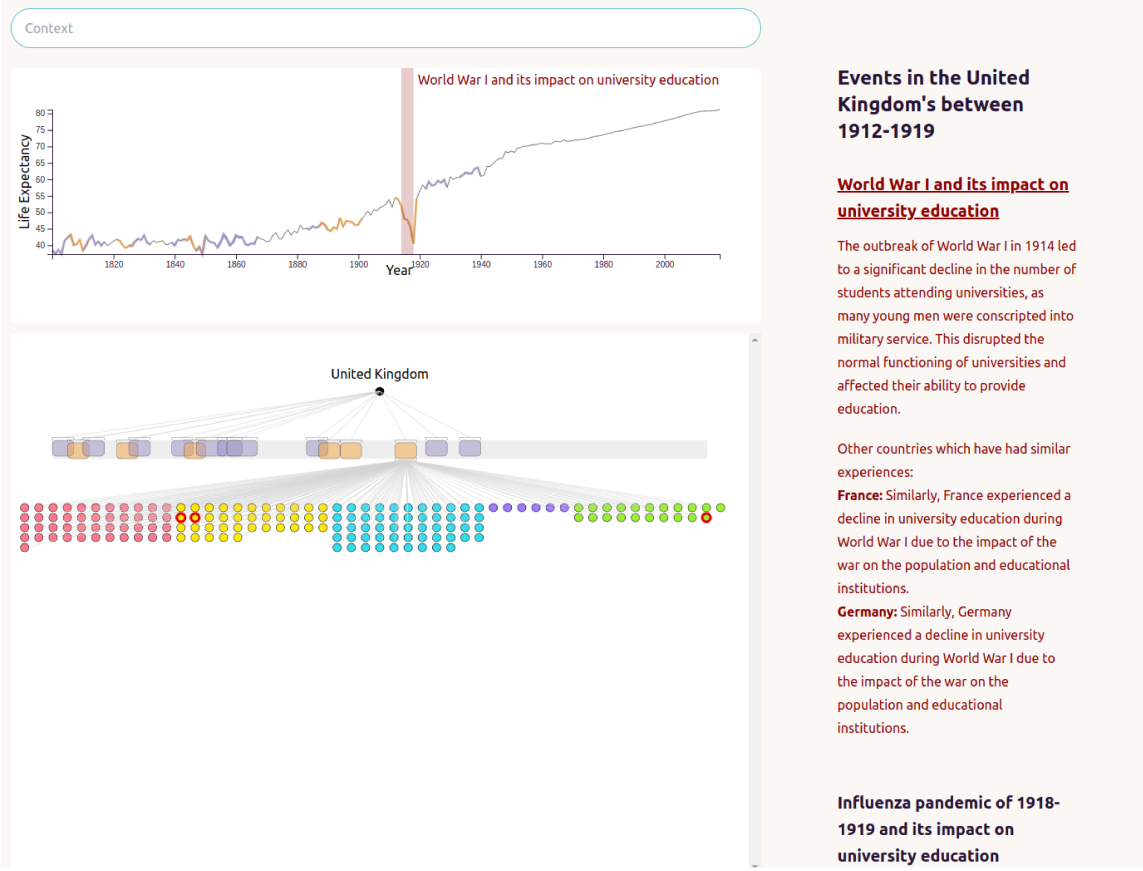


Figure 4.10: Example of ChatKG being used to investigate the life expectancy in the United Kingdom under the context of university attendance and history. The most prominent pattern (1912-1919) shows decreased life expectancy in the UK due to WW1 and Influenza. Additionally, the contextual results describe lower university attendance in the UK due to army conscription and lower funding. The highlight in dark red occurs due to the mouse hovering over the text generated by the *IA*, showing an annotation in the line chart of when WW1 impacted the UK's life expectancy.

Interview: I interviewed Catherine, an MA in history, to collect feedback on whether ChatKG is a valid approach to explore history data and verify its ease-of-use. First, Catherine was given a brief explanation of how to use ChatKG and

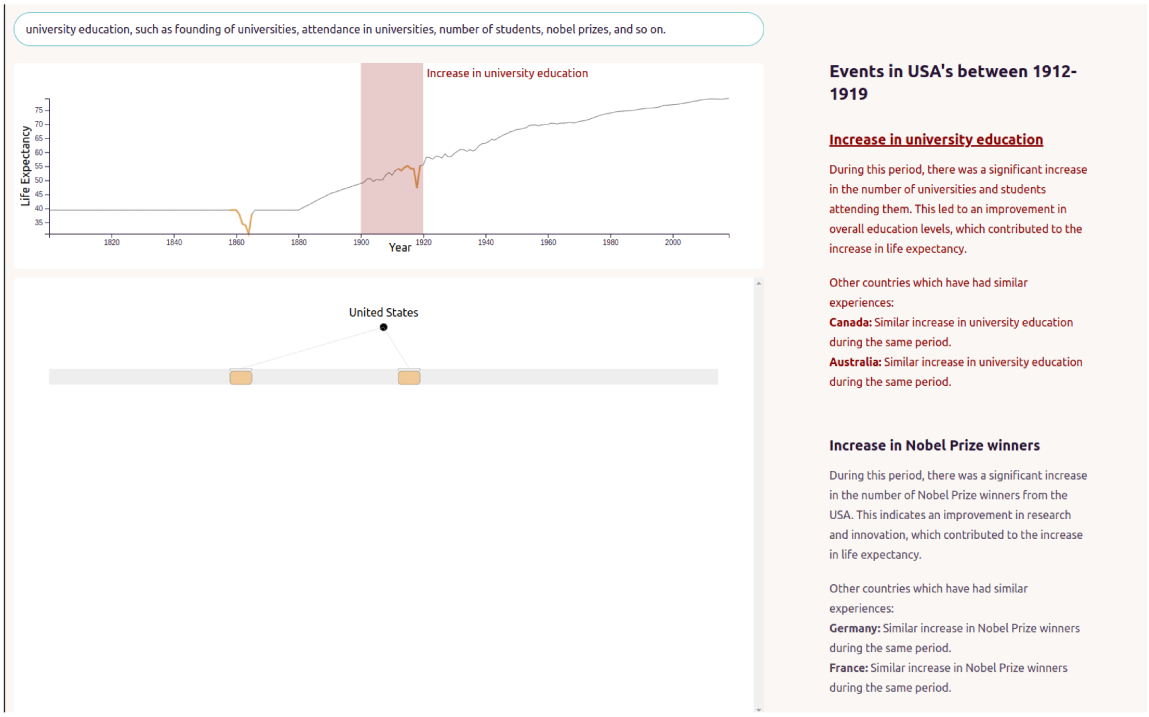


Figure 4.11: Example of ChatKG being used to investigate the life expectancy in the USA under the context of university attendance and history. The pattern which relates to WW1 and Influenza (1912-1919) also describes of the significant increase in university attendance, which is opposite of what was found from the UK's example in Figure 4.10.

was asked to use it for a use-case she might be interested in. She said Russia's life expectancy is interesting to view, and by exploring it, she identified multiple factors that lead to death across Russia, confirming her expectations. She also checked the line-chart's "deepest valley" between 1927 and 1945 to find reasons. She exemplified how she would use ChatKG in practice: "Let's say I knew that WWII was a big deal to Russia, but not that Public Health was an issue. I would then go and find books on health and Russian history. It would be helpful because I hadn't thought of nutrition and public health." She concluded by saying that ChatKG "is good because it highlights major influencing factors reasons but maybe are not talked about as much as other factors." Catherine ended the interview by saying that the current example is interesting for educational purposes, but an example that would be very beneficial for history research would be either the exploration of all available data of a single country (e.g., other datasets from UNData such as democratic index and co2 emissions) or multivariate analysis of

history datasets.

4.6 Limitations

Other use cases were investigated. Within the same UNData [161] repository, there are many other datasets, such as CO2 emissions, democracy index, and child mortality, which would bring interesting information as well. Such datasets could also be used to analyze a single country's indexes. Other potential use cases could involve sports, where users could investigate team (dataset) statistics throughout a season (temporal data). Also, other potential heuristics for pattern detection could be added to the pattern detector, such as the inclusion of trends or other time-series algorithms [16]. Of course, such use cases would also be subject to the *IA*'s ability to provide relevant information.

ChatKG has shown potential, but its goals also show its limitations. For instance, ChatKG is not well-suited for inference or explanatory analysis. It is also not well suited for non-temporal or multi-variate data since it focuses on one dimension (e.g., country) of the temporal data. Finally, it also is not suited for simultaneously comparing a large amount of information. That is, with too many datasets or too many patterns, ChatKG becomes cluttered. If the use case was the life expectancy of each city in the world, the number of cities would cause the visualization to become cluttered. I estimate that anything above 200 different temporal datasets would start hindering the analysis, although this is expected to be tested in a future evaluation. Additionally, the line chart was only usable for up to three to four lines being displayed, but more than that would cause issues with the mouse-hover functionality and become hard to read due to overlapping. Although ChatKG may benefit users in identifying time spans with many patterns, it is still necessary to investigate better ways to display overlapping patterns.

The visualization design of ChatKG focused on a specific *ontology*, which can be interpreted as a strict limitation of ChatKG and of ChatKG as a whole. A greater flexibility in the *ontology* definition could, perhaps, provide a more ample application of the ChatKG. However, I purposely decided against this to focus on ChatKG's main goal, which is to allow users to explore a time-series dataset and the *explicit knowledge* collected from its patterns. The design of a flexible visualization

that allows exploratory analysis of other temporal-based KGs is challenging and left as future work.

The *IA* itself also showed several limitations. While testing ChatKG with the GPT-4 API, it would sometimes answer that it is "unable to complete the task because it has no access to the internet" or "The task is not clear and I am unable to complete it". Additionally, due to its usage within AutoGen, the monetary cost is significant, hindering the use of ChatKG. Although GPT-4 is the *IA* model most used in the literature, I noticed during the development of ChatKG that the guardrails and price imposed onto it by OpenAI limit its use within ChatKG in any large real use-case. Yet, at the same time, running models from LMStudio [69] locally requires a very powerful machine to achieve near real-time interaction. My tests using a Ryzen 1600 CPU and an NVidia 1070GTX were unable to reach near real-time. Therefore, caching mechanisms are still needed to provide the user with a good exploratory experience. It is also important to note that due to the inconsistent nature of GPT-4 and other similar models, the workflows discussed in the use cases of Section 4.5 were not consistently followed. Instead, each time the *IA* was queried, different steps were made. Yet, the output did not vary as much as the steps taken by the *IA* to reach said output. Once again, caching strategies and user feedback, such as a confirmation that the information was correct, are strategies considered for future work. Additionally, although I propose an *IA* prompt template that successfully achieves the goal proposed, I recognize that there was no strict evaluation applied to the prompt. Prompt engineering for *IA* is a very new field, especially when evaluating in conjunction with visualizations, so no standard is yet set. Once better and widely applied evaluation processes for prompts are proposed, I intend to revisit and improve ChatKG.

Finally, although ChatKG is novel as a visualization of a *KG*, the data we are visualizing is similar to visualizations of event data or process-mining data. In future work, I intend to investigate the applicability of KGs and, consequently, ChatKG as a way to visualize data from these domains and evaluate the ChatKG approach versus research in those areas.

4.7 Conclusion

In this chapter, I laid out how to model a *KG* as a connective layer of a time-series dataset, patterns, and *explicit knowledge* from an Intelligent Agent (*IA*). I presented ChatKG as a novel visualization to explore the *KG* and implemented an example *VA* tool that identifies patterns among the life expectancy fluctuations in history across many countries, extracts the *IA*'s *explicit knowledge* of these patterns, and exposes the populated *KG* and contextualized visual annotations to the user. I discussed my findings from this example and interviewed an expert who verified the usability of ChatKG. The results indicate that, within its goals, ChatKG has successfully aided users in their exploratory analysis of temporal datasets, existing patterns, and the *explicit knowledge* extracted from the *IA*.

Chapter 5

Visual Analytic Knowledge Graph (VAKG) [1]

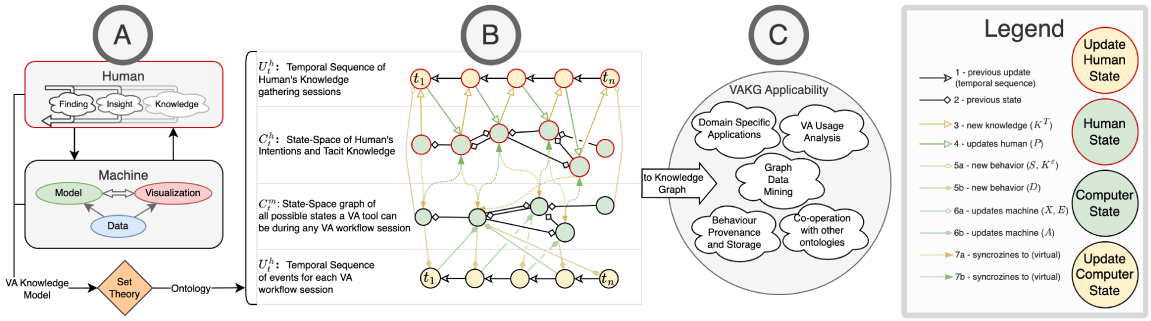


Figure 5.1: *VAKG* unfolds the interactions within the current knowledge model (A) into a temporal knowledge graph (B), which is structured as a 4-way graph containing two temporal (green) and two static (yellow) knowledge graphs. By using *VAKG*, one can structure and store the user's knowledge-gathering process and all related interactions for eventual analysis (C).

THE primary goal of *Visual Analytics (VA)* is to enable user-guided knowledge generation. *Q4EDA* and ChatKG demonstrated ways where users are able to use *VA* to gain new knowledge by investigation and interaction. Literature in *VA* theory explains the mechanics of how the different aspects of a *VA* tool bring forth new insights through user interactivity. The concept of intelligent agents is slowly being incorporated into the *VA* workflow, as discussed in chapter 1 and the next chapter. Yet, such agents are currently an entity that cooperates with users for knowledge-gathering tasks but does not learn from user behavior and insights. If intelligent agents were able to learn from past users to help new users through recommendations or automation capabilities, then *VA* would be better equipped to cater to other types of users, such as data-illiterate and lay users. As an initial step towards investigating how to use intelligent agents to learn from user's knowledge discovery, this chapter proposes *Visual Analytics Knowledge Graph*

[1] This chapter was based on *Christino, L., & Paulovich, F. V. (2023). From Data to Knowledge Graphs: A Multi-Layered Method to Model User's Visual Analytics Workflow for Analytical Purposes. arXiv preprint arXiv:2204.00585.* Submitted to Computer Graphics Forum in October 2023.

(VAKG), a conceptual framework that structures and collects user's behavior and insights during knowledge discovery, which can then be used in downstream tasks.

5.1 Overview

In this process of user-guided knowledge generation, *VA* knowledge models and ontologies have shown to be beneficial to better understand how users obtain new insights when executing a *VA* workflow. Yet, the gap between theoretical models and the practice of knowledge generation analysis is wide, and theory has mainly been used as a baseline for practical works. Also, two concepts are typically ambiguous and intermixed when analyzing *VA* workflows: the *temporal* aspect, which indicates sequences of events, and the *atemporal* aspect, which indicates the workflow's *state space*, which is the set of all states of the *VA* tool and its user occupied during a *VA* workflow. Also, the lack of guidelines on how to *analyze* the recorded user's knowledge-gathering process when compared to the *VA* workflow itself is apparent. We bridge this gap by presenting *Visual Analytics Knowledge Graph (VAKG)*, a conceptual framework to bridge the gap between *VA* workflow modeling theory and application. Through a novel *Set-Theory* formalization of knowledge modeling, *VAKG* structures a *VA* workflow by temporal sequences of human and machine changes over time and how they relate to the workflow's *state space*. This structure is then used as a schema for storing *VA* workflow data and can be used to analyze user behavior and knowledge generation. *VAKG* is designed following the needs and limitations of relevant literature, allowing for modeling, structuring, storing, and providing analysis guidelines for user behavior and knowledge generation.

5.2 Introduction

VA tools allow users to harness insights and knowledge from datasets [193]. By tracking this insight-generation process with provenance methods, like screen and mouse-click recording, researchers and industry alike can better understand the relationship between their tools and their users. The theoretical foundation of *VA* of Sacha et al. [193] proposes a "knowledge generation model" which not only

discusses how the human and the machine interact during an insight-generation process but also discusses what elements within these interactions relate to the various data collected in provenance methods. Among existing works, Federico et al. [72] use the theory of Sacha et al. [193] to propose *VA* as a workflow of *events*, which can potentially be associated with provenance concepts. Other works which propose novel *VA* systems also use the knowledge modeling theory to describe the process where their users gather knowledge, providing them with a basic best-practice design guideline of how to model [193, 72], structure [44, 195] and understand user behaviour [241, 152, 107, 251].

Structuring *VA* workflows has been a hot research topic [195, 44], attracting enormous interest in tracking and analyzing user behavior to understand how knowledge is generated [251]. In addition to investigating the role of the *VA* workflow during analysis, such works investigate how the users' pre-existing knowledge [107, 20] influence their experience during *VA* tasks. However, behavioral analytics do not yet use *provenance* in light of the existing knowledge models or ontologies. Instead, each research endeavor develops its own method to acquire, structure, and analyze the users' knowledge-gathering process. Therefore, although Sacha et al. [193] describes the conceptual relationship between *Machine* and *Human*, some essential aspects are overlooked in behavior analysis. For instance, given that a *VA* tool follows the workflow of a specific *VA* model, how would one use this model as the means to acquire, structure, and store ongoing user interactions, or namely *behavior provenance*? Or how might the recorded data be analyzed to investigate and compare the knowledge gathered among several users, or namely *knowledge provenance*? And how can a single dataset be defined which relates the user's behavior and the gathered knowledge where one can discover which sequences of actions lead to a new insight or which insights were attained due to using a specific, perhaps new, visualization? Or even how to use the answers to these questions for other downstream tasks, such as aiding the development of new tools or comparing different *VA* tools?

Visual Analytics Knowledge Graph (VAKG) addresses these open questions. *VAKG* is a novel conceptual framework that proposes a formalized process to extract the underlying *VA* model of a *VA* tool, to design a knowledge graph *ontology*

following the model, to define the data to be collected from the user behavior and knowledge gathering which fits said *ontology*, to populate a knowledge graph containing *behavior provenance* and *knowledge provenance* data, and finally to use said knowledge graph for analysis of the relationship of behavior and knowledge. For this, I use existing *VA* knowledge models [193, 72, 236] and reinterpret them as *sets of information* and the process of how these sets interact. This way, *VA* is separated by its temporal aspect (e.g., temporal sequences of events versus atemporal *state spaces*) and ownership aspect (e.g., *Human* versus *Machine*). I then define a novel multi-layer knowledge graph structure that follows the *sets of information* and their relationships.

The main contributions can be summarized as follows:

- A reinterpretation of *VA*'s knowledge model through *Set-Theory* and the relationship between the modeled sets;
- A domain-agnostic knowledge graph structure definition based on *VA*'s knowledge model; and
- A novel usage of a multi-layered Temporal Knowledge Graph architecture as a storage, analysis, and visualization mechanism of *VA* workflows for understanding the relationship between user behavior and knowledge acquisition.

Consider this sample workflow: two data analysts [32, 87] intend to investigate supermarket transactions dataset [218] using Tableau [159]. Their workflow can be summarized as downloading the dataset, verifying it is correct, and checking the store's profitability by creating and analyzing various visualizations. By mapping each step of the users' workflow to entities of a *VA* model [72], *VAKG* provides a knowledge graph structure that relates user behavior and knowledge acquisition. Then, the knowledge graph can be built by recording each user's behavior and thought process. For instance, "creating a profitability bar chart" would be related to the next task of "inspecting the tallest bar" and the new knowledge of "country X is the most profitable". Finally, this knowledge graph can be used for downstream tasks using graph analytics. Questions like "Which user had more insights during the process?" or "Which user took the least amount of time/steps to find the answer?" can be answered through the page-rank and shortest-paths algorithms, respectively. Therefore, *VAKG* not only models a workflow but also defines what

data is relevant to be stored, such as user insights and interactions, in order to analyze the users' knowledge-gathering process, providing a unified and repeatable theoretical approach to bridge *VA* knowledge models, *behavior provenance* and *knowledge provenance*.

The remainder of the chapter is structured as follows. In Section 5.3 and Section 5.4, I introduce relevant concepts and discuss related work involving techniques that seek to formalize the *VA* knowledge flow, usages of knowledge graphs within *VA*, including how they differ from *VAKG*, and other concepts that tackle the ongoing knowledge evolution during data analysis. In Section 5.5, I extend the existing works of the theoretical knowledge model of *VA* to formalize *VAKG*. In Section 5.5.3, I present possible applications of *VAKG* while comparing it with existing methods and justifications for further extending *VAKG*. I conducted a case study to demonstrate the practical application of the *VAKG* to a *VA* tool that analyses interactive clustering of textual documents in Section 5.5.3 called ModKT. The researchers developing ModKT are using the results to decide the next steps in their work. Finally, in Section 5.6, I discuss current limitations and the next steps within the research plan. In Section 5.7, I draw my conclusions.

5.3 Theoretical Background and Definitions

Researchers typically prefer to define their workflow descriptively for particular use cases or follow certain well-tested processes. Theoretical research in the model design of *VA* workflows reflects this diversity very well. To properly position *VAKG* within the theoretical literature, I first define how the theoretical literature sets itself. This chapter will follow the definitions of Chen et al. [44] where the contribution of theoretical *VA* works is categorized as one or more of the following:

Definition 5.3.1 (Principles and Guidelines)**.** *Qualitative descriptions or rules that define a process that may lead to the desired outcome. Examples can be found in works that extract the qualitative elements of a VA workflow and define rules based on it [195, 28].*

Definition 5.3.2 (Taxonomy and Ontology)**.** *A collection of concepts that defines a well-defined structure. Such research usually focuses on a novel theoretic ontology to structure the knowledge generation workflow [193, 241, 195, 45, 43, 175].*

Definition 5.3.3 (Conceptual models)**.** *Abstract representation of a real-world process using a collection of theoretical taxonomies, typologies, and guidelines. For my purposes, a VA knowledge model is a model of a user's knowledge generation throughout a VA process. Arguably, the most prominent example of such a model is of Sacha et al. [193]. Generally speaking, knowledge modeling defines a workflow where insights lead to knowledge generation [6].*

Definition 5.3.4 (Theoretic frameworks)**.** *Collection of operators which to measure a process (e.g., mathematical operators). For instance, the theoretic system defined by Federico et al. [72] can describe and measure the process of many existing VA systems and tools.*

Definition 5.3.5 (Quantitative laws)**.** *Describes causal relationships between conceptual models by means of a theoretic framework. For example, Federico et al. [72] apply this concept when comparing multiple VA knowledge models.*

Definition 5.3.6 (Theoretic systems)**.** *An extension of a conceptual model that uses theoretic frameworks to define a real-world process formally. Federico et al. [72] extend several conceptual models in such a way as to formalize their methodology.*

These concepts are not consistently used in the *VA* literature [72]. In order to better contextualize *VAKG*'s goals, *VAKG* itself defines a theoretic system based on the set-theory theoretic framework, the conceptual model of Sacha et al. [193], and the *ontology* of Federico et al. [72]. Beyond theory, the practical use of *VAKG* by applying the proposed theoretic system in practice by performing *behavior provenance* and *knowledge provenance* analytics. Because of this duality of *VAKG*, I classify it as a *conceptual framework*. Nevertheless, the goal of *VAKG* was defined by investigating the connections between theoretical and practical related works.

5.4 Related Works

This section presents an overview of how existing theoretical and non-theoretical works are related to *VAKG* while also considering the definitions of Section 5.3.

5.4.1 Related Theoretical Works

Knowledge modeling defines a workflow where user insights lead to knowledge generation [193, 6]. For this, it defines the relationship between users' interactivity and all computer operations and data [193]. For instance, Figure 5.1(A) summarizes this knowledge model, showing how knowledge generation and user interactivity are linked. Although such work is instrumental as a foundation throughout the *VA* literature, it cannot be directly applied in practice for provenance analysis.

Definition 5.4.1 (Provenance)**.** *Tracking and using data collected from a process, such as a Vi&VA tool being used by a user.*

On the other hand, *ontology* structures [193, 241, 195, 45, 43, 175] are being used as a means to link knowledge models to real-world workflows. Vis4ML [195], for instance, describes an *ontology* for machine learning in *VA*, and, with it, users can easily model and structure a machine learning workflow. Howsoever relevant these works may be for *VAKG*, their contribution is still only theoretical, not tackling how to store any data generated from executing a *VA* workflow nor discussing how or if such data can be collected and used for downstream tasks, such as data analysis. In other words, research on taxonomies and ontologies that structures knowledge gathering in *VA* does not, by design [44], provide an overarching *theoretic system* to link *VA* theory and the practice of provenance.

Since the origin of *VA*, significant work has been done to demonstrate the breadth and depth of *knowledge* within *VA* [193]. The *theoretic system* of Federico et al. [72] is versatile enough to describe many existing *VA* tools. More specifically, they show how the subsequent interactions and *feedbacks* between the user and the computer are related. They also describe how automatic processes in data mining can generate new visualizations or how machine learning can help the user understand the data itself. Nevertheless, although the works listed and described by Federico et al. [72] may differ, *VA*'s purpose of creating insight or knowledge through a given workflow is common to all of them and is generally done through interactivity between the user and computer [72, 193, 43]. Even though their *theoretic system* can formalize the *VA* knowledge model and exemplify

its application in practice, it by itself still lacks an *ontology* to structure, store, and relate the provenance-related data, such as user behavior and the knowledge gathered.

Although the presented theoretical research, such as knowledge models, taxonomies, ontologies, and theoretic systems, are instrumental to understanding how current *VA* systems produce knowledge, I have also identified their insufficiency in providing insights into the ongoing knowledge generation process throughout a *VA* workflow. In other words, they cannot be used to simultaneously model, store, and create links between a *VA* tool's usage, the user's behavior during a *VA* workflow, and the user's knowledge-gathering process. *VAKG* attempts to bridge this gap. However, my work does not try to redefine any of the taxonomies and principles described so far. Instead, *VAKG* uses the same taxonomies and principles as most [72, 43, 175]. Also, although *VAKG* provides a more comprehensive structure to relate user behavior and the knowledge-gathering process, I recognize the existing works' advantage in other areas (e.g., data mining [195] and machine learning [242, 195]). Therefore *VAKG* does not aim to supersede existing structures or ontologies with its own. Instead, *VAKG* requires that a given *VA* tool be modeled using existing *VA* models, then used to define its *Knowledge Graph* structure. Thus, *VAKG* bridges the gap between *VA* theory and its applicability in practice to provide a cohesive structure to relate and analyze user behavior and knowledge gathering.

5.4.2 Related Applications and Frameworks

Theoretical research on *VA*'s knowledge model has tackled the problem of knowledge gathering in many different ways. However, knowledge gathering within these works and systems is seen only as theoretical background. Federico et al. [72] lists many systems where a notable example is the work by Keim et al. [122], which creates an application-specific knowledge-gathering process by utilizing automated analysis with human interaction; however, by verifying these related works, I note a lack of standardization of how to apply the theory in practice. Federico et al. [72] argues that since this knowledge-gathering loop is conceptual, it is "often inconsistently used," which shows a missed opportunity to define how to apply such theory in practice in a consistent way. This inconsistency has another

consequence: although their results relate to each other, these works do not seem to be able to communicate. In other words, I am unable to compare their results.

Furthermore, the two sides of knowledge gathering are often not well separated: the temporal sequence and the workflow's *state space*, which denotes the set of all possible states independent of time. In other words, although a knowledge-gathering process can be defined as a linear sequence of new knowledge "events" over time, it can also be defined as a time-independent set of all gathered knowledge. With *VAKG*, I first explain the advantages of separating these concepts and using each of the concepts in a unified framework. Different from other works [72], *VAKG* uses this as one of its core design goals.

Definition 5.4.2 (State Space)**.** *The set containing all possible configurations of a system or tool.*

User Behaviour Tracking and *behavior provenance*: User-tracking and behavior analysis research has also been active [251]. For instance, the user-tracking taxonomy of von Landesberger et al. [241] models user behavior as a graph for analytical purposes. However, *VA* tools cannot integrate directly with theoretical works such as these. Instead, existing *VA* systems use these taxonomies as a theoretical or conceptual background while using the user-tracking data solely for specific domain use cases, as is extensively discussed by Xu et al. [251]. For instance, the user's *Tacit Knowledge* [72] is tracked in *VA* by many different feedback methods, such as manual feedback systems [21, 152], manual annotations over visualizations [208], and inference methods that attempt to discover the user's insights by analyzing their interactivity patterns [165, 20]. However, these works do not directly use any previously discussed theoretical results. Instead, they are only seen as a motivation for their domain-specific solutions. Among these *VA* systems, InsideInsights [152] and SenseMap [165] are the only ones that get close to addressing this limitation. SenseMap first creates a graph network with *behavior provenance*, then allows users to analyze the recorded graph by manually constructing a so-called "Knowledge Map". InsideInsights, instead, records user behavior and user annotations simultaneously during the user's analytical process. Though InsideInsights and SenseMap provide a way to record and analyze user behavior, the proposed solutions are domain-specific and do not discuss the relationship

between users' behavior and the knowledge gathered by the user. For instance, InsideInsights does not allow tracking auto-generated insights [209] and does not account for automatic computer processes [72] or external agents [68, 157]. *VAKG*, however, also tackles these aspects.

Knowledge Provenance: Significant research has been done to better understand the concept and applicability of knowledge gathering in practice regarding *knowledge provenance*. Knowledge provenance is a specialization of *Data Provenance* [63, 82] for collecting, storing, and tracking users' knowledge-related events. Knowledge provenance researchers argue that tracking user's knowledge gathering can be done by recording any change in the available datasets [62] (e.g., data pre-processing) or updates in visualizations [20, 251, 241]. Among such works, Chang et al. [41] attempt to use visual analysis within a Knowledge Base system, storing knowledge extracted from experts into a "compressed" format. Works such as these show examples of applying provenance to understand users' knowledge gathering.

Still, although these works describe ways to link knowledge gathering to user interactions, it is rare to see a differentiation between the temporal sequences of user-generated events and the atemporal *state space* of the *VA* workflow. Therefore, the following two concepts are either merged or ambiguous in these works: the *temporal* aspect, which indicates what and when users executed *VA* tasks, and the *atemporal* aspect, which indicates what the possible *VA* workflow states and how they transition between each other are. Instead, when these works explicitly define a structure, they either store the temporal sequences of events without indicating whether they occurred previously or the *state space* without recording the temporal sequence of events. Similarly, these works assume that *knowledge provenance* is a subset of data provenance, or in other words, that all knowledge-related changes can be extracted from the user's behavior. This does not match with the knowledge definition of *VA*'s knowledge models [193, 72] where certain concepts, like behavior and knowledge, are separate. Likewise, most related works do not tackle how to interpret multi-user *VA* workflows [20], nor allow for comparisons between the user's exploratory space when compared to their motifs [251]. *VAKG* bridges these gaps by modeling the difference between *behavior provenance* and *knowledge*

provenance and the difference between temporal events and atemporal *state space*. *VAKG* encodes this model into a knowledge graph that relates users' behavior and knowledge-gathering sessions.

Knowledge Graphs (KGs): While Knowledge Provenance focuses on tracking and storing knowledge, *Knowledge Graphs* (KGs) [74, 45, 137] have aimed to be a proper way to structure and analyze knowledge-related data. KG is a widely used technique to structure knowledge as a graph network, usually done by formalizing the structure as an *ontology* through the Web Ontology Language format [43, 195, 242]. For instance, DBPedia [12] uses *ontology* design and KGs to transform unstructured knowledge into structured knowledge. In other words, KGs are a graph database of knowledge that employs knowledge model [193] ontologies. Compared to typical databases, the structure of KGs focuses less on the usual row-based structure [40] but uses the relationships between taxonomies as the foundation of knowledge. Although KG itself focuses on the structure of knowledge-related data, it is supported by various other graph-theory contributions, such as Graph Neural Networks (GNNs) [118], graph visualizations [42, 106] and graph operations [116], like Page Rank and Traveling Salesman. KGs are, therefore, not limited to only providing a functional structure, but given a KG, users can employ graph analysis techniques to query and analyze the data.

Temporal Knowledge Graphs (TKGs): A notable sub-type of KGs is *Temporal Knowledge Graphs (TKGs)*, where the graph edges encodes the temporal relationship of the data, such as "order of events" or "time difference between events" [92]. That is, while a KG is a graph structure where knowledge reasoning is modeled as connections between classes or properties, such as "George Washington *is a* human" and "Canada *is a* country", a *Temporal Knowledge Graph (TKG)* models these connections as the temporal relationship between the classes or properties. Many types of TKGs exist, and their temporal relationship varies among them. For instance, TKGs can relate two nodes by temporal co-occurrence. An example of such a KG would be all purchases done between different businesses within a supply chain, where the product "Mayonese" may have been bought by "Walmart" from the seller "Hellmann's" on "25/06". In this TKG, the connection between the three nodes: Walmart, Hellmann's, and Mayonnaise, would be "25/06". Though

some existing works which define knowledge graphs [40, 251, 164, 137] or ontologies [195, 43] are already used for structuring knowledge and *behavior provenance*, no current work, as far as the authors know, uses TKGs to structure *knowledge provenance*.

Process Mining: The act of structuring and analyzing a process in a graph format has been extensively researched by Process Mining [234]. Indeed, the relationship between Process Mining and *VA* has grown tremendously in recent years. Process Mining proposes a way to define any given process by a workflow consisting of nodes and their relationships. The concepts of *events* and *knowledge graphs*, which are very relevant for *VAKG*, have appeared in many recent works [71], showing how Process Mining is a proven form of modeling processes for provenance purposes [258, 233]. Yet, Process Mining is centered on behavior and events, that is, *behavior provenance*. *VAKG* aims to relate behavior to *knowledge generation*, which differs from existing works' goals.

5.4.3 Survey Compilation and Goals

I compiled a survey analyzing the most relevant literature cited so far to verify how the theoretical concepts are applied in practical related works. I found that the main differentiation of *VAKG* is its theoretically grounded pipeline, which, in a simplified manner, one must: model a given *VA* tool using a *VA* model and *ontology* [72], declare a knowledge graph structure that matches said *ontology*, perform data collection through *behavior provenance* and *knowledge provenance* to populate the knowledge graph, and finally analyze said knowledge graph (see Section 5.5). Thus, *VAKG*'s major goals are:

- **G1. Analysis-centric *VA* Model:** Temporal and atemporal interpretations of *Human* and *Machine* components of the *VA* workflow are used, but inconsistently, so *VAKG* proposes a consistent one which partitions the *VA* workflow as:
  - **G1.1:** Temporal-sequences of user's knowledge gathering (Knowledge Provenance or *Human Updates*) [193, 195, 175, 241, 72, 28, 242, 45, 117, 62, 20, 55, 152, 21];

  - **G1.2:** User intentions and insights which occur within a *VA* workflow (*Human state space* or just Human State) [193, 195, 175, 72, 28, 242, 43, 45, 12, 42, 106, 117, 20, 107, 55, 152];
  - **G1.3:** The *VA* tool's states during all *VA* workflows are modified due to user behavior (*Machine state space* or just Machine State) [193, 195, 175, 241, 28, 43, 45, 12, 42, 106, 117, 36, 62, 20, 107, 55, 209];
  - **G1.4:** Temporal-sequences of the *VA* tool events/tasks which are executed during *VA* workflow sessions (Behaviour Provenance or *Machine Updates*) [193, 195, 175, 72, 28, 12, 117, 36, 62, 20, 209, 152, 21];

- **G2. Ontology of the *VA* Workflow:** Formalization of a structure that, while being rooted in an existing *VA* knowledge model [193, 72], describes the *VA* workflow following **G1** [195, 175, 28, 242, 43, 45, 12, 42, 106, 117, 62];

- **G3. Data Retention:** The structure is used as a schema of a data retention solution where to collect and store user behaviors and interactions during a *VA* workflow [175, 241, 72, 43, 45, 12, 42, 106, 117, 36, 20, 107, 209, 152, 21];

- **G4. Data Analysis Capabilities:** Use the data and/or structure to perform analysis, such as per-user analysis, user comparison, usage comparison, and so on [242, 43, 45, 12, 42, 106, 117, 36, 62, 107, 55, 209, 21];

The next section describes how *VAKG* reaches these goals.

5.5 The VAKG Conceptual Framework

Let's assume a group of researchers created a *VA* tool for the analysis of temporal series and now wants to understand if, how, and what users learn while using their tool. The *Visual Analytics Knowledge Graph (VAKG)* method gives this group a formalized process to extract the underlying *VA* model of a *VA* tool, design a knowledge graph that follows the model, and define which data from the user needs to be collected for a thorough provenance of the user's behavior when using the tool and their newly acquired knowledge from the tool.

First, *VAKG* requires that a *VA* knowledge model is matched to the tool [193, 72] (see Figure 5.1[A]). By *VAKG* reinterpretation of the model in the lens of *Set*

Theory (**G1**), *VAKG* identifies what are the unique elements that constitute a *VA* workflow of that specific *VA* tool and what are their relationships to each other. This *VA* workflow is then structured following *VAKG*'s *ontology* that relates the users' interaction events and knowledge generation (see Figure 5.1(B) and **G2**). The result is a knowledge graph structure that separates the workflow's temporal aspect, which is defined as behavior sequences of events (**G1.4**) and knowledge-gathering sequences of events (**G1.1**), the workflow's atemporal aspect, which is structured as the *VA* tool's *state space*(**G1.3**) and the users' knowledge *state space* (**G1.2**). *VAKG* then uses the knowledge graph structure as the design pattern for a multi-layer *Temporal Knowledge Graph (TKG)* where the *VA* tool can record user sessions (**G3**). Finally, this populated knowledge graph is available to users, such as the research group of the example above, to apply graph-network techniques to analyze, predict, and recommend user behavior and knowledge-gathering effectiveness when using the tool (**G4**).

5.5.1 Foundation: VA Knowledge Model and Set Theory Reinterpretation

The theoretical background of *VA*'s knowledge model is a foundation work for research within *VA* (see Section 5.4). Unlike such works, I use the knowledge model of Sacha et al. [193] as a foundation to formalize and derive *VAKG* (**G1**). This section reinterprets the *VA* knowledge model to define the four aspects of the Analysis-centric *VA* Model here being discussed: human update, human state, machine state, and machine update.

The simplistic representation of *VA*'s knowledge model shown in Figure 5.1(A) characterizes its two main actors: *Humans* and *Machines*. This concept originates from the literature where knowledge is generated over time [193] though the interaction between Human and Machine [72]. The literature also proposes a mathematical interpretation of the *VA* model called the "Conceptual Model of Knowledge-Assisted *VA*" as the foundation of the knowledge model, which is expressed visually in Figure 5.2.

All in all, the *VA* interactivity model is divided between two separate actors (machine and human) and describes how knowledge is generated, converted, and used within the *VA* discourse. Each actor is then associated with a taxonomy of available

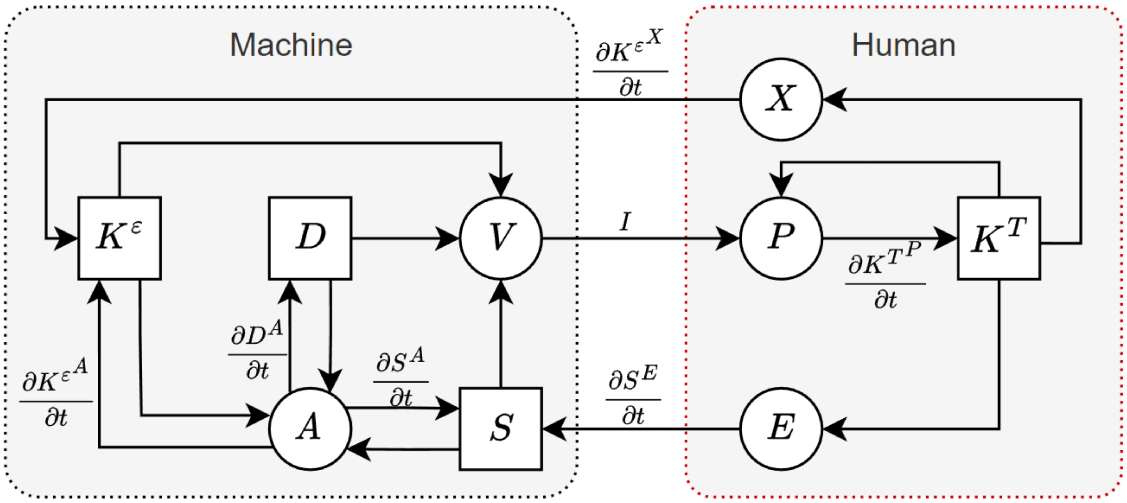


Figure 5.2: Conceptual Model of Knowledge-Assisted *VA* [72]. *VAKG* structures its mathematical framework by deriving the equations from this model.

actions (1): analysis A, visualization V, externalization X, perception/cognition P, and exploration E. These actions are connected by intermediate stateful taxonomies (2): explicit knowledge K^{ϵ}, data D, specification S, and tacit knowledge K^T; and (3) a non-persistent artifact: image I'' (see Figure 5.2).

From Figure 5.2, I find that the circle nodes $\{V, P, E, X, A\}$ represent elements that cause changes within a *VA* tool. For instance, the visualization **V** resides in the machine space, and it causes changes in the perception/cognition of the user **P**, providing new insights. Similarly, the exploration task **E**, which is in the human space, executed by the user can update the *VA* tool's specification **S**, which may update the data or visualization being shown within the *VA* tool. From Figure 5.2, I also find that the set of rectangle nodes $\{K^T, D, S, K^{\epsilon}\}$ represents static information. For example, elements such as data **D**, specification **S**, and tacit knowledge **Kt** represent the fact that the *VA* workflow has static information about a dataset, the state of the *VA* tool, and the user's tacit knowledge, respectively.

From Federico et al. [72], I can identify all the moving parts within the knowledge model iterative loop of a given *VA* tool. The first contribution of *VAKG* is to reinterpret the iterative loop of Figure 5.2 through the lens of *Set Theory*, allowing this process to be applied to other *VA* models and tools. Therefore, first, I define four sets of information: the *Machine Update* $U_t^m = \{V_t, A_t\}$, the *Machine State*

$S_t^m = \{D_t, S_t, K_{t+1}^\epsilon\}$, the *Human Update* $U_t^h = \{X_t, P_t, E_t\}$, and the *Human State* $S_t^h = \{K_t^T\}$. Next, from Figure 5.2, I extract how each of these elements relates to each other. Each equation below represents which information (rectangle node) directly depends on a process (circle nodes) and which processes directly depend on information:

$$K_{t+1}^T \Leftarrow P_{t+1} \Leftarrow K_t^T + V_t(I) \tag{5.1}$$

$$K_{t+1}^\epsilon \Leftarrow X_{t+1} + A_{t+1} \Leftarrow K_t^T + (K_t^\epsilon + S_t + D_t) \tag{5.2}$$

$$S_{t+1} \Leftarrow E_{t+1} + A_{t+1} \Leftarrow K_t^T + (K_t^\epsilon + S_t + D_t) \tag{5.3}$$

$$D_{t+1} \Leftarrow A_{t+1} \Leftarrow K_t^\epsilon + S_t + D_t \tag{5.4}$$

By using these equations, I reach that the human state and machine state are updated as follows:

$$S_{t+1}^h = \{K_{t+1}^T\} \Leftarrow U_{t+1}^h(S_t^m) \tag{5.5}$$

$$S_{t+1}^m = \{D_{t+1}, S_{t+1}, K_{t+1}^\epsilon\} \Leftarrow U_{t+1}^m(S_t^m) + U_{t+1}^h(S_t^h) \tag{5.6}$$

That is, the *human state* is updated due to a *human update* caused by some change within the *machine state* (Equation 5.5). Similarly, the machine state is updated due to a human or machine state change (Equation 5.6).

Following this process, any *VA* tool can be decomposed into the four sets of state and process entities, and the equation list with the relationships between the entities within the sets. Back to the example scenario discussed in the introduction: a data analyst wishes to investigate the supermarket dataset [218] using Tableau [159]. In this simplified scenario, the "*VA* tool" is tableau. The available usages of Tableau can be mapped to the nodes of Figure 5.2. For example, let's assume a user wants to create a visualization in Tableau. The data D is the supermarket dataset, the state of tableau S represents what visualization, if any, is currently being shown, and the creation of a new visualization E would update the state of tableau S, generating the new visualization V with which the user can investigate P. In other words, the node E, part of the Human Update set, leads to a new visualization.

In mathematical terms: $S_{t+1} \Leftarrow E_{t+1} \equiv S^m_{t+1} \Leftarrow U^h_{t+1}$. That is, in this example a human update U^h_{t+1} led to a new machine state S^m_{t+1}. Still, no new data D has been generated yet, so it was removed from the equation.

Now, if the user discovers a new insight K^T from the visualization and adds it as a custom text or annotation X to the visualization, new explicit knowledge K^ϵ would be saved into the tool, causing subsequent updates following the equations above. We see, therefore, that the equations above are helpful not just to define how each of the processes $\{V, P, E, X, A\}$ updates the static information $\{K^T, D, S, K^\epsilon\}$, but to define how these updates can simultaneously be understood by its ownership (machine or human) and by its timing.

5.5.2 VAKG Ontology and Knowledge Graph Definition

So far, I have described *VAKG*'s foundation through its four aspects: human update, human state, machine state, and machine update. I also described how each aspect interacts with the others through set equations. Yet, to store data of the users' knowledge generation process, *VAKG* defines a *Knowledge Graph* (KG) structure where its nodes and relationships correspond to the four aspects of *VAKG* and their update relationship according to the set equations. This structure allows *VAKG* to use existing graph databases directly, unlike the domain-specific *VA* ontologies designed recently [195, 242, 43]. The final structure is exemplified in Figure 5.1(C), where the four color-coded horizontal lanes display each of the four aspects.

By following the Web Ontology Language (OWL) [44], *VAKG* divides the space in two ways: by its ownership (human or machine) and by its timing (state or update), which defines the four *ontology* classes: Human-Update, Human-State, Machine-State, and Machine-Update. From Equation 5.5 and Equation 5.6, *VAKG* defines the relationships between the four classes, which are represented in Figure 5.3. Namely, the relationship links (1) and (2) found in Figure 5.3 relate the previous human/machine state/update to the current one, (3) and (4) represent Equation 5.5 where a change in K^T leads to an update in P, and (5) and (6) similarly represent Equation 5.6. Finally, *VAKG* defines two extra relationships (7), synchronizing the two *state spaces*. This way, if a change in specification (e.g., new

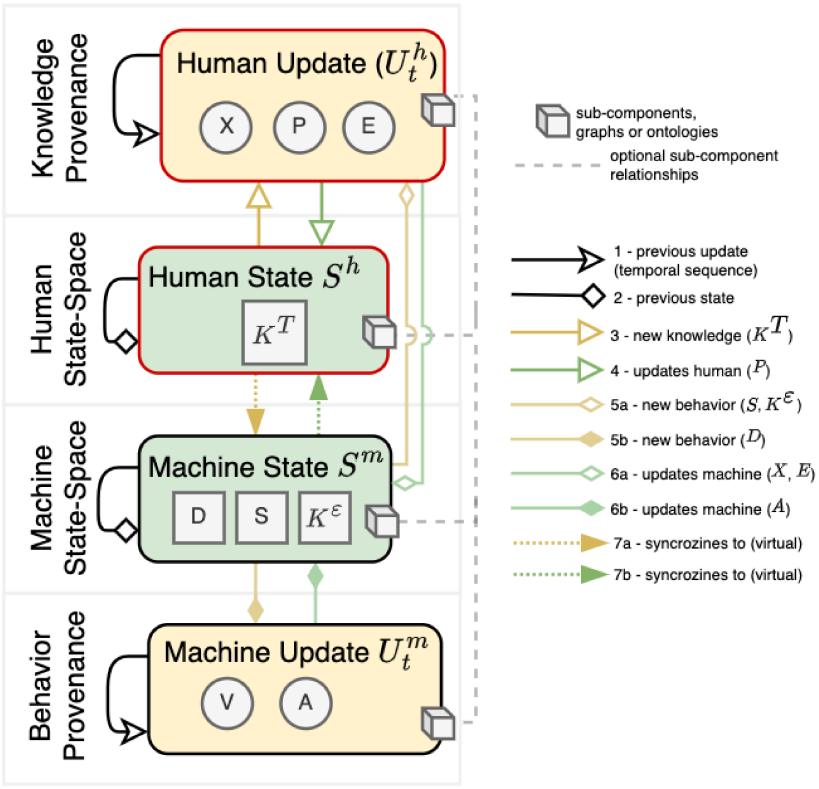


Figure 5.3: *VAKG* ontological design. The four different lanes of *VAKG* are represented. Two are KGs that describe the possible states of the inner property maps or sub-graphs. Two are TKGs describing the sequences of updates, such as the sequence of insights and knowledge gathering by users in a *VA* workflow or the sequence of computer events.

visualization) causes the user to perceive something (through (5a)), leading to new knowledge (through (3)), *VAKG* relates the starting *machine state* and ending *human state* through (7b). Similarly, if this new knowledge leads the user to externalize (6a) (e.g., add text to the visualization) ending in a new explicit knowledge K^ϵ, *VAKG* relates the starting *human state* and ending *machine state* though (7a). Figure 5.1(B) exemplifies a simple knowledge graph following *VAKG*'s *ontology*.

VAKG Property Map and Data Collection Guideline

An integral part of my proposal is to record users executing a *VA* workflow and to enable its usage for analysis. This process is called *provenance* (see Section 5.4). While the usual way of thinking of *Knowledge Graph (KG)* is to focus on *classes* and their *relationships*, *VAKG* instead gives significant importance to *property-maps*

(also called *class properties* or *data properties*). Property-maps employ the idea that every *class* can contain attached data. In *VAKG*, the property-maps of the four classes of nodes are expected to contain the relevant information of that specific class. For instance, in Figure 5.3, we see that the class human state should contain the information related to the user's tacit knowledge K^T, and the machine state information related to the dataset D, specification S, and explicit knowledge K^e. However, the *property-map* design pattern is interchangeable with the other common design patterns [160], which removes any perceived limitation of my approach.

VAKG, therefore, records the information of a node as a property-map, but how should it be recorded? And what information *exactly* should be included? This question is the underlying reason for the descriptive formalization being discussed in this section (see Section 5.5.1) because without it, I would not know precisely what information should be stored in each of the node's property-maps. For instance, I have previously described how a machine state would store information related to the dataset D, specification S, and explicit knowledge K^e, but how much of such information should be stored? Although theoretically, one could argue that storing all information related to a given state is the solution.

It is not reasonable to expect that the usage of *VAKG* would necessarily require such an amount of information. Therefore, *VAKG* proposes that the property-map of any *State* should, at the very least, uniquely identify that specific *State* within the entire *state space* of *VAKG*. Similarly, the property-map of any *Update* should uniquely identify the changes between the two Machine or two Human *States*, including the timestamp of when the change occurred. This definition establishes that a given Machine or Human *State* can repeat if the same condition occurs multiple times. It is important to note that since each specific use case of *VAKG* may vary, this part of *VAKG* is treated as a *design guideline*.

Therefore, it is essential to note that the center two lanes of Figure 5.1(B) and Figure 5.3 are *atemporal* because their connection is not temporally dependent. In other words, *Machine* and *Human* states are related not through temporal dependency but through their transition relationship. Structures like finite-state machines and discrete-time Markov chains also use atemporal transition relationships similar

to *VAKG*. For example, a machine state is related to a human state through Figure 5.3(7b) if that machine state S^m caused the human state S^h to leave a prior state S^h_a and reaches another S^h_b. This may also be read as "S^h_a lead to S^h_b when S^m happened" where the word "when" does not refer to "exact time" but to the idea of "consequence" instead.

This way, by repeating an earlier example, if a change in specification (e.g., new visualization) within a machine state S^m causes the user to perceive something (through (5a)), leading to new knowledge (through (3)) and consequently a new human state S^h_b, *VAKG* relates the starting *machine state* S^m and ending *human state* S^h_b through (7b). *VAKG* also links the two human states by relationship (2), as shown in Figure 5.3. Note that the same process happens when a new human state leads the machine state S^m_a to change to a new state S^m_b.

A consequence of this structure is that nodes in the machine and human space-states which are close (e.g., low number of relationships between the nodes) indicate that these nodes are similar since one state can quickly be reached from another through a low number of "updates". Also, if two machine states or two human states are directly connected, only a single update is responsible.

5.5.3 VAKG in Practice

The running example used until now involves two Tableau users verifying and analyzing a global supermarket store. In this section, I expand on this example as a use case of *VAKG*. I also discuss another use case with a *VA* tool called ModKT [184].

Tableau Use-Case

The first use case to discuss is the running tableau example where two [32, 87] data analysts investigate the supermarket dataset [218] using Tableau [159]. By watching the two videos, I can extract a list of tasks, interactions, questions, and insights that each user did. For brevity, here is a small sample of these insights: "task: download data", "task: find least profitable country", "interaction: create new visualization", "interaction: hover over the visualization", and "insight: the least profitable country is *C*". Each process step can be mapped to one of $\{V, P, E, X, A, K^T, D, S, K^\epsilon\}$ from Figure 5.2. For instance, "download data" is a

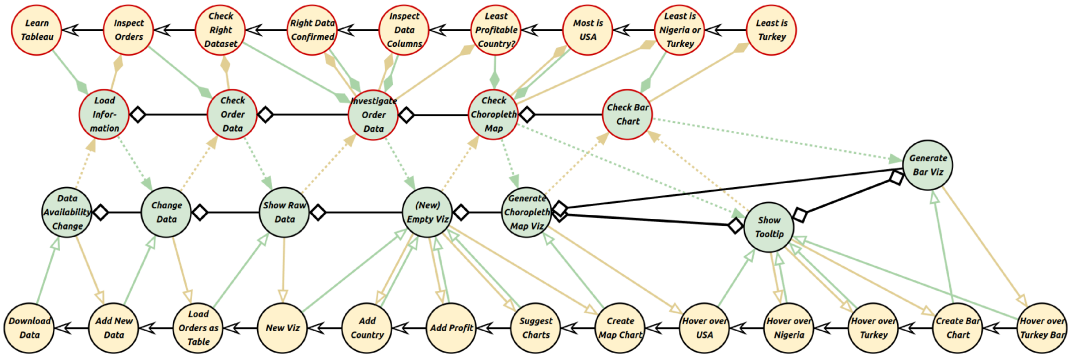
change to the data D, "create a new visualization" changes the specification S, and "found least profitable country" is a perception process P resulting in new knowledge K^T. By mapping all users' steps to the proper taxonomy, *VAKG* defines what data of each step one may need, such as the modified data in D, the new visualization type in S, and the new insight in K^T. *VAKG* also classifies each workflow step as machine update, machine state, human state, and human update (see Figure 5.3). For instance, a data change is a new machine state, and a new insight is a new human state. Similarly, *VAKG* associates the sequence of actions, such as the act of looking at the visualization P is the human update that led to the new insight K^T (see Figure 5.2). After applying *VAKG* to all steps, the result is the knowledge graph seen in Figure 5.4 of the videos' content [32, 87].

ModKT Use-Case

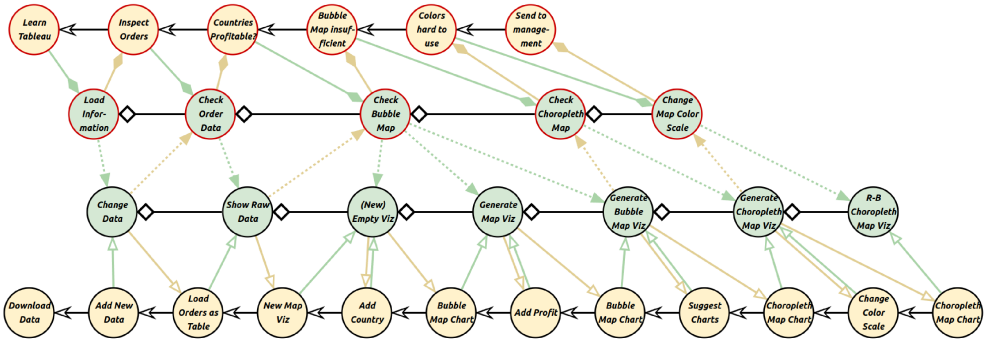
I also apply *VAKG* to ModKT [33, 184], an interactive clustering *VA* framework, to investigate which features of the tool are being used, how effective the features appear to be given insights gained while using ModKT, and to surface relevant next steps of its authors' research. In this section, I describe the tool, how *VAKG* was applied to it, and some preliminary results extracted from informal usage of the tool. This example demonstrates how *VAKG* can be applied to more complex *VA* tools and workflows.

ModKT is a tool that ingests a set of documents, such as research articles, extracts key terms of each document, and applies key-term-based clustering [201] to the corpus. ModKT uses the articles' metadata, such as abstract, authors, title, journal, bibliography type, publication year and month, and URL, for clustering. Users can visualize each document through Word Clouds, the corpus of documents through dimensionality reduction, and the comparison of the extracted key terms to custom user-defined words. Users can customize the parameters for clustering and dimensionality reduction to discover sets of (dis)similar documents and visually analyze their (dis)similarities. An overview of the system is presented in Figure 5.5.

For this user study, I have set up ModKT with a list of 660 scientific articles in the computer science field covering various text-mining visualization subjects. In order to apply *VAKG* to it, I follow the methodology process of Section 5.5: model



(a) User [32] downloads and analyses the data by validating if the data is correct, then finds an insight with a choropleth map, and finally validates it through a bar chart.



(b) User [87] downloads but does not validate the data. He then builds step-by-step a specific choropleth map design to forward to management without discussing any insight.

Figure 5.4: *VAKG* of users performing visual analysis of a global superstore's profitability. The two graph networks are shown separately for better readability. Still, all *green* nodes (*state space* nodes) with the same name are a single node in the *VAKG* graph, which is composed of both graphs simultaneously where the *state space* (green nodes) connects to the individual user's sequence of events (yellow nodes).

the *VA* tool, structure the knowledge graph, perform provenance to store user sessions with the tool, and analyze the resulting knowledge graph.

Due to the data and interactions used and expected by ModKT, I notice that even though the *VA* knowledge model of Federico et al. [72] (see Fig.5.2) could be used, it has elements that are not used by the tool, differing from examples given so far. For instance, ModKT does not allow externalizing knowledge X into new explicit knowledge K^e.

So far, *VAKG* has been exemplified only on one of the available *VA* models [72]. However, I can apply the same *VAKG theoretic framework* to other *VA* models by

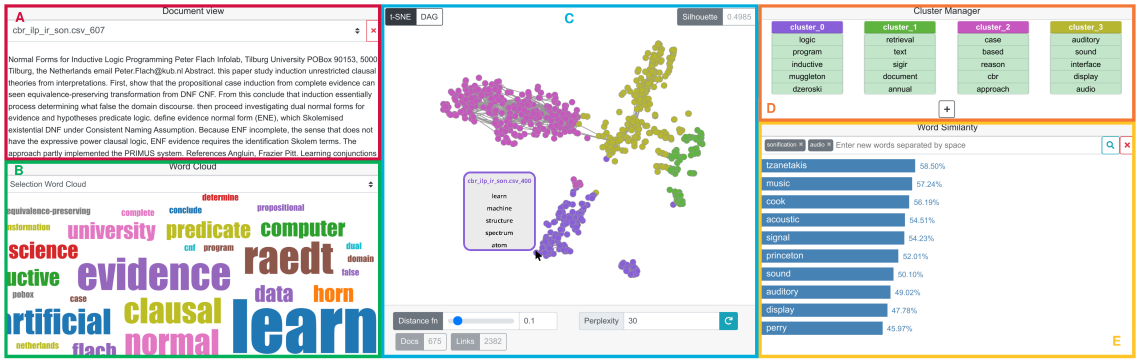


Figure 5.5: The modular architecture of ModKT interface provides a glimpse into its functionality, operating on a collection of 660 documents related to computer science subjects. The interface includes several components: A Document view (A) that shows the content of a selected document; a Word cloud view (B) that presents either the focus document or a cluster in a visual form; a Graph view (C) that illustrates the similarity relationships between the documents in the corpus; a Cluster Manager (D) allowing users to examine clusters and provide feedback to the clustering algorithm; and a Word similarity view (E) which presents a bar chart indicating the similarity between user-provided query words and the most similar identified words.

following the same procedure of applying a set Theory reinterpretation to the *VA* model and extracting the *VAKG ontology* out of the equations. In the case of ModKT, let's consider that its *VA* model follows Figure 5.6[Left]. The difference between this new model and the one used in Section 5.5 is the absence of X, K^{ϵ} and any relationships that either X, K^{ϵ} had with any other entity of the model. Following the framework above, the *VAKG ontology* is shown in Figure 5.6[Center].

Next, I apply provenance to the ModKT tool. For this, I developed a sample implementation of this *VAKG* structure [135] that receives an API call from ModKT at every user interaction. This sample implementation collects and populates a knowledge graph with user behavior data $\{V, S, D, A\}$, which is collected from mouse interactivity, and user thoughts $\{P, E, K^{T}\}$, which are collected by asking the user to type or speak into the microphone. Using speech-to-text and natural language processing, I extract keywords from the user's text and associate text and keywords with the user's behavior at the time. One of the collected user behavior and thought processes is shown in Figure 5.6[Right].

Next, I requested three researchers from the lab where I work to use ModKT with *VAKG*. They were given 400 documents with abstracts and titles and were

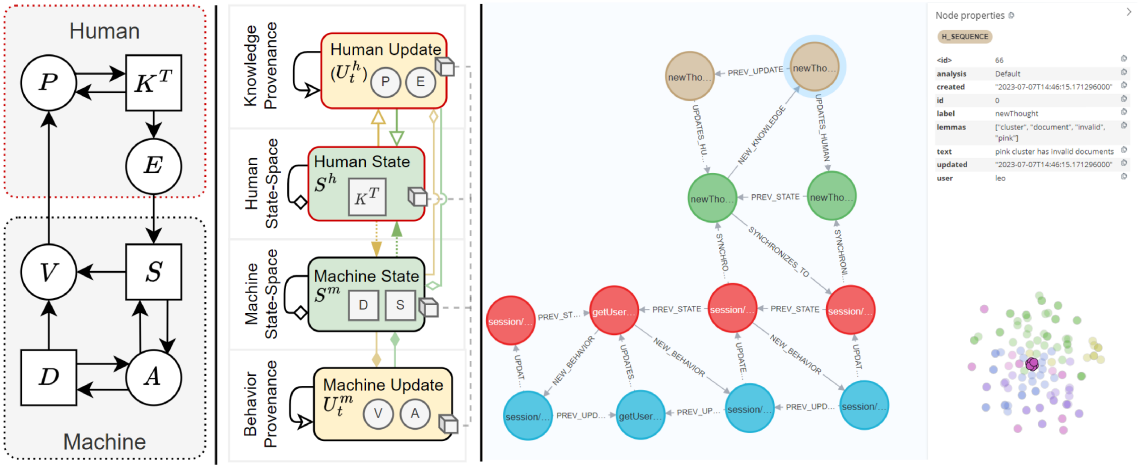


Figure 5.6: *VA* knowledge model of ModKT [Left], its corresponding *VAKG* structure [Center], and a knowledge graph generated using such structure [Right]. The model and structure only differ from Figure 5.2 due to the removal of elements X and K^{ϵ}. The knowledge graph shows four interactions and two user thoughts, the last of which is highlighted and displayed on the right panel as "Pink cluster has invalid documents". This thought was provided by the user when the pink-colored documents were investigated, as shown by the ModKT dimensionality reduction visualization on the bottom right.

tasked with finding visualization-related articles that could be included in their next research article. With the resulting knowledge graph, I could understand how the tool is generally used, list several shortcomings of the tool, and compare how the users differ in their process. The full resulting *VAKG* knowledge graph is shown in Figure 5.7.

With a knowledge graph generated, I can now explore the graph through the node-link diagram of Figure 5.7. Although all three users (B, C, and D) started out with a T-SNE projected visualization, user [C] immediately changed to a force-based layout because of the large amount of overlap, which was included in his Human Update "T-SNE not useful, switching to DAG". Though user [C] started by interacting with the visualization, user [B], instead, started by changing the clustering parameters, aiming to create a cluster to show documents related to visualization. Although user [B] could create such a cluster, their process was thwarted because the vast majority of the abstracts found were focused on NLP research and not visualization.

Interactivity-wise, the graph shows that although users took different approaches. To analyze common patterns among users, I could investigate the graph through the visualization, but for better scalability, I opted to run graph queries [162] to fetch certain information. For instance, by querying for the nodes where the users used the forced-based layout (DAG), I can see that all three users used the forced-based layout (DAG) and changed its parameters at some point. Also, by fetching which documents were clicked by each user, all users were shown to have clicked on some of the documents to read more through the abstract and word cloud view. That said, all users were also shown not to have been very successful in finding visualization-related abstracts, which indicates that the issue was not the users nor the tool but the insufficient number of documents loaded into the tool.

Feedback related to the tool functionality was also collected from the users. They discussed topics specific to ModKT, such as layout problems, the aforementioned T-SNE overlap problem, a less-than-ideal experience when reading the abstracts, and little usefulness of the word cloud. *VAKG* also collected indirect feedback on the tool's functionality. For instance, using simple counting and the aforementioned Page-Rank algorithms, I queried the number of state nodes visited by multiple users, but it was very small. That is, the three users had nearly no overlap in their interactivity, showing that the search space of the tool is vast, likely too vast. The tool's researchers concluded that reducing the possible interactivity and replacing text-only panels with static visualizations is a potentially good next step for the tool. This was corroborated by counting the number of interactions the users had until they reached certain conclusions.

Though *VAKG*, ModKT researchers could analyze user exploration paths, check which features of ModKT were most and least used, check which clustering parameters were used, and collect much feedback for future steps. ModKT researchers claim to have gained insights into the tool's capabilities and limitations by visualizing and analyzing the workflow of the individual users, giving them valuable insights into the next step of their research.

While this process could have been done through surveys, thinking-aloud sessions, screen recording, and other manual techniques, the entire process of

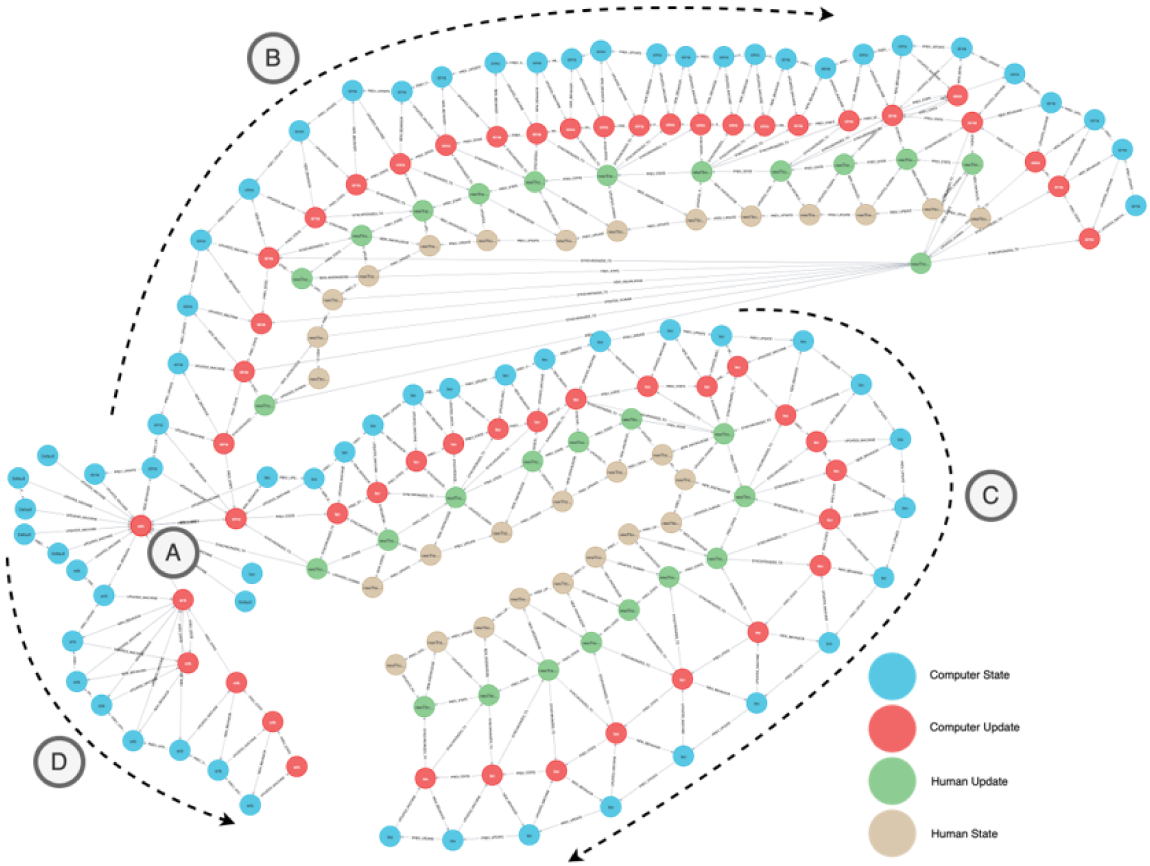


Figure 5.7: *VAKG* generated from three users interacting with ModKT. All users start at [A]. Users B and C had the same first step before diverging. User B performed two investigations, one with many steps [top] and a short one [bottom], ending in the same machine state. User D provided no thoughts, but the collected interactivity showed a back-and-forth interactivity pattern before concluding.

collection and structuring was done automatically by *VAKG*. Indeed, ModKT researchers praised *VAKG*, indicating that future user studies of their tool would be able to be done in a much more automatic and scalable manner. What previously would involve planning and manual labor, now users of ModKT just had to do interactions in an unsupervised environment and write or speak their thoughts into the built-in text widget added to ModKT. The sample implementation provided [135] collected and populated the knowledge graph shown in Figure 5.7, which was then analyzed to reach the conclusions above.

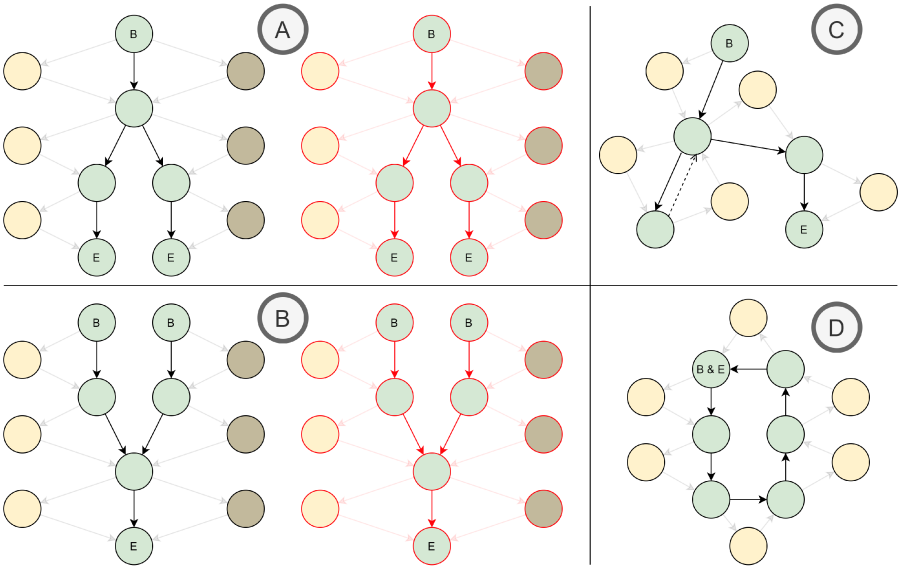


Figure 5.8: *VAKG* examples of graph patterns where B is "beginning" and E is "End". In [A], the two users began together but diverged at a certain point, and in [B], they started with different tasks but eventually converged. In [C], users backtrack using an "undo" operation, and in [D], users loop to a prior state.

Using VAKG for Analysis

VAKG's structure allows users to leverage their existing techniques to perform analysis. The *VAKG* of Figure 5.8 displays features that can occur in a *VAKG* graph and their respective meaning. Figure 5.8 shows two users performing a workflow with diverging paths [A], converging paths [B], backtrack [C], and loop [D]. The example of [D] can also be seen in the Tableau example of Figure 5.4(a) when the user interacts with a map visualization through tooltips. Knowing this, I can apply graph analysis techniques to *VAKG* to answer questions.

One of the most ubiquitous techniques for graph network analysis is PageRank [86], where it is possible to extract and rank graph nodes based on the number of their connections to other nodes. I can, for instance, extract the users' most "important" state by applying PageRank over *VAKG*'s *Machine-States*. By applying PageRank of the *Machine-States* of the Tableau example (see Figure 5.4), I find that user 1 interacted with tooltips more than any other interaction. PageRank also reveals that this specific machine state has the highest amount of *update* relationships among all *Machine-States*. Now, if I consider the full *VAKG* where both graphs of

Figure 5.4 are merged, then PageRank of all relationships indicates that the node "New Empty Viz" is the most visited node with 19 connections. By filtering the connections by user, we see that this result is mainly due to the first user, who has 8 connections to his *Machine-Update* timeline versus 4 of the second user. The same analysis can be done from the perspective of the *Human-States* to discover that the "Check Choropleth Map" node is the most connected, which leads us to conclude that the users gathered more insights from the choropleth maps than any other visualizations. It is important to note that although these results can be checked visually in Figure 5.4, in examples with hundreds of users where each performs hundreds of interactions, the use of such a ranking algorithm becomes significantly more important.

Other graph network and knowledge graph techniques and tools [103, 116, 40, 162, 243] can also be applied. A cycle detection algorithm [176] can find any closed cycles within *VAKG*. By applying it to the same Tableau example, I find that user 2 has no loops within his *Machine-States*, which means that he never retraced his steps (see Figure 5.8[D]). By applying a shortest-path algorithm [150] over the *Human-Update* nodes, I also find that the second user had fewer knowledge-related events, such as insights or questions, than the first, information which can aid in investigating if the tool is properly achieving its goals. I can also apply graph summarization [82] to simplify large graphs, apply KG completion [139] to analyze whether its users explored all the features of a *VA* tool, explore KGs through other tools [40], and analyze the KG by its embedding [245].

I also extend the same examples to analyze users' workflows while performing tasks with different tools. For instance, assuming a third user performs a similar workflow to users 1 and 2 [32, 87] but with a different tool, such as PowerBI [54]. Graph analysis through PageRank, shortest-path, and other previously discussed techniques can again be used to compare how well the two tools performed. Indeed, if *VA* tools shared a *VAKG* of their user evaluation, other researchers would be able to download them and add new data from their own users and/or tools, allowing such researchers to compare their users or tools to existing state-of-the-art tools and past users using techniques like PageRank and shortest path to demonstrate the effectiveness of their new tool in terms of knowledge gathering effectiveness,

which can potentially be used to transform the way *VA* research discusses and discloses user evaluation.

5.5.4 VAKG Evaluation Discussion

I aimed to follow existing theoretical work's example-based evaluation. However, due to the novelty of *VAKG* as a conceptual framework with modeling, *ontology*, structuring, and analysis components, no other work, as far as the authors know, can directly compare. That said, *VAKG* does not aim to supersede any specialized work in their areas. Instead, *VAKG* uses related work as its foundation. *VAKG* can also be easily extended by other works, potentially adding more data to *VAKG*'s property-maps as sub-components.

For instance, here I compare *VAKG* to Vis4ML [195], which focuses on proposing an *ontology* for ML-related tasks. Compared to Vis4ML, *VAKG* focuses on a different aspect of the *VA* workflow: the classification of *VA* taxonomy based on ownership (Human and Machine) and temporality (*state space* or process) and the relationship between them. *VAKG* proposes a knowledge graph structure and a methodology for populating the knowledge graph. Vis4ML only proposes a structure with no direct application to define or populate a knowledge graph or to define how to analyze the resulting data. Indeed, by comparing *VAKG* to all other related work, *VAKG* stands out as the only one that proposes a methodology to structure a given *VA* tool's model as a knowledge graph that can perform knowledge and *behavior provenance*. Similar results are found when comparing *VAKG* to other theoretical-focused works [43, 193, 194, 72], which are here omitted due to space constraints.

When comparing *VAKG* to the results of practical related works, I find that *VAKG* is uniquely positioned to provide a comprehensive knowledge graph for their required behavior and user-knowledge analysis requirements. However, it is important to note that this comparison is limited because *VAKG* is a conceptual framework. For instance, InsideInsights' results [152] show that allowing users to visualize the *VA* workflows of certain analysis processes is highly beneficial through interviews and usage scenarios. In the ModKT user case (see Section 5.5.3), I confirm that visualizing the resulting knowledge graph is useful. Yet, the results

of this work show that *VAKG* provides a structure for analyzing *behavior provenance* and *knowledge provenance*, as opposed to InsideInsights' report of user behavior. Indeed, in practice, most works focus on behavior analysis [242, 43, 45, 12, 42, 106, 117, 36, 62, 107, 55, 209, 21]. *VAKG* is novel in its inclusion of *knowledge provenance* as part of the resulting knowledge graph.

5.6 Limitations and Future Work

When comparing to other ontologies, it is essential to note that *VAKG*'s focus is not on its descriptive power [195], but on its ability to model and structure the user's knowledge gain process. Therefore, *VAKG* does not solve the issue of how to perform user-tracking [152]. *VAKG* also does not expand the analytical arsenal of user behavior or provenance techniques [20, 62], but provides a novel structure that is optimized for the use of said techniques for various analytical use cases. In future work, I aim to investigate the best approaches for user-tracking, behavior/knowledge provenience, and knowledge graph analysis when applying *VAKG* in domain-specific use cases. If required, I might contribute novel approaches. I also plan to investigate semi-automatic or automatic provenance techniques to assist the applicability of *VAKG*.

Although *VAKG* has focused on defining a property-map way to store the knowledge-gathering process, other works have proposed other methods as well [195]. That said, knowledge graphs are not limited to a single structure at a time, as is the nature of graph data, so it is easy to imagine that two different knowledge graphs could co-exist. Therefore, although I argued that *VAKG*'s structure is more capable than other existing ontologies, I recognize that this is mainly because the resulting knowledge graph can be extended, allowing others to use different ontologies or models as part of *VAKG* through custom property-maps or by linking *VAKG* nodes to a totally separate custom knowledge graphs. However, I believe that this integration needs to be addressed separately in domain-specific frameworks or application use cases. Results and evaluation of these future works will also be driven by their use cases. Since existing ontologies [72, 195, 61, 251, 241] can then co-exist with *VAKG*, I plan for future work to explore possible combinations of related work's ontologies as future domain-specific contributions.

I have experienced that *VAKG* can quickly result in large and complex KGs, which are hard to visualize and may cause issues related to storage space if used indiscriminately. So far, I have provided examples that were simple enough to be explained and visualized. Still, I attempted to store dozens of user workflows as a *VAKG*, and the result was too complex to visualize. Indeed, I recognize that the complexity depends on the modeled and recorded workflow, though graph network analysis is always possible. I plan on investigating better ways to visualize both simple and complex VAKGs, especially when considering what analysis is being done as future work.

The most critical limitation of *VAKG*, perhaps, is that user-tracking has been broadly seen negatively. User protection laws and initiatives, like Europe's General Data Protection Regulation (GDPR) [180] and Apple's "Ask not to track" features, are just a few examples. Although *VAKG* is not a novel way to perform user-tracking, user consent for tracked and behavior analysis is undoubtedly a relevant concern. However, this concern is not new and is shared by all related works which tackle user-tracking or behavior analysis. I also argue that in many cases, the users of *VAKG* are the same whose behavior is being tracked, which means that they probably would accept and welcome the necessary tracking since they would do the analysis. Further study is needed to analyze how impactful this would be.

5.7 Conclusion

This chapter has presented *VAKG*, a conceptual framework to structure a given *VA* tool as a 4-way temporal knowledge graph that describes user behavior and knowledge gathering during the execution of a *VA* workflow. *VAKG* proposes that by modeling a *VA* tool with its framework, we obtain a knowledge graph structure that captures the required substances from user knowledge-gathering sessions. Users then populate the knowledge graph with behavior events, such as interactions, and knowledge events, such as intents and insights. Then, the knowledge graph can be used to analyze user behavior, the knowledge-gathering process, and the interactive relationship between the two. The resulting knowledge graph is, by design, standardized across users and tools, allowing for graph-based analytics of domain-specific processes (e.g., EDA), usage patterns, and user

knowledge gain performance among multi-user and multi-tool scenarios.

In practice, *VAKG*'s resulting graph represents an overview of the *VA* workflows' usage and the collective experiences and knowledge generated by their users. *VAKG* is extensible and adaptable to various situations and domains, including its extension to incorporate other models or ontologies. Using *VAKG* as a provenance architecture, the generated knowledge graph can also be analyzed through existing graph-analytics techniques, such as visualizations, shortest path analysis, and page ranking. I applied *VAKG* to two examples: data analysis with Tableau and ModKT [184], and discussed how the resulting knowledge graph allows us to better understand the path taken by the user to reach new knowledge, how users differ in their experience of seeking knowledge, and which parts of the tool were most and least used, among other results. When compared to existing works, *VAKG* was shown to be unique in its approach in bringing *VA* model theory into practice for *behavior provenance* and *knowledge provenance* tasks.

Chapter 6

Knowledge-Decks - Automatically Generating Presentation Slide Decks of Visual Analytics Knowledge Discovery Applications [1] [2] [3]

USING Visualization & Visual Analytics (*Vi&VA*), users are able to see their data in a new light. Info-graphs and visualizations is considered a powerful method to present information, allowing users to better understand their data compared to purely depending on raw data displays, such as numeric tables, and aggregated statistics such as mean and variance. Even though this is considered central to *Vi&VA*, the sharing of insights is not often discussed. Yet, I argue that the simpler it is to share insights gained from *Vi&VA* tools, the more democratization of *Vi&VA* we would achieve. I have discussed in chapter 1 and chapter 2 of the gap between being able to gain new knowledge from a *Vi&VA* tool compared to how to collect how and what this new knowledge was in order to share it with others. With *VAKG* (see chapter 5), I have shown a method to structure, collect and store user data, such as behavior and knowledge, and exemplified a way to use the collected data. In this chapter, I present Knowledge Decks (*KD*) as an approach to use the collected user data as the means of creating shareable slide decks, which can for instance be used for presentation purposes.

6.1 Overview

Visualization and Visual Analytics (*Vi&VA*) tools allow users to harness insights and knowledge from datasets. Recalling and retelling user experiences from the usage of such tools has attracted significant interest. Nevertheless, each user session is unique, and the path between start and finish is not always linear. Even when

[1] This chapter was based on *Christino, L., Hill, T., & Paulovich, F. (2022). Knowledge-Decks: Automatically Generating Presentation Slide Decks of Visual Analytics Knowledge Discovery Applications. arXiv preprint arXiv:2212.01469.* Submitted to EuroVIS in November 2023.

[2] User Data Collection: `https://www.youtube.com/watch?v=ghe7zYPLUWs`

[3] Slide Deck Generation: `https://www.youtube.com/watch?v=Yhoj7fygMIw`

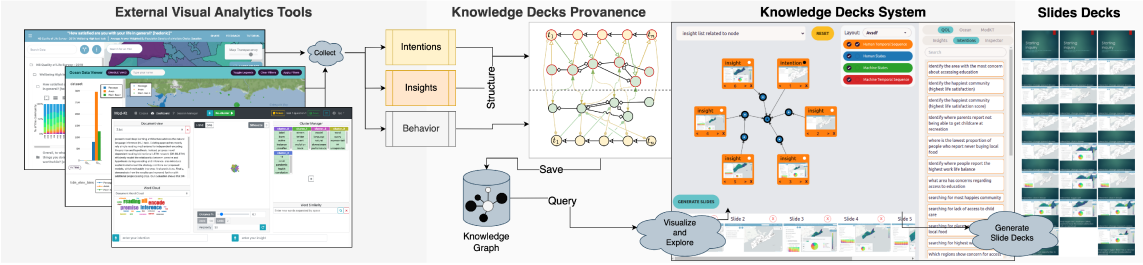


Figure 6.1: *Knowledge Decks (KD)* is a systematic approach to collect and structure user intention, behavior, and insights from external *Vi&VA* tools as a Knowledge Graph. The result can be visualized through an interface that displays a semi-layouted node-link diagram of knowledge discovery paths. The knowledge path displayed is linearly read based on user-selected parameters to generate linear Slide Decks that retell the knowledge discovery stories of the external tools' users.

different users have the same intention when using a tool, they may follow different paths and uncover different insights. Currently, those who want to analyze user data are expected to manually collect and process user data. This process is time-consuming, especially when there is the need to gather users' insights and behavior and share them with others. I present *Knowledge Decks (KD)*, a systematic approach that collects user intentions, behavior, and insights during knowledge discovery sessions, automatically structures the collected data as a Knowledge Graph, populates an interface to visualize Knowledge Pathways, which I call *storylines*, as an intelligently layouted node-link diagram, and generates linear narrations of the knowledge discovery process as PowerPoint slide decks. To evaluate *KD*, I have attached it to two existing *Vi&VA* tools where users were asked to perform pre-defined tasks. The *KD* interface was then shown to experts in the tools. Interviews with the experts showed the relevance of *KD* in both commercial and research applications when investigating how each tool was utilized when automating the collection of insights and for the quick generation of slide decks containing screenshots and texts from the users' point of view.

6.2 Introduction

Visualization and Visual Analytics (*Vi&VA*) tools allow users to harness insights and knowledge from datasets [193]. In general, through such tools, users wish to answer existing questions, execute a pre-planned task, or aim to explore the

information contained in the data. In all cases, the user performs actions and interactions considering an initial intention until they reach their goal, such as uncovering new insights. This process of user-guided knowledge discovery, also called *Vi&VA*'s knowledge-gathering workflow, is extensively studied. Theoretical knowledge models [193, 72], behavior collection techniques [152, 98] and behavior analysis [107, 20] are recent works which attempt to study how to understand and analyze user-guided knowledge discovery processes. The results of this analysis are used for various means, such as user evaluations in research [20] or to generate output media, such as reports or slide decks [171].

Among the behavior collection techniques, the collection process most often used is provenance [177]. Provenance can be used to record, which can be used to recall the story of what and how users found a new piece of knowledge [250] for the direct benefit of the users themselves [250] or to another party. After collecting data through provenance, the result is by and large manually formatted into datasets or visualizations to be included in *Vi&VA* tools [251] or formatted into non-interactive media, such as info-graphs reports [128, 264], or slide decks [171]. Indeed, slide decks are yet to be dethroned as the best way to give presentations of such results [200].

However, typically, each user-guided knowledge-gathering session within a *Vi&VA* tool (henceforth called *storyline*) collected through provenance is unique and non-linear. Even when different users have the same intention when using a *Vi&VA* tool, they may uncover different insights due to their differences in expertise [72]. Likewise, even when the same insight is found, users may have used different ways to arrive at it [107]. These inconsistencies become apparent when analyzing the collected provenance data [250]. Then, the process of transforming the data from its original format, such as video/audio/logs recordings of surveys [250], into a cohesive data structure that can be analyzed or shared [128, 141] is time-consuming [171, 200]. Additionally, summarising the *storyline*s collected into a media format that tells a story, such as a slide deck, is a challenging task on its own as well [171, 128], especially when considering that the person who will read and use the slide deck may have varying degrees in expertise on the *Vi&VA* tool. Furthermore, if the *Vi&VA* tool is updated, a new loop of behavior collection, analysis,

and transformation would be required, adding to the time already consumed and the complexity of the task [250, 72].

In this chapter, I discuss, design, and implement *Knowledge-Decks (KD)*, a novel approach that collects user *storyline*s during knowledge discovery sessions [4], transforms the collected data into a Knowledge Graph and provides an interface where one can explore and extract auto-generated narrations of users' *storyline*s as PowerPoint slide decks [5]. *KD* bridges the gap between provenance and the generation of slide decks out of the aggregated user *storyline*s. For this, *KD* collects user *intentions*, user *interactions* within a *Vi&VA* tool due to that intention, and the *insights* gained along the way. By mapping the collected data into a Knowledge Graph (see chapter 5), *KD* displays a visualization tool to query and explore parts of the Knowledge Graph, select *storyline*s of interest, and generate PowerPoint slide decks out of the selected *storyline*s.

KD is evaluated by being applied to two different *Vi&VA* tools. After giving tasks to users, all user intentions, interactions, and insights were collected and structured, and the resulting *storyline*s were explored through the *KD* interface by experts of each tool. Each expert was interviewed regarding how their current workflow is when they need to use their tool to find an answer to a question, how they format it into a presentation to be shared with others, and how *KD* would impact their work when making the same process. The experts were tasked to compare *KD* to their own process of creating presentations, training new hires, performing surveys with users, and sharing insights or *storyline*s with peers. Although similar introductions were given to the experts, each expert tested *KD* by taking into consideration their own scenarios, which consequently showcased different *KD* use cases. One expert focused on using *KD* to generate slide decks about the explored data, such as collecting lists of insights, and the other generated slides about the interactions done in their tool. Interviewees said that through *KD*, they could better understand their own users and how each tool feature is used, significantly accelerating the generation of screenshots and slide decks to communicate of novel insights and unexpected behaviors to their peers and superiors.

[4] Data Collection Video: `https://www.youtube.com/watch?v=ghe7zYPLUWs`
[5] PowerPoint Generation Video: `https://www.youtube.com/watch?v=Yhoj7fygMIw`

6.3 Background and Related Work

Users of Visualization and Visual Analytics (*Vi&VA*) tools have knowledge-gathering sessions (*storyline*s) that *Knowledge Decks (KD)* aims to collect. The literature in *Vi&VA* knowledge modeling has shown that these *storyline*s can be understood as an iterative workflow of user intentions, behaviors, and insights [193, 194, 72]. For instance, Federico et al. [72] describe how *Vi&VA* tools and frameworks can be modeled following knowledge modeling methodology. Their methodology of describing *Vi&VA* as an iterative workflow of events between Human and Machine actors shows that it is possible to understand a *Vi&VA* problem as a sequence of events [195]. This model also includes events not triggered by users, such as automatic processes (e.g., data mining and machine learning). Yet, developers and researchers usually use more practical means to model, collect, store, and utilize their users' *storyline*s. For instance, some researchers collect and analyze user behavior using provenance [62, 177]. Different from more application-focused works, *KD* uses the theoretical work of interpreting user *storyline*s as user intentions, behaviors, and insights and applies behavior and knowledge provenance techniques to collect the requisite data (see chapter 5).

To populate the *KD* with user data, one must first perform provenance by collecting information from the user. Existing works have discussed the collection of machine-related events, such as changes in datasets [62], updates in visualizations [20, 251], and other similar events. In a *Vi&VA* tool, these events either occur due to user interactivity or due to some automatic process [72]. By capturing sequences of such events, one can recall user behavior during a user-guided knowledge-gathering session within a *Vi&VA* tool (*storyline*) [250]. On the other hand, tracking user's *Tacit Knowledge* as defined by Federico et al. [72] is an emerging problem which is either done by manual feedback systems [21, 152], by manual annotations over visualizations [208], or by inference methods that attempt to extract users' insights by recording the users' screens, video or logs. The collected user information is then, generally speaking, manually formatted and analyzed as a post-mortem task [20, 99].

Among the alternatives, InsideInsights [152] is an approach that skips part of the manual process, since it collects user insights from annotations that users

input during their analytical process. InsideInsights demonstrated that collecting user annotations is a legitimate way to extract and store user insights. *KD* is similar to existing works in its attempt to collect user interaction [20] and to use feedback collected from users, such as textual annotations of insights [152, 250] as provenance methods. Yet, *KD* goes beyond in its novel "collective knowledge gathered" exploration tool, which displays relevant portions of the collected data for *storyline* discoverability. *KD* is also unique in its proposal to generate a linear PowerPoint slide deck from one or more *storyline*s.

After collecting the data, *KD* structures it into a format that can be queried. Landesberger et al. [241] similarly perform behavior provenance by modeling user behavior as a network graph (or network diagram), which is then used for analysis. Other works also use network graphs as a way to model knowledge [74], to perform data provenance [204, 62], to perform insight provenance [90], and to perform analytic provenance [250, 148]. Out of the variations of network graphs, a subtype uniquely positioned to model the user's *storyline*s is the *Knowledge Graph (KG)* [74, 98] due to their specific applicability in associating two kinds of data: temporally-based event data [98] and relationship-centric data [12]. As *Vi&VA* is modeled based on events [193, 72] and the goal is to find and analyze the *relationship* between events [82] in order to extract *storyline*s out of the KG, I decided to use KGs as one of the cornerstones of my approach. In this way, *KD* is novel in its usage of Knowledge Graphs as a way to not just structure users' *storyline*s, but structure them in a queryable format such that *KD* can provide visualizations of individual *storyline*s, enabling their transformation into slide decks.

Specifically related to KG visualization, works like CAVA [40], enable the exploration and analysis of KGs through visual interaction. KG4Vis [136], on the other hand, uses the advantages of the knowledge graph structure to provide visualization recommendations. ExeKG [262] uses KGs to record execution paths within data mining and machine learning pipelines. Indeed, there is an ever-increasing number of works using KGs for visualization or analysis purposes in *Vi&VA*, many of which say that KGs are suitable for knowledge analysis and extraction [137]. *KD*'s novelty among such works is the usage of KGs to structure users' *storyline*s in a queriable format and to simultaneously use the *KD* structure

as a visualization target of portions of the graph that relate to a specific *storyline*. This approach is what allows the explicit mapping between intentions and insights, which is then used to generate the slide decks.

Hardly any research has attempted to model user behavior or the associated knowledge discovery. Instead, info-graphs are one of the most used methods to visually encode a *storyline* [128]. Indeed, Zhu et al. [264] details how info-graphs can be automatically generated through machine learning [46] or pre-defined rules [84]. When applied to storytelling, displaying insights as visual or textual annotations on top of visualizations has been the aim of several works [84, 47, 46]. Although their results are very relevant for the narration of *storyline*s, they only list user insights, not the process of how a user might have reached them. Instead, slide decks are better suited to narrate the events that led to the insight [200].

Regarding slide decks, StoryFacets [171] is a system that displays visualizations both in the dashboard and slide-deck formats. Although slide decks have well-researched limitations [128], StoryFacets also argues about how and why slide decks are advantageous when narrating a sequence of events. Other works have succeeded in recalling and retelling the knowledge discovery process in other means [250]. For instance, Ragan et al. [177] list many tools that collect and structure user behavior and insights in a queriable format. Nevertheless, they have not proposed a means to generate slide decks from user *storyline*s. In this context, *KD* is novel in how it provides users the ability to generate slide decks from the data automatically collected, structured, and visualized.

6.4 Methodology

This section discusses the goals and how I transform provenance data into slide decks.

6.4.1 Context and Goals

To best explain and contextualize the approach used, I use a *Vi&VA* tool called the Well-being Mapping Tool (WMT) [50] as a running example. WMT was developed to support the not-for-profit organization *Engage Nova Scotia (EngageNS)* [93] to analyze data collected as part of the 2019 Quality of Life survey. This survey is

composed of 230 questions about the well-being of residents across the province of Nova Scotia, Canada. Almost 13,000 people responded. WMT allows for the visualization of the entire survey through maps and charts. Although different stakeholders have successfully used WMT to gain insights, sharing or recalling these insights requires significant manual effort. The usual process used by experts from EngageNS involves taking screenshots or recording their screens as they use the tool and manually annotating the insights and how they were found. Although this is a typical process, *what if there is a more automatic way to record users storyline (see Definition 6.4.1) and generate shareable media with the relevant gathered knowledge?*

Definition 6.4.1 (Storyline)**.** *The process taken and results obtained by users during user-guided knowledge-gathering sessions within a Vi&VA tool.*

Bridging or automating the gap between knowledge generation and final presentation is not only a challenge for the users of the WMT tool, but it is potentially an issue for users of any *Vi&VA* tool. To address this challenge, I present *Knowledge-Decks (KD)*, an approach to support the generation of slide decks from provenance data. In more concrete terms, the objective is to ***generate slide decks for presentation, which are linear in nature, from non-linear interwoven storylines that are automatically collected from (potentially multiple) users of a Vi&VA tool.***

Definition 6.4.2 (Provenance)**.** *Tracking and using data collected from a tool's usage.*

Considering this context and based on existing methods discussed in Sec. 6.3, I identified the goals as:

G1: Given a *Vi&VA* tool and users who can describe their intentions and insights while interacting with the tool, *KD* should capture and structure the users' *storyline* provenance;

G2: Given the provenance data of one or more users, *KD* should allow for the exploration and visualization of the collected *storyline*s;

G3: Given a *storyline* of interest, *KD* should format it into a shareable format for downstream tasks such as presentation, storytelling, etc.

To implement such goals, *KD* defines a strategy to collect provenance data during user sessions to constitute a *storyline* (G1). Then, it defines an approach

to forward the provenance data to a structured database (G1) accompanied by a querying mechanism to visualize *storyline*s out of such database (G2). Finally, it defines an approach to allow users to explore and download the *storyline*s of interest as draft PowerPoint slide decks (G3). This entire process is exemplified by Figure 4.5 and will be discussed in more detail in the following sections.

6.4.2 Defining Knowledge-Deck Storylines

To extract *storyline*s from a collection of user experiences (G1), I followed the steps of existing works that define the content of a *storyline* [72]. The definition of a *storyline* used in *KD* uses three main concepts: users have an *intention* when using a *Vi&VA* tool. Once the user has the *Vi&VA* tool open, they perform *interactions*. After any interaction sequence, users acquire new *insights*, which might or might not lead to new *intentions* and, consequently, new *interactions*, defined by

Definition 6.4.3 (Intention)**.** *A question, the will to discover something new, or some reason that led the user to use the Vi&VA tool.*

Definition 6.4.4 (Interaction)**.** *Events within a storyline, such as clicks, filters, or scrolls.*

Definition 6.4.5 (Insight)**.** *An answer or new contextualized finding retrieved by the user by using a Vi&VA tool.*

To formalize these three concepts, consider the model of Federico et al. [72] as seen in Figure 6.2, where circles represent processes and boxes containers of continuously accumulated and accessed data. The nodes definitions are: visualization V, perception and cognition P, exploration E, data D, and specification S, tacit knowledge K^T, explicit knowledge K^e, and automatic analysis methods A. For example, considering our running example, I could say that the WMT tool shows a map visualization V, generated from the survey data D, and the algorithm to transform it into the visualization S. The user can then look at the map P, explore it with their mouse E, and learn something new K^T.

From this, I can define the three concepts of *intention*, *insight*, and *interaction* as such

Definition 6.4.6 (Insightful Loop)**.** *The process which the user took from perceiving information (P) and due to tacit knowledge gained from it (K^T), interacts with the tool (E).*

Definition 6.4.7 (Insight-less Loop)**.** *The process that the user took from perceiving information (P) and no new knowledge was gained from it before the next interaction with the tool (E).*

Considering this formalization and following the *KD* process (Figure 4.5), the first step is to define which data has to be collected during provenance so that the *storyline* defined above can be reconstructed. For that, *KD* follows the methodology of VAKG (see chapter 5), which is a conceptual framework that structures *storyline*s into temporal sequences of human and machine events and, additionally, relates the events through a state-space Knowledge Graph [98]. This way, I extract a Knowledge Graph structure seen in Figure 6.3 from the model of Figure 6.2. The structure of Figure 6.3 categorizes each concept by its ownership (human or machine) or by its timing (state or update). The four classes of nodes used in *KD*'s Knowledge Graph are: Human-Update, Human-State, Machine-State, and Machine-Update. The relationships between each class are: links [1&2] relate the previous human/machine state/update to the current one, [3&4] represent where a change in K^T leads to an update in P, and [5&6] similarly represent where a change in K^ϵ, D or S leads to an update in V or A in the Machine-Update or in X, P or E in the Human-Update. The final two relationships (Figure 6.3[7]) synchronize the two state spaces.

KD is then responsible for collecting data related to all nodes highlighted in orange shown in Figure 6.3. The legend of Figure 6.3 also shows that any new perception P is collected from the user as an *intention*, any new exploration E is collected as a new *insight*, and any new *interaction* which causes changes in the tool's specification S, visualizations V, or triggers an analysis A (e.g., runs a statistical model) should be captured by *KD*. With this definition, we can now specify what data needs to be collected from users to perform the provenance of the three aspects: interaction, intention, and insight.

On the behavior provenance, examples of interaction events expected to be collected from the tool are filters, selections, and aggregations being applied at any given time and the state of each visualization, such as panning position, zoom, or the selection of which part of the data is to be visualized. Assuming the running example of the WMT tool, *KD* is responsible for collecting any changes to the tool

when users: pan/zoom the map, select a survey question to be shown on the map, select areas in the map for filtering, and change any of the available visualization options. Further examples are discussed in Sec. 6.5.

On the HUMAN side, *KD* expects to collect the knowledge provenance, that is, intentions and insights from events happening on or within the user, such as new perception *P* or exploration *E*. The approach taken by *KD* deviates from some related works that capture user interactivity through an external method of observability, such as through video/audio recordings and thinking aloud sessions [20, 99], which requires manual post-processing by the one doing provenance. Instead, *KD* follows an integrated approach [152, 21] by requesting user inputs of their intentions and insights within the tool itself, which can be done in modes by either typing or speaking. Users are also allowed to draw shapes on the screen to indicate if there was anything in the tool that led them to the intention or insight. Assuming the WMT example, users would be expected to open the tool and type their intention, such as "What is the quality of life in the capital of the province?". This intention would be written by the user prior to any interaction, and once any insight is found, they would report their findings and, optionally, draw what in the tool led to the insight.

Following our running example, I can now formulate what is collected from users of the WMT tool as part of the provenance process:

HUMAN Update: label (insight or intention), created time, URL, screen size, text, keywords, shapes drawn, user id, analysis id

HUMAN state-space: label (insight or intention), created time, last updated time, keywords time

MACHINE state-space: label of what the event is, created time, last updated time, the status of the hierarchical structure, mapped variable, bar charts parameters, map position/zoom, areas selected in the map, math operation used

MACHINE Update: event name, created time, URL, user id, analysis id, and all the same data from the related Machine state-space except user id and created/updated time

Once the data is collected, *KD* uses the structure of Figure 6.3 as a schema for

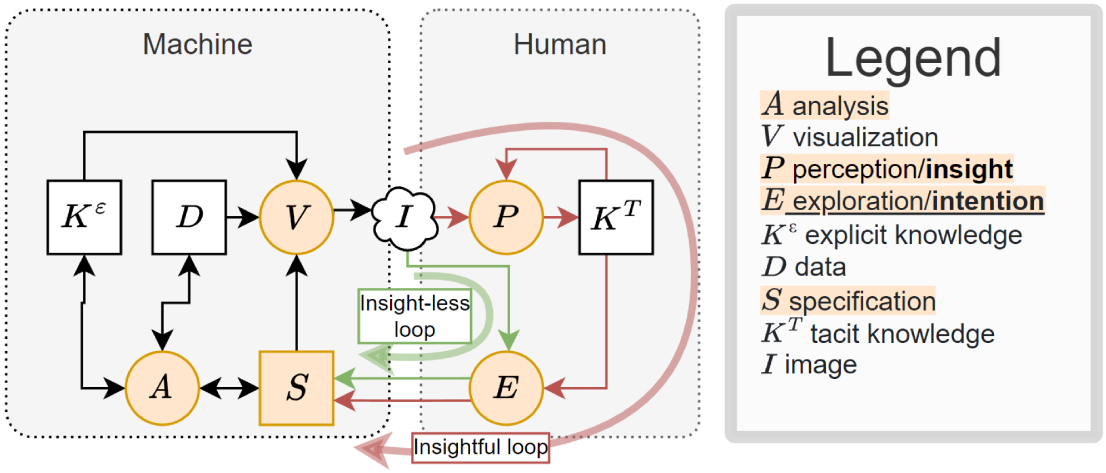


Figure 6.2: Simplified knowledge-assisted model. In red from V to E is the **insightful flow** where the user has a new insight, while in green is the **insight-less flow** when the user performs an action without any new insight. The two parallel red and green links between E and S represent the user's intent when a new specification S, such as an interaction with some visualization, is made. This causes the execution of some analysis algorithm A and/or the rendering of a new visualization V. Nodes in orange identify which data will be tracked in the approach.

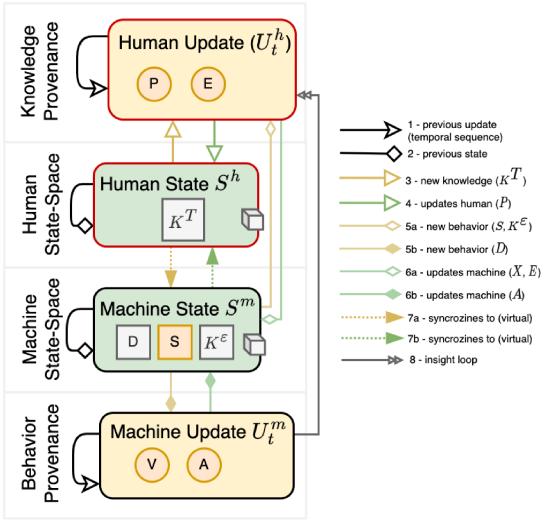


Figure 6.3: Knowledge graph structure used by *KD*. The Four lanes of the knowledge graph represent the four classes and their relationships. *KD* uses this structure when collecting and populating the knowledge graph from the user's interactivity (behavior provenance), insights, and intentions (knowledge provenance).

storing the *storyline* data.

6.4.3 Collecting and Structuring Storylines

With the knowledge graph schema, I can now discuss how *KD* is attached to *Vi&VA* tools for data collection and provenance. The *KD* approach is an extension of the VAKG sample made by [51]. It is made of three components. The first two are a data collector and a structuring process following the definitions of Sec. 6.4.2. The third is an interface for users to visualize the *storyline*s, select parameters for slide-deck generation, and download the auto-generated slide decks.

Knowledge Graph Database. First, to collect and store the data, the implementation must receive and store events in a database while following the design of Sec. 6.4.2. I chose Python as the language of choice and an API-based design for their wide usage within the community. Additionally, I use Neo4J [162] as the graph database due to its wide usage in knowledge graph applications. Due to these choices, any *Vi&VA* tools using the knowledge graph schema can connect and send their events to the same database if their developers wish to.

Collecting Interactions (Behavior Provenance). Next, to populate the machine-side events, *KD* must keep track of the *Vi&VA* tool's state at all times (behavior provenance). For this, I limited the scope of *KD* to web-based *Vi&VA* tools to define how to attach *KD* to existing tools. *KD* provides a publicly available [51] JavaScript library with which web-based *Vi&VA* tools can inform *KD* of any new events. In short, the library implements two methods with which the *Vi&VA* tool informs *KD* of any new interaction or change within itself following Sec. 6.4.2. This requires the *Vi&VA* tool to inform its current state as a hashable JSON and its stateful URL. This way, *KD* can recall the state of the *Vi&VA* tool and include it in the generated slide decks.

Collecting Intentions and Insights (Knowledge Provenance). Similar to the interactions, the library also provides ways for the *Vi&VA* tool to send intentions and insights to *KD* (knowledge provenance). Here, *KD* expects users to type or speak their insights and intentions and optionally draw on the screen as described in Sec. 6.4.2. *KD* extracts keywords from the text, and the resulting combination of keywords plus text is sent to *KD*. To generate such keywords, I use the natural

language processing library *Spacy* [238] to retrieve the list of lemmas from a given natural language text.

From preliminary tests, I found that a more reliable way to compare a new input to previous texts (e.g., insights or intentions) is to request confirmation from the user. I also noticed that short, concise texts were better suited to *KD*'s goals than larger ones because users would spend too much time writing/reading texts instead of focusing on their goals with the tool. Therefore, *KD* limits the number of characters allowed per text to a certain amount α, and once the text is typed, *KD* gathers the top similar texts and displays them back to the user, asking whether any existing insight or intention is similar to theirs. To calculate this text similarity, *KD* uses the text's keywords (lemmas) and scans the database for similar keywords among the existing insights and intentions already recorded in the Knowledge Graph database. The similarity measure used is a cosine similarity measure calculated from a word2vec representation of the keyword list and is also implemented by *Spacy* [238]. If there is any match with a score greater than a certain threshold β, *KD* collects the texts that generated these keywords, which can be accessed from its neighboring HUMAN temporal sequence node, and display up to γ of these texts to the user ranked by similarity score. These three adjustable variables were selected as $\alpha = 75$, $\beta = 0.8$, and $\gamma = 5$ during preliminary tests but can be modified by the user. The user can finally specify whether their intention or insight is equivalent to any existing one or if it differs from them all. This user-feedback step provided a reasonably reliable way for *KD* to better match new intentions or insights with previous ones compared to purely automatic processes.

Additionally, *KD* also asks for an additional optional input: whether there was an existing element in the *Vi&VA* tool, such as visualization, text, or color, which caused the user's intention or insight. The user is given the ability to draw shapes on the screen in this step, which is recorded and included in the slide deck generated by *KD*. Users can also say that nothing in the interface caused the new intention or insight. The drawn elements and the current URL of the website are saved as part of the user's HUMAN temporal sequence to be used when generating the slide decks.

With this, the user's perspective of using a *Vi&VA* tool equipped with *KD* is as

follows: the user opens the tool due to some intention and types it as a new entry within the intention text field. The tool displays similar intentions from among all previous users and asks if the intention is new or equivalent to a previous one. If the user finds a previously typed intention similar to theirs, the user selects it. Otherwise, the user flags it as a new intention. The user interacts with the tool and finds an answer. The user types the answer in the insight text field and once again selects from the list of previously typed insights whether or not any of them are similar to their own. The user is also given the option to draw a shape on the screen to indicate what caused this new insight, to which the user draws a circle over part of the tool's visualization. At the end of this example, the KG stores the user's intention, including the typed text, keywords, and the website URL. Then, after each interaction event, *KD* records the sequence of events, including the URL, to reproduce each state of the tool. Finally, *KD* records the final insight, including text, keywords, URL, and drawn elements. A video for the WMT tool is provided, which exemplifies this flow [49].

6.4.4 Knowledge-Decks

So far, I have discussed how *KD* collects and structures the users' *storyline*s. *KD* provides an interface displaying visualizations from the knowledge graph to query and visualize the recorded *storyline*s. This part of *KD* was designed with both **G2** and **G3** in mind, where users should be able to visualize the narration of users' knowledge discovery stories (G2) and generate slide-decks out of them (G3). The *KD* visualization is publicly available [51].

Querying Storylines. The *KD* interface exemplified in Figure 6.4 demonstrates how *KD* displays the visualization of the "insight list related to node" (A) question given an intention selected from the to the right (C). A *node* refers to an intention or insight that the user selects to answer the question. *KD* has three built-in questions, each of which has an insight-related variant and an intention-related variant:

Q1 Insight/Intention list related to node;

Q2 Closest insight/intention related to node;

Q3 Interactions from closest insight/intention which lead to node;

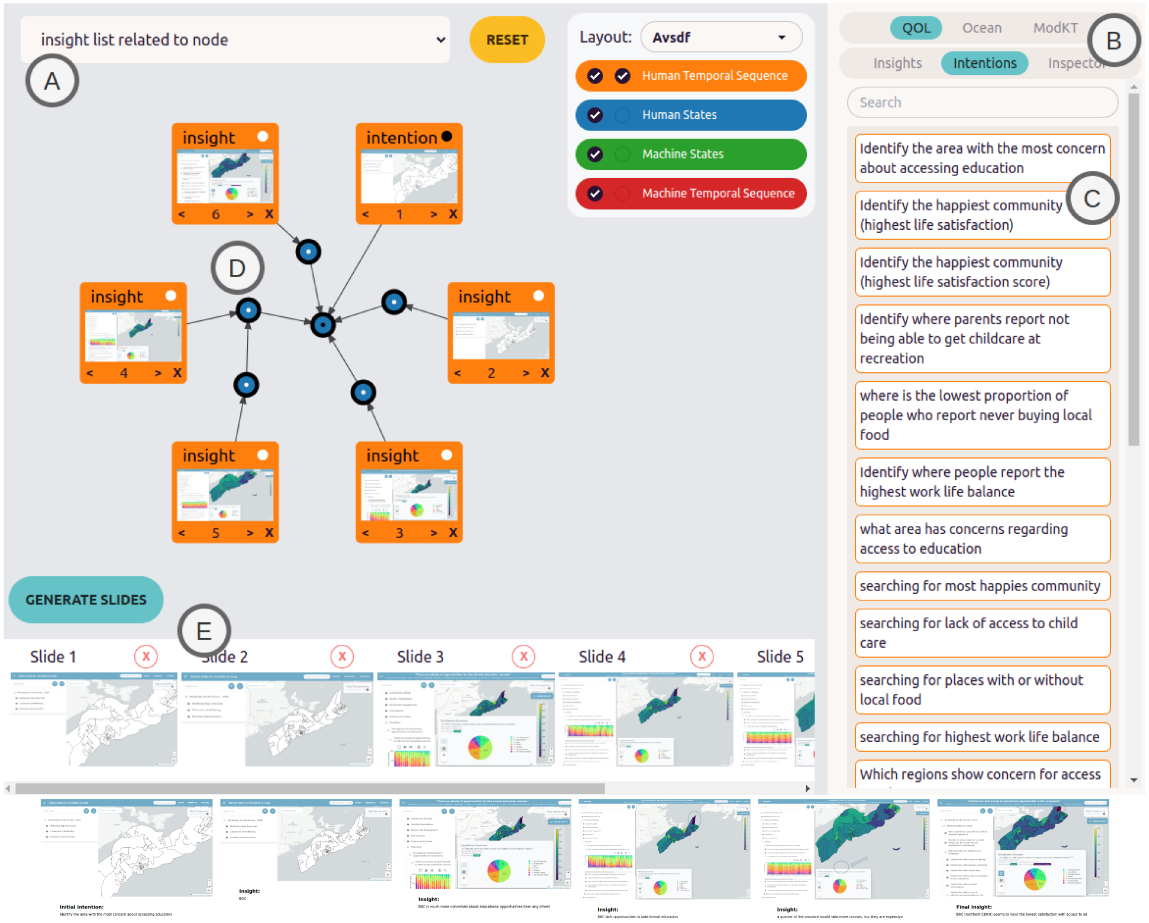


Figure 6.4: *KD* visualization interface of the data collected from the WMT tool. Users can visualize sections of the *KD* knowledge graph by selecting a type of *storyline* [A], filtering among the available data [B], and selecting an intention or insight [C]. *KD* extracts the relevant section of the knowledge graph and displays it as a node-link visualization [D]. *KD* also automatically collects a picture of the VA tool at the state of each node in the graph and displays it in the visualization [D] and the slide deck preview window [E], from which users can reorder/remove slides and download the PowerPoint file. The slides of the generated file are shown below the tool's screenshot.

KD lists all insights and intentions of the knowledge graph in (B), allowing the user to filter the lists by *Vi&VA* tool and insight/intention. Once the user drags one of the intentions or insights into the main visualization, *KD* extracts the *storyline*s from the database to display as a node-link visualization. Each question above has *four* associated database *queries*. The "insight list related to node" query (Q1) is exemplified below:

```
MATCH p=(n1)-[:UPDATES_HUMAN]-()-[r:PREV_STATE*0..10]
    -()-[:UPDATES_HUMAN]-(n {label: "insight"})
WHERE id(n1) in [{{ids}}] and n.tool in [{{tools}}]
RETURN p
```

Through this query, defined through the cypher query language [79], *KD* extracts all paths p from node $n1$ connected to node n of type *insight*. This path navigates through the *Human State* part of the knowledge graph with up to 10 hops, which limits the search space of the database, allowing for better performance. When running *KD*, these parameters can be manipulated by modifying the cypher queries. *KD* also filters the results with a $WHERE$ command so that the node $n1$ matches the node (insight or intention) that was selected by the user in the previous step. The question $Q1$ also has three more queries, one where n is type *intention*, one to show only the *insights* that are from the same user as the selected node, and one to show only the *intentions* that are from the same user as the selected node. Due to limited space, all other queries can be seen in the project's open-source repository [51].

In order to visualize the user *storyline*s, the *KD* interface converts the results of queries into node-link visualizations. The example of Figure 6.4 shows *storyline*s collected from WMT. In this specific query, five insights in orange originated from the selected intention "identify the area with most concern about accessing education" containing five insight nodes. This visualization follows the schema defined in Sec. 6.4.2, where orange nodes are the Human Temporal Sequence (Behavior Provenance), and blue nodes are Human States. Similarly, the arrows and their directions follow the relation specifications of Sec. 6.4.2. By using *Selenium* [88], *KD* takes screenshots to populate the visualization.

The label panel on the center-top of the screen can be used to control the node-link layout, allowing users to have the option of a pure force-scheme layout or fixed layouts implemented in Cytoscape.js [81]. *KD* allows its users to customize the visualization by defining which nodes should be fixed or which should follow a force-scheme layout. *KD* also allows the user to select the node-link layout algorithm to be used. In other words, the layout being used in Figure 6.4[D] makes orange nodes match an *avsdf* layout and leaves the blue nodes to be positioned through *force-scheme*, which allows for the above questions to be much more visible than pure force-layout and fixed layouts (Figure 6.5). Users can also reposition any of the nodes with their mouse.

If any nodes within *KD* are clicked, extra information about the nodes is displayed in an inspector panel, such as in the right part of Figure 6.5. Within the specific example of Figure 6.4, if the user clicks on each of the insight (orange) nodes and checks the inspector panel, they could discover that four different users were involved in these insights.

Yet, this vastness of layout options with no optimal default set made the visualization too hard to use for the proposed goals. Additionally, the node-link displayed was, at first, too large or complex to help explore or select *storyline*s due to the large amount of clutter. Therefore, in order to optimize *KD* for the exploration of *storyline*s, two main additions were needed: simplify the node-link graph by hiding parts of the knowledge graph and the definition of a default layout to be used.

To solve the first point, *KD* only retrieves the nodes related to the query being used. There are four types of nodes: orange, blue, and green, but the visualization in Figure 6.4 only shows two of these types of nodes because *KD* limited the scope of the node-link to the query being made. Namely, the question $Q1$ of Figure 6.4[A] and the query $Q2$ focuses on the relationship between *insights* and *intentions*, therefore *interactions* (nodes in green and red) are not included in the results.

However, the query $Q3$ exemplified in Figure 6.6 does need all four node types to be visualized since it relates interactions (nodes in red) to intentions and insights (nodes in orange). Additionally, although all relations described in Sec. 6.4.2 could

be displayed in the node-link visualization, the output of question $Q3$ does not need to display all available relations to achieve its purpose of describing the interaction *storyline*s to the users. Therefore, the visualization of $Q3$ hides the relations 4, 5_a, 6_a and 8 of Figure 6.3, reducing visual clutter while providing the user with the information needed to examine the user interactivity *storyline*. Indeed, optimizations were done for each query to display only relevant information in the node-link. All these optimizations are part of the cypher queries themselves. Users can, however, edit these specifications by modifying or adding new cypher queries [51].

The second point is solved by applying a predefined layout combination to each query type. In Figure 6.4, we can see that for the query "insight list related to node", the predefined layout was *avsdf* [81] applied to the *orange nodes*. Queries $Q1$ and $Q2$ follow this specification. This setting can be seen in the color legend checkboxes on the top-center of Figure 6.4. This choice of combination was made because of the circular nature of *avsdf*, which caused the blue nodes to naturally rearrange themselves using *force-scheme* to better cater to the outer circular layout of the orange nodes. The query $Q3$ of Figure 6.6, however, uses another combination where all nodes use the *Cose-Bilkent* layout [67]. These predefined combinations are not fixed, and users can modify the selections, as is shown in Figure 6.5.

The end result of these optimizations was that users were more clearly able to see and explore the visualization. For instance, in Figure 6.4, the user can count the number of *hops* between the intention and each insight, allowing them to conclude that four insights were obtained right after the selected intention (3 hops), while the insight of slide number 5 took a bit more (4 hops). Before the optimizations, the nodes were not clearly separated, and many other nodes and relations were also being shown, causing users to be unable to reach the conclusions described above.

KD also uses the graph structure to automatically generate a linear order from the resulting graph by using a breadth-first graph algorithm to prepare the *storyline*s for generating a slide deck. This order is displayed in both the numbering of the node-link visualization and the slide panel (Figure 6.4[E]). The user may wish to reorder the slides to, for instance, match the *number of hops* discussed above or keep the default order, which tells the story where insight 5 came right after

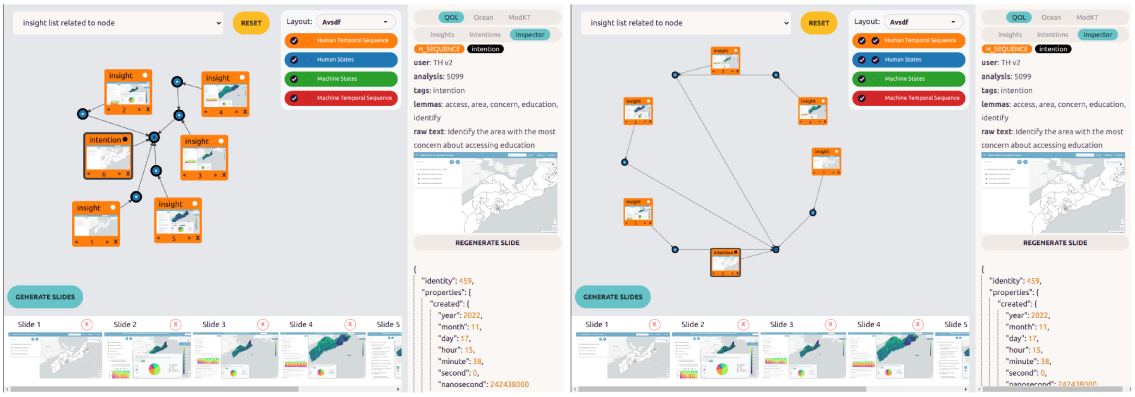


Figure 6.5: *KD* interface example where the node-link is placed following only force scheme (left) and fully static following the avsdf layout (right).

insight 4. Reordering can be done through the arrows within the node-link or by dragging and dropping the slides in the slide panel. Users can also remove slides from the slide deck or re-include previously removed slides in either panel. Finally, users can generate and download the slide deck by clicking the "generate" button, which will populate a PowerPoint presentation with the images displayed, the relevant texts of the intentions, insights, and behaviors, and any shape drawn by the user in the data collection step, as is shown in Figure 6.4. Once a slide deck is generated and downloaded, users can edit it through presentation software like MS PowerPoint [6].

I discussed queries *Q*1 and *Q*2 while focusing on *storyline*s that relate insights and intentions. Following that, *Q*3 is the one that allows the visualization of *interaction storylines*. The example of Figure 6.6 shows a *storyline* that there were ten interactions (red) between an intention and an insight (orange) within the WMT tool. The intention found was "Identify the area with the most concern about accessing education", while its corresponding insight was "B0C (Northern CBRM) seems to have the lowest satisfaction with access to ed". Like before, users can click on each node to investigate the exact contents of each node through the inspection panel, which allows users to verify the step-by-step interactions done to the tool in each step. The user can also reorder or exclude the nodes from the slide deck generation, and once generated, users will have a slide deck with step-by-step screenshots of their tool, the intention/insight texts written, and any annotations

[6] As seen in this example: `https://youtu.be/Yhoj7fygMIw`

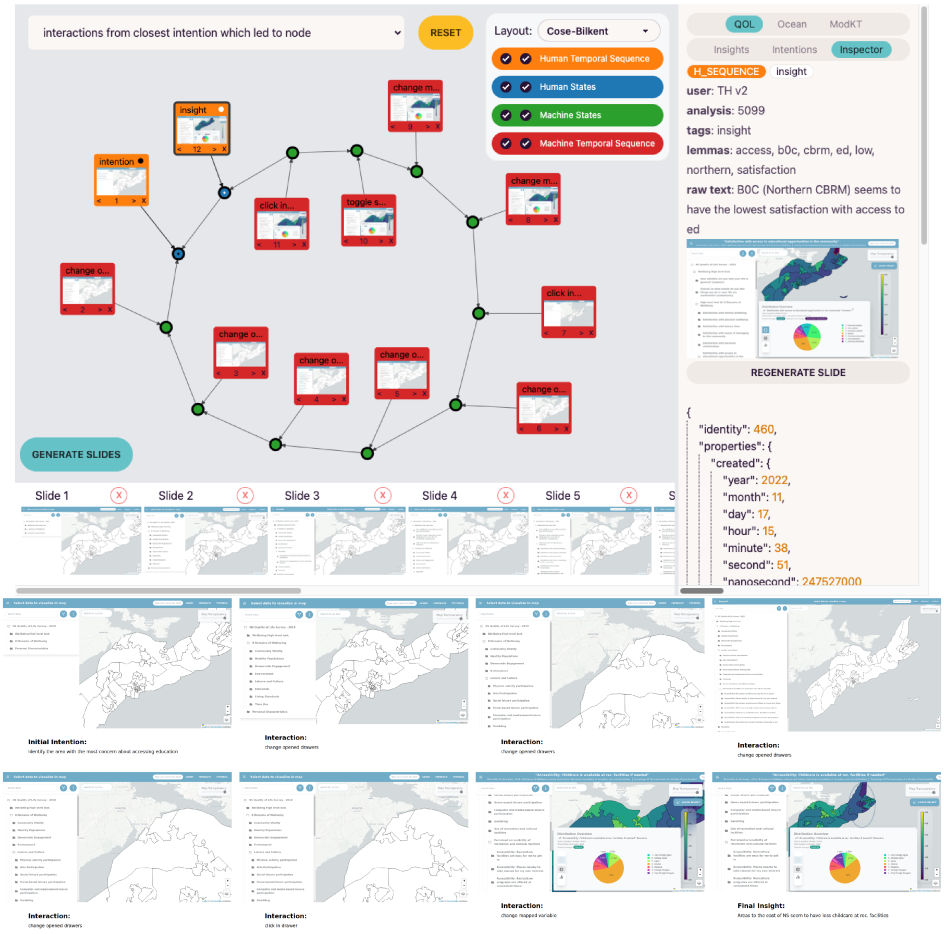


Figure 6.6: *KD* interface example a node-link diagram shows all interactions between intention and insight, and below is the generated slide deck.

drawn.

6.5 Use Case of the Well-Being Mapping Tool (WMT)

So far, I have used WMT as a running example to aid in defining and describing *KD*. In this section, I expand the example into a proper evaluation of *KD* when applied to WMT, which was done in cooperation with an expert from the tool's owner, Engage Nova Scotia.

To collect the required data, I asked 9 participants to perform pre-defined tasks with the tool to collect data and build the knowledge graph. A survey with a demographics questionnaire and 5 tasks were given to the users. During the tasks, participants were asked to write or speak about their intentions and

insights, whether they were part of or not related to the task. Indeed, all tasks were exploratory in nature, and participants were asked and encouraged to browse the tool if they so wished. The tasks were used to provide a common starting point and a common goal to all participants, but the analysis of the answers or the participants is not relevant for the evaluation of *KD*.

The data was collected and fed to *KD*. The resulting knowledge graph was vastly complex, with 505 nodes and 2,164 unique relationships. A preliminary analysis showed that users had 21 unique intentions (total of 36) and 26 unique insights (total of 27). On the machine side, there were 143 unique events (total of 252). I also looked through all insights related to WMTQ1 (the first task of WMT) and found that the 9 participants reached insights with an average of 20.9 interactions (machine-side events), and on average, each participant within WMTQ1 reached 1.44 insights. Similarly, across all WMT tasks, users had an average of 1.13 intentions and 1.29 insights per task. Overall, users had more insights than intentions. Also, on average, each intention took 6.11 interactions (machine-side events) to gain insight.

The examples seen so far (Figures 6.4, 6.5, and 6.6) are all direct examples of the analysis and slide decks extracted from such data. Based on the collected data, I explored *KD* with an expert in the WMT tool. He works with Engage Nova Scotia and, for this example, was tasked to create a report to share relevant information he might find related to the state of education in Nova Scotia. He opens *KD* and looks through the intention list, selecting the "identify the area with most concern about accessing education" to be explored. Then, he sees the *KD* interface of Figure 6.4. He finds that three users had similar insights who said that the "B0C" postal region answered the insight. The node-link visualization also shows that four of the insights are connected to the intention through 3 hops. Checking the fifth insight (slide number 5), he noticed that someone said that "a quarter of the province would take more courses, but they are expensive.". He also noticed that the insight of slide number 5 was done by the same user of insight of slide 4, or in other words, the user first had an insight more closely related to the intention and then another insight right after that, which was less directly related, though still relevant. He generates a slide deck of these insights and annotates in the

PowerPoint file downloaded that slides 4 and 5 are from the same user.

He then changes the *KD* query to show the interactions from one of the users. He sees Figure 6.6 where he notices screenshots of slides 2 through 7 are very similar. Checking the inspector on each, he confirms that there was little visual change between each step, but extra information in the inspector indicates that the user was clicking on certain options of WMT that are not relevant to the report he is making. He removes these slides from the slide deck generation through the slide panel at the bottom of Figure 6.6 and generates a slide deck of 6 slides, which includes one slide with the intention, four describing interactions, and one final slide with the insight. He unites the two slide decks in PowerPoint, adds description and conclusion slides, and sends it to his boss.

I also interviewed another WMT expert, who aided us during *KD*'s preliminary tests. She was positively interested in *KD*'s speed in showing all insights related to a given intention and that he would use extensively the ability to quickly extract screenshots of past insights to populate their slide decks, similar to the expert's experience. Overall, she believes her current workflow is equivalent to *KD* when considering the time required to build slides, but *KD* scales much better. Additionally, she said that *KD* already has the text inputted by the user, which would save her time in writing explanations for the slides herself. Her final feedback was that it would be great to have a text interface, similar to chatGPT, to ask natural language questions to the data.

6.6 In-the-wild Evaluation with ModKT

To evaluate *KD*, I attached it to another *Vi&VA* tool that fits the pre-determined requirements (see Sec 6.4.1). ModKT [33, 184] (video) is a VA tool that ingests a set of documents, such as research articles, extracts key terms, and applies key-term-based clustering [201] to the corpus. ModKT uses the articles' metadata, such as abstract and title, for clustering. Users can visualize the documents through Word Clouds, dimensionality reduction scatter-plots, and bar charts, which compare key terms extracted from the documents to custom user-defined words. Users can customize the parameters for clustering and dimensionality reduction to discover sets of (dis)similar documents and visually analyze their (dis)similarities.

Similar to WMT, I once again asked users to use the tool to collect their *storyline*s and populate *KD*. I then interviewed experts in the ModKT tool to evaluate *KD* slide generation within the expert's self-reported use case. I also compare whether the ModKT expert's opinion and feedback match the WMT expert's. It is important to note that while the WMT expert aided during the development of the *KD* approach, the ModKT experts did not.

To attach *Knowledge-Decks (KD)* to ModKT, I once again had to define what data from the tool is to be collected. By using the process shown in Sec. 6.4.2, the data collected are as follows:

HUMAN Update: label (insight or intention), created time, URL, screen size, text, keywords, shapes drawn, user id, analysis id

HUMAN state-space: label (insight or intention), created time, last updated time, keywords time

MACHINE state-space: label of what the event is, created time, last updated time, clustering configuration, dimensionality reduction selection/parameters, selected/highlighted document, words inputted in the word similarity panel

MACHINE Update: event name, created time, URL, user id, analysis id, and all the same data from the related Machine state-space with the exception of user id and created/updated time

I collected data from 4 users, leading to a knowledge graph with 861 nodes and 3688 relationships. Out of the nodes, there were 14 unique intentions (total of 22), 20 unique insights (total of 23) and 363 unique interactions (total of 399). Further analysis can be done, but it was judged not relevant for the evaluation of *KD*. The *KD* interface was then populated with the resulting knowledge graph.

After applying *KD* to ModKT, I invited an expert to verify how applicable *KD* is to their daily work. The expert works with ModKT, is a Ph.D. student in Text Mining, and has experience applying various techniques for text mining, text clustering, and exploring text-based datasets. In this interview, I first asked about her own experience with slide decks, how much effort is needed when making slide decks about her tool, and what the normal workflow is in case she needs to share information from her tool with colleagues. Then, I gave *KD* for the expert

to use and gather her feedback when asking if and how *KD* would impact the questions asked earlier.

In the first batch of questions, she discussed that there are various meetings and presentations in her day-to-day that use slide decks, such as reading groups, progress reports, conference presentations, teaching, and thesis defenses. Additionally, even informal meetings with colleagues might require a simple slide deck to explain more complex concepts. If they need to convey a single insight from a tool, such as ModKT, she and her colleagues would take a screenshot and forward it to each other through social media with some text for explanation. In the case of conveying multiple insights, however, they would use a slide deck. Creating a slide deck would follow academic ideas, with slides to explain data/use-case, then a couple of the results with insights, and then a slide for the conclusion. According to her, this process would take an hour more or less, but if there are more than ten users generating insights, then it would take even more. She and her group would use video recordings to record and convey the interaction or behavior that led to a certain insight. She said that all such media would be useful in writing and publishing academic papers.

Next, she was shown *KD* loaded with the data collected from ModKT users. After showing the tool and letting her use it to explore the users' knowledge pathways and the slide generation, she said that *KD* would greatly help extracting and sharing user behavior. While she normally records a video to store and share user behavior, the process involves rewatching and editing the video. She would also scan the video to take screenshots of the most important moments of insights. However, she said that using *KD* to generate a slide deck would greatly simplify and automate this process since it already takes the screenshots for her. Also, she said that the process she usually uses for user studies can be significantly automated with *KD*. Normally, she would need to watch her user and write down the insights and think aloud results, but *KD* does this automatically for her, which would avoid much of the manual labor she faces in academic writing. She said she could also perform bigger user studies in less time, which might help her evaluation results.

Another point she raised was that if new users came to the tool and found a

problem with ModKT, she could open *KD* and see what the user did before getting to the problem. This would help to remove bugs and other problems from the tool before user evaluation. Regarding the node-link visualization, she was able to understand well and communicate the results during the interview, but she noted that lay users or users from outside academia may find it difficult to use the visualization for exploration, but the tutorials and the generated slide decks themselves can be used in such cases. She noted that such slide decks have the potential to be uniquely useful for psychology research to better understand user behavior in general. The analysis of how users crossed paths during their usage of ModKT is also of interest to her since it could show different configurations of her tool that answer the same questions. She finalized mentioning that *KD* seems to have a very wide usage potential, like testing the effectiveness of tools before being released to the market, focused marketing potential after better understanding the user and your own tool, and the potential applicability in much of the current user evaluation process in her field.

6.7 Discussions, Limitations, and Future Work

Although *Knowledge-Decks (KD)* has arguably reached its goal, I recognize that more advanced processes could be applied in regards to how to process the texts within the user's intention and insight. For instance, the *KD* keyword generation is done by lemma extraction but could also be performed by topic modeling [64], KeyBERT [94], or other machine learning models. Similarly, *KD*'s Knowledge Graph could be used in many more potential graph network analyses, including advanced procedures of graph completion and recommendations, which in turn could generate other styles of slide decks, such as slide decks focusing on the "summary" or "tutorial" or specific aspects of the tool. Indeed, the slide deck generation could also have been enhanced with NLP, such as through chatGPT or other similar techniques to enrich, correct, summarize, or explain the various intentions and insights. Though I judged it better to use simpler techniques in this work to allow for easier reproducibility and broader applicability of the *KD* approach, I intend to investigate the usage of such techniques in future applications.

The *KD* visualization approach can also be improved with the points raised

from the interviews conducted. For instance, although I used a node-link approach due to its wide use for graph network visualization, the evaluation of *KD* against other visual metaphors when applied to the same slide deck goals and using the same querying engine would be agreeable. Further research could check for other visualizations and interactivity strategies for users to explore and analyze the data type contained within a *KD* knowledge graph, including natural language conversations with the graph.

One of *KD*'s core complexities is the need to retool the existing *Vi&VA* tools with the connection to implementation. Although this decision was by design to automate data collection, I recognize that not all tools are easily adaptable to be attached to external libraries like ours. Future works may investigate other ways to collect user intention, insight, and behavior without any modification on the *Vi&VA* tool's part, such as through browser extensions or by recording and processing the video/audio feed of users while they use tools and automatically processing them into a usable Knowledge Graph.

Another key issue when handling user data is the privacy concern, especially on the users' part, regarding how user data is being utilized. Throughout the study, I maintained complete anonymity between users, only displaying an *ID* provided by the user, but privacy and accountability are concerns usually raised by users and researchers alike [250]. Although no participant raised this issue during the tests, there are attempts of companies and research institutions to restrict or limit the collection of user data of any kind. Further evaluation is required to say that *KD* can be used in commercial scenarios. Nevertheless, *KD* does not explicitly attempt to solve this issue other than only collecting and displaying anonymous information.

The *KD* methodology can also be applicable in different use cases and goals, such as providing recommendations retelling previous users' experiences within the *Vi&VA* tool itself. Indeed, the interviewees provided us with valuable feedback on other potential uses of this methodology other than slide decks, which I intend to investigate. Of course, these goals and limitations differ from *KD*'s, so I digress.

Finally, certain design choices may be of concern. For instance, using slide decks to narrate the user's knowledge discovery process has its issues [171], such

as being seen as useful only for presentations to novice users. However, the ubiquitousness of PowerPoint slides for presentation purposes, including infographics and storytelling, must be considered. PowerPoint files are also easily editable, allowing for the inclusion or exclusion of information, such as a summary slide or the removal of an unrelated insight slide, to be handled on a case-by-case basis. As StoryFacet [171] says, adding animations, filtering, and other features of PowerPoint or similar software would also potentially enhance the slide decks generated, but these modifications are already beyond my original goal. Also, *KD*'s requirement of only supporting web-based and stateful *Vi&VA* tools is a limitation I recognize. Yet, I believe many tools fit *KD*'s requirements, which makes *KD* a promising starting point.

6.8 Conclusion

I have presented *Knowledge-Decks (KD)*, a novel approach to collect user *storyline*s as knowledge graphs, query the knowledge graph to extract *storyline*s of interest, visualize them as a multi-layered layout approach, and generate slide decks out of a linear sequence extracted. *KD* collects user intentions, interactions, and insights during user knowledge discovery sessions when using *Vi&VA* applications and tools and automatically structures the data into a knowledge graph. By selecting a specific pre-defined query and insight or intention, the *KD* visualization displays a node-link metaphor of the *storyline*s that corresponds to the query, allowing users to investigate the insights, intentions, and interactions of their users. *KD* also allows users to generate PowerPoint slide decks out of said *storyline*s telling the story of the selected query, such as "which insights users obtained with their tool" and "how was it done".

KD was evaluated by being attached to two existing *Vi&VA* tools where users were asked to perform pre-defined tasks. By collecting user intentions, interactions, and insights, three types of stories were available to users. An expert from each tool was interviewed regarding their use of slide decks in their day-to-day, and then I asked about their thoughts on *KD* and how it compares to their current workflow. The experts provided valuable feedback and overall agreed that *KD* provides more efficiency due to the speed of generating slides and reliability since

the slides generated are from users instead of their own, making them more useful in presentations. Limitations and potential usages of *KD* were discussed, pointing to further research to investigate what other applications the *KD* approach can be of use. In short, the experts validated *KD*' usefulness in exploring and generating slide decks that can be used to share user stories with peers or to be used as a draft version of a presentation.

Chapter 7

Future Directions and Intelligent Agents (IAs) [1]

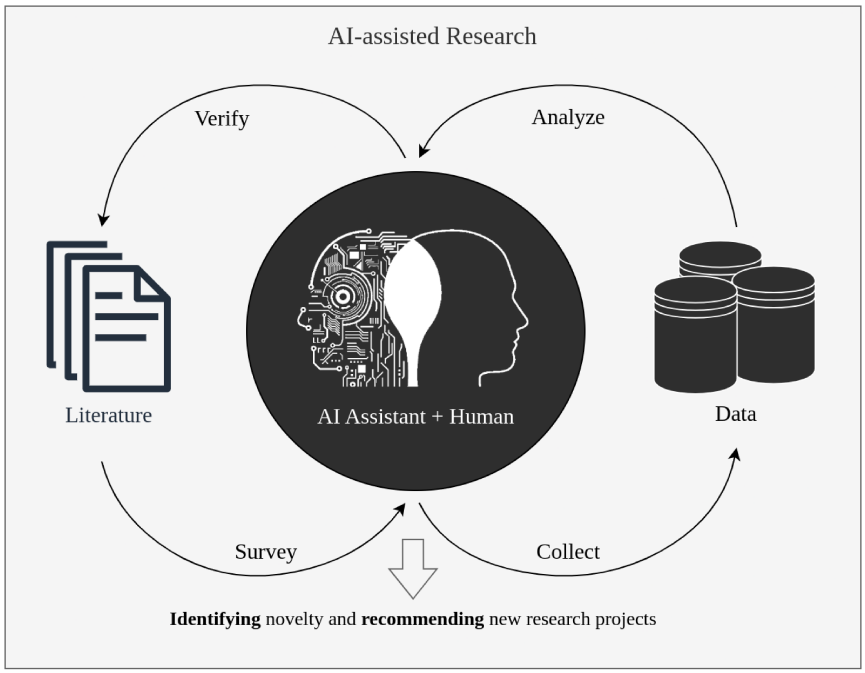


Figure 7.1: Intelligent Agents (*IA*) can cooperate with humans during a scientific process.

ONE aspect of the direction of this dissertation that has had large advances recently is Intelligent Agent (*IA*). I have investigated its usability in ChatKG (chapter 4) to collect related knowledge from an automatically detected pattern within a time series. Ongoing work, which is soon to be submitted, has also shown that the flexibility of an *IA* like ChatGPT allows for not just the collection of knowledge, but also the summarization of collected knowledge and the formatting of knowledge [146, 239]. Assuming a text dataset with the collective knowledge users have written about a certain topic, *IA* are already able to read, summarize,

[1] This chapter was partially based on *Oliveira Jr, O.N., Christino, L., Oliveira, M.C.F., & Paulovich, F. V. (2023). Artificial Intelligent Agents for Materials Sciences. Accepted in the Journal of Chemical Information and Modeling (JCIM).* Certain parts were redacted when included in this dissertation due to the specificity of its Chemistry contents, which are irrelevant to the discussion.

and suggest relevant outputs, such as insights or suggestions, out of the dataset. In order to understand how *IA* will impact the future of *Vi&VA* research, this chapter discusses the current advances of Intelligent Agent, as is exemplified by Figure 7.1. I then finalize discussing the research proposed throughout the dissertation as a limitation and future work in light of these recent advances.

The utilization of Artificial Intelligence (*AI*)-powered research agents has the potential to revolutionize the way scientific work is performed. The potential impacts of *AI*-based tools on scientific practices are both anticipated and feared by researchers, as revealed in a recent survey conducted by Nature [235]. The workflow of academic research publication is somewhat standardized, and users of Intelligent Agent (*IA*) such as ChatGPT [169] may justifiably wonder whether this workflow can be automated. In fact, *IA* can already search for knowledge sources, summarize them into bullet points, and write convincing arguments about the contents in the summary. Only a few years ago, this possibility was considered too far into the future to be worth discussing; today, we see an increasing number of writing materials, from blog posts to entire books, being partially or fully prepared by *IA*. Amazon has just limited the number of books an author can publish per day in an attempt to address the issue[2]. We take the view that *IA* may soon incorporate established strategies of the academic research workflow [109] into LLM-based (large-language models) tools, particularly to cater to specific aspects of a research area. I illustrate examples of the impact of *IA* as 5 possibilities I envision occurring in the near future. In describing the functionalities of the *IA*, I assume they will have unlimited access to data.

1- Discovery and design. This functionality has been addressed in many recent contributions from the literature, and therefore, I only present a brief description. Discovery and design of new research, such as new materials in material sciences, have been made possible with the large databases of materials properties, especially with the materials genome initiatives [121]. For instance, machine learning (ML) algorithms have been used to predict materials' properties based on existing

[2] `https://tinyurl.com/2vy6szu6`

data [100] and in quantum chemistry simulations [216]. In producing new materials, *AI* can be employed to optimize the synthesis process or recommend potential combinations of existing materials to create composites with enhanced properties [48]. Many are the areas for which *AI* systems have been developed, including those related to the aerospace and automotive industry, for energy storage and conversion, and in seeking sustainable materials (for a perspective, see Oliveira Jr et al. [166]).

2 - Consultation of the literature to verify specific information, similar to a scientific fact-checking functionality. The *IA* is expected to search and explore existing literature given a prompt. This could be triggered, e.g., when a researcher wants to verify if a new idea s/he just came up with has already appeared previously in the literature; or when a researcher wishes to fact-check some statement in a paper which s/he believes might be mistaken. This task may seem straightforward at first glance because it only requires the *IA* to search for information already available in the scientific literature, but several practical difficulties may arise [13]. First, there may be controversies about the prompt and how the answers are formulated in the literature, requiring heavy involvement from the researcher to consider the alternatives and judge properly on a case-by-case basis. Of course, such a task could be relegated to another *IA*, a concept I call *autonomous agent swarm*, where multiple *IA*s specialized in different aspects of the problem work in cooperation to respond to a prompt. The issue of controversies can also bring to light differences of opinion between a researcher and the *IA*. The concept I call *autonomous agent ensemble* may be applied with multiple specialized *AI* agents voting for a solution to the problem. This voting process, analogous to the ensemble machine learning algorithms, can be tweaked by the users *a priori* or *a posteriori* to best match their view on existing controversies. This personalization of an *IA* to mimic the will of a single person is referred to as *personal AI* [38].

When searching for an answer, the *IA* may need to extract information not only from text but also from non-textual material such as figures, graphics, images, and videos, among other sources. This limitation of interconnecting LLMs to other models is still unresolved, but possible solutions have been discussed within the

concept of a *general AI model* [259] or *multimodal AI* [1, 19]. Another limitation of current *IA*s is their single-shot accuracy [19]. In our view, if a researcher demands a fact-checking procedure, the *IA* must provide accurate answers even when the matter is unresolved. This requires measuring how well the literature agrees with a given reply, to which extent there are disagreements, or whether there is a lack of consensus altogether. Since current models require multiple, and usually conversationally-aware, prompts to answer a single prompt correctly, extracting such an answer from the summarized collective knowledge from the literature poses a severe additional challenge. In fact, the concept of *knowledge* itself is still under debate. For instance, in our materials science scenario, would knowledge be the collection of papers published in the scientific literature, the summarized literature available from the *IA*, or the knowledge that we, as users of the *IA*, reach after using it? Despite the difficulties in establishing definitions for such fuzzy concepts, efforts are underway to develop fact-checking systems and tools that rely on machine learning, natural language processing, and crowd-sourced verification [138]. These tools can assist in identifying false information and highlighting discrepancies. Achieving a fully comprehensive and accurate fact-checking system remains a complex endeavor, though.

3 - Data analysis. Automated data analysis is being tackled by the major *AI* developers. Microsoft Co-pilot, ChaptGPT Plus, and Google Assistant have been capable of ingesting data files, such as CSV tables, and extracting answers from the data given a prompt. Data analysis products like Tableau and PowerBI have added *AI* capabilities to their visualization toolkit to generate infographics. These examples show how fast *AI* has taken over data analysis, though certainly many limitations remain, and also possibilities open to further exploration. For instance, beyond support to analysis execution, authors in a recent study [97] consider how *IA*s could contribute to analysis planning. Considering the complex nature of a data analysis activity, which involves multiple interconnected stages of coding, reading, and reflecting, they conducted an empirical study investigating to what extent and in which ways a hypothetical *IA*s could be useful to help analysts in planning a data analysis.

Data analysis as a functionality of our proposed *IA* requires multiple strategies, which depend on the researcher's prompt, the results from information verification as described in step [1], the available data, and the data type. Machine learning is now widely employed to discover patterns in large amounts of data. In many such applications, it does not matter how the data has been acquired or the specific nature of the data because no detailed interpretation is required from a machine learning algorithm. Such applications also do not investigate the *information boundaries* of the data, something we must consider to verify the statistical relevance of the data and the findings. In contrast, the data's nature matters to our *IA*, and it must be "knowledgeable" about the type of method employed to acquire or generate the data and its information boundaries. For example, the *IA* should be able to identify which type of optical spectroscopy or microscopy was used to obtain the data and how to interpret it/ e.g., the *IA* should identify the bands and seek their correct assignment to chemical groups in vibrational spectroscopy. This latter type of information must be mined from the literature and properly parsed to match the prompt given to the *IA*. Any result should also include its confidence or statistical relevancy.

A similar reasoning applies to a system capable of extracting visual explanations. Visualizations and infographics are considered more effective than text to convey explanations for a given data set [7]. When explaining patterns observed in time series, for example, rather than providing a textual description, it is better to show the associations between patterns and time as a visualization, allowing users to interact to confirm textual explanations, as exemplified in the work by Christino and Paulovich [52]. In fact, the essential problem of automating data analysis has been shown to be solvable in the near future [104]; however, I aim beyond automation. In our proposed *IA*, I envision *Visual Analytic Democratization*, where all processes and result acquisition would be automated to reduce the barrier to data and visual analysis faced by non-specialized researchers and students. For instance, I argue that less experienced practitioners in material and computer sciences would not be required to know *a priori* how to apply a peak outlier detection algorithm (and its possible limitations) to a multimodal database. Instead, they could expect the *IA* to apply valid, possibly alternative, approaches and

explain and argue the results back.

4 - Survey of the literature. We have so far discussed the *IA*'s information-gathering capability. LLM models have writing capabilities at a near-human level [109]. Indeed, articles purely *AI*-written have already been accepted in peer-reviewed journals, which at the very least shows that *AI*-written articles can pose as human-written ones, as exemplified in the work by Cotton et al. [57]. Tools like Avidnote[3] and Kahubi[4], which aid researchers in writing questionnaires and articles, are a first step towards *AI*-driven automated writing. We propose that our *IA* should ingest the available literature and use its writing ability to write summaries, reviews, and surveys. Surveys on a given topic in chemistry, materials, or computer sciences can be generated automatically with a few prompts. While the currently available *LLM* tools are likely to generate surveys with a superficial analysis, particularly because they still do not have access to the whole of the scientific literature, these limitations will probably be eliminated soon. For such surveys, *LLM* tools could be integrated with recent strategies that combine natural language processing with analysis of networks to obtain semi-automated literature surveys [9, 203, 29]. Moreover, enhanced with the data analysis capability, it could also provide summaries of the citation patterns and descriptive analyses of the communities identified in the paper citation networks, informing the most important keywords, topics, and critical papers characterizing the different communities in the field.

An essential feature of this functionality would be the ability to summarize text to distinct audiences and gradually refine the descriptions at different levels of abstraction, which is more challenging than just identifying very broad or high-level topics. Additional desired features would be the ability to write tables, mathematical equations [229], and produce images [227, 25], all of which are important elements in article or survey writing. Still, this seems to pose no significant difficulties given the current state of tools already available with the LLMs. Such features render the *IA* capable of extracting and grouping the research topics within the literature and displaying them as taxonomy tables or visualizations.

[3] `https://avidnote.com/`
[4] `https://kahubi.com/`

5 - Identifying novelty in research projects and recommending new projects. Researchers must plan their careers in the short term by choosing relevant scientific-technological problems to address. Abilities like support to ideation and brainstorming can aid in this planning and are some of the proven uses of LLMs [114]. Indeed, the back-and-forth process of ideation and decision-making required by researchers to develop novel ideas and prove them valid is an expected feature of a research-focused *IA*. Yet, although tools like ChatGPT and others can use existing information for ideation and brainstorming, as of now, the topics available for consideration are limited to those covered in its training corpus. We can only detect the *absence* of what would potentially exist through data mining, mathematical techniques, and simulations [244]. For instance, similar to how *AI* was already used to simulate human interaction [172], it can use physical laws to simulate and discover new materials that could possibly exist [35]. The capability of discovering what was not discovered before (a.k.a. novelty) will be an essential requirement in preparing research proposals for which a high degree of novelty is demanded. This functionality has similarities to the previous one of conducting literature surveys, but it should include a specific feature to identify potential research directions that differ from existing ones. While there is software to detect originality in papers, mostly based on text similarity, equipping an *IA* with this functionality requires a step further in assessing research trends, assessing the feasibility of a research plan (which depends on the infrastructure available to the researcher served by the *IA* and his/her knowledge and expertise) and prospecting the potential impact.

Means of achieving this goal are in sight. *LLM* tools such as ChatGPT were originally assumed to be an all-encompassing *AI* chat-bot, but we are now aware of issues such as *hallucinations*, which are due to their inability to recall precise information [4]. Projects like AutoGPT [253], BabyAGI [230], AgentGPT [228], MemGPT [170], and ChatGPT plugins [169] are all attempts to attach new capabilities to LLM tools. For example, coupling ChatGPT with the Wolfram [229] environment allows for highly complex math calculations. This enables it to answer math-related questions by forwarding them to Wolfram rather than *hallucinating* an answer (see [5]). Similarly, ArXiv and Google search can be attached to AutoGPT,

[5] `https://writings.stephenwolfram.com/2023/03/chatgpt-gets-its-wolfram-superpowers/`

allowing it to fetch information beyond its training dataset to answer prompts. ChatGPT Plus can be attached to a Python runtime, allowing it to perform any coding operation, such as reading uploaded files, extracting statistical information from CSV tables, and generating images and PowerPoint presentations with existing libraries.

This *plug-in ecosystem* concept expands the potential of LLMs [169]. With a proper physics-simulation plugin, the *IA* I propose would be able to collect existing data about how to generate new *knowledge*, such as by synthesizing new materials from step [1]. The *IA* would additionally reason about the existing *knowledge* with step [2], summarise current findings with step [3], and use such summary within the physics-simulation plugin to verify if any new non-existing material can be synthesized. This process will likely be time-consuming, but its fully automated nature and its potential for parallelization would provide researchers with novel theories or hypotheses that are ready to be tested, thus accelerating the standard research cycle.

The *IA* of the future The four functionalities I devised for the *IA* were all based on existing technology. That is to say, even in cases where severe limitations still exist, they can, in principle, be solved with proven strategies in data science and *AI*. Therefore, I am confident that these functionalities can be implemented in the next few years, particularly based on current projects such as LangChain [226], Make.com [225], and other tools [253, 230, 228]. We now discuss a more long-term perspective related to the prediction of a new paradigm of knowledge generated autonomously by a machine[173]. Within this paradigm, we would not have a mere *IA* but an *AI* researcher. The idea of an *AI* researcher is much more ambitious, for the tool would have to investigate the literature, brainstorm ideas, propose and implement a research project, and then communicate the findings in a scientific paper.

For the sake of simplicity, as it will probably be done in developing an *AI* researcher, I address the different tasks in the research workflow individually. Choosing a scientific problem and producing a paper is perhaps the least difficult to implement. With the features associated with the analysis of the literature

and novelty identification, the *AI* researcher will probably choose a suitable topic for the project. As for the writing up, writing full essays and scientific papers has been proven feasible with LLMs [109], despite the many limitations [249]. In contrast, developing the research project (even in projects on theory or simulations, without experiments) and generating scientific contributions will be much more challenging. It may take decades before this can become a reality.

As a final comment, the use of *IA*s may be accompanied by considerable changes in scientific publishing. Scientific journals will likely adopt machine-readable formats for their content. Machine-readable formats would allow for more efficient and effective indexing, searching, and analysis of scientific articles, as well as enable the development of more advanced text mining and NLP tools. The adoption of machine-readable formats may require publishing companies to develop new business models. This is because machine-readable formats could enable new forms of content delivery and distribution, such as through the use of application programming interfaces (APIs), which could provide opportunities for third-party developers and other organizations to build value-added services on top of scientific content. Traditional subscription-based revenue models may be impacted, as access to scientific content could be provided in different ways, such as through pay-per-view or micropayment models. In summary, in our ongoing digital era, just as we have seen how machine learning techniques have revolutionized scientific practices in many fields, such as in protein folding through alphafold [37] and in image processing through convolutional neural networks [206], large language models and intelligent agents have the potential to strongly impact not only the field of material and computer sciences, but disciplines in natural sciences as a whole.

Chapter 8

Conclusion

In this dissertation, I have discussed extensively how *Vi&VA* can grow closer to *Visual Analytic Democratization*. In order to discuss how well this dissertation achieved the original goal, first, let's compare it to the work discussed. Recalling my research question:

> *How to use Provenance to automate the Visualization & Visual Analytics' knowledge gathering process in order to model, structure, store, query, and share the users' storylines? And does this bring advantages back to the users and, consequently, to the idea of Visual Analytic Democratization?*

Let me discuss the different components of this question and how my dissertation addressed them:

Automation: *Q4EDA* and ChatKG described in chapter 3 and chapter 4 discussed the experience of a user within a *Vi&VA* tool in order to answer a question. In *Q4EDA*, users can select parts of a visualization to retrieve information about it from Search Engines, and in ChatKG, I show how a *Vi&VA* tool can automate this questioning process through Intelligent Agents.

Model: With the *VAKG* framework described in chapter 5, I showed how to use existing modeling literature in practice, transforming them into a methodology to define the relationship between Machine and Human within *Vi&VA*, what type of responsibilities and operations each perform, and how the resulting taxonomy can lead to building up an ontology. I also defined what a user *storyline* and how this modeling process serves to concretely define the user's natural knowledge-gathering process.

Structure: Both ChatKG and *VAKG* discussed and described Knowledge Graphs (*KG*s), a structure designed to store knowledge. With ChatKG, I showed that

a *KG* can be used to structure the *explicit knowledge* to associate information from an *IA* and findings from a temporal dataset. In *VAKG* I discuss how this same structure can be used to save a user's *storyline* for posterior analysis and downstream tasks.

Store: While describing *VAKG*, I discuss how one can collect user data through *provenance*, structure as a *KG* and store it into a database. By focusing on a specific use-case, *KD* records storylines of multiple users using multiple *Vi&VA* tools into a *KG* database in order to be used. Throughout this dissertation, I defined, contextualized, and exemplified *knowledge provenance* and *behavior provenance* as a way to better understand how past users utilized a *Vi&VA* tool, and how they can be used to store the user's experiences.

Query: In most of the dissertation, I have shown how querying for knowledge is an integral part of *Visual Analytic Democratization*. *Q4EDA* shows the advantage of allowing users to query for information in a visual manner. ChatKG automates this query though *IA*. *VAKG* and *KD* populate a database that can be queried to extract interesting statistics, check how the tool is being used, and better understand users.

Share: With *KD* (chapter 6) I demonstrate that by using the *VAKG* framework on a stateful system, we are able to collect screenshots and textual annotations from users in such a way that we are able to reconstruct the user's *storyline*s. By having an output as a slide deck, users of *KD* can easily collect and share the user's insights and the means they used to reach the insights.

As a whole, I showed that by allowing users to more easily search for knowledge, be it external knowledge from other databases, from *IA* or from a repository of knowledge populated from past user experiences, *Vi&VA* is able to better able to cater users, especially users with little expertise in *Vi&VA*.

8.1 Limitations and Future Research Directions

In this dissertation, I proposed methods and techniques that better allow users to utilize existing knowledge for their own gain. However, when considering the

research theme "How can *VA* be simplified or automated so that it can be used and further democratized, that is, developed and used more broadly?" proposed in chapter 1, I do not claim this dissertation has exhausted all potential research of the proposed research theme. Due to recent advances in *IA* discussed in chapter 7, a potential future where visualization tools are not needed is also possible. Extensive discussions of the usefulness of visual interactivity were had during my research among important researchers in the field. Therefore, it is important to note that we are living in a time where much change is happening. This dissertation showed the advantages of visual selections in *Q4EDA* and slide decks in *KD* to collect and share knowledge, but due to the ubiquitousness of natural language, *IA* has the potential to disrupt the field in a large way. I plan to further integrate *IA* as part of Visual Analytics Democratization in future research.

Additionally, the way advances in *IA*, such as ChatGPT, arrived was very impactful in this dissertation. Although I claim it is and will impact *Visual Analytic Democratization* more and more, the amount of skepticism and worry had by some within the *Vi&VA* field and among the general population during the last year limited the results shown in this dissertation. For instance, I argued that *Q4EDA* can be considered to be better than directly searching Wikipedia, but would ChatGPT be even better? And even though *KD* records screenshots and texts to represent the collective user knowledge, would *IA* be a better alternative to summarize the texts and generate slide decks, especially when using multi-agents for fact-checking (see chapter 7)? Even though the results of this dissertation are relevant, I argue that due to the recent advances in *IA*, this dissertation brings more questions and potential future research than actual solutions. That said, I also argue that this dissertation contributes by demonstrating that *IA* and visualization techniques can be used in tandem, as is the case with *KD*, and proposing further methods of interoperability in chapter 7. I do challenge future researchers to contrast the solutions raised in this dissertation to a near future where further *IA* advances will be available since, with it, we would together reach *Visual Analytic Democratization* faster.

This dissertation defined and discussed many topics ranging from *storyline* to *democratization*. However, many of these topics are ambiguous in the literature.

During the development of certain parts of this dissertation, there were substantial challenges in properly communicating the definitions of these topics and how they are put together. For instance, the publication of *Q4EDA* took years to be accepted because of the dual nature of its proposal: a technique to convert visual queries and a system to collect visual queries and display Wikipedia results related to it. Areas outside *Vi&VA* usually evaluate a technique by comparing it to certain scores. Machine Learning research, for instance, can use the model's accuracy as such score. However, a technique to transform a visual query into a search query, such as *Q4EDA*, has no score to be compared to. Therefore, I used the opinion of the user who made the selection as the way to evaluate the technique. Future research in using visual query, especially in the context of *IA*, is planned to not just utilize state-of-the-art methods, but also to compare to *Q4EDA*.

The proposals in this dissertation also have shown to be very hard to evaluate. *Q4EDA* itself allows users to select visual elements, which is quicker than the alternative of exiting the tool to search for the result in Wikipedia as proposed in chapter 3. Yet, how to evaluate if it is "better"? The evaluation shown in chapter 3 attempts to answer this question, but since there is no other technique that also converts visual queries into search queries, the evaluation of the "technique" became much like what would be expected out of the evaluation of a "system" instead. Due to this issue, the system called *LINKED* was removed from the publication of *Q4EDA* in order to better position the work as a technique. The same issue can be found in ChatKG (chapter 4) and KD (chapter 6). With the theoretical contribution of *Vi&VA*, another significant challenge arose. As was discussed in chapter 5, the main contribution is in a methodology to apply an existing theoretical model directly in practice through the use of Set-Theory and *KG*s. However, I had to use many examples, including creating a sample implementation, in order to demonstrate its use. This caused *Vi&VA* to also be confused as a "system" by some.

Another limiting factor was the amount of terminology used throughout the dissertation, which impacted reviewers to pinpoint which aspect of each proposal was novel. For instance, in *VAKG*, we propose an ontology. Although it is possible to submit such work in a Semantic Web venue due to the ontology proposed, the ontology itself is not novel, but the novelty is instead in the fact that it is based

on a visual analytic model. Yet, at the same time, researchers in Visual Analytics are not accustomed to the terminology borrowed from the Semantic Web. Similar issues occurred when discussing *Knowledge provenance* and *Behavior provenance*, which are terms from the Data Management field. This challenge was tackled through repeated reviews by polishing the submissions. This made me come to the conclusion that works that are able to only use well-known terminology in the fields of their submission venues tend to be more successful. In the case of this dissertation, however, this was not viable without significant changes in its goals and results.

8.2 Summary

In this dissertation, I have discussed extensively how *Vi&VA* can grow closer to *Visual Analytic Democratization*. For this, I surveyed the literature for the current state-of-the-art knowledge modeling approaches (O_1) and proposed two approaches (*Q4EDA* and ChatKG) and a system (LINKED) to aid in users in their seek of knowledge, making *Vi&VA* more approachable to users (O_2). I used this foothold to discuss and propose *KG* as the preferred knowledge repository method (O_3), leading me to propose *VAKG* to aid in the automatic structuring user *storyline*s by collecting user interactions, intentions, and insights through *provenance* methods. This model is then exemplified by being applied to various *Vi&VA* tools, where I showcase in *KD* that user *storyline*s can be (semi)automatically collected and stored. I discuss how sharing the collective user knowledge can be done by the generation of slide decks out of user *storyline*s and finalize discussing how *IA* has impacted the research so far and will be even more central to Democratizing Visual Analytics. Each step of the dissertation was demonstrated through examples, use cases, and interviews with domain experts who are not experienced *Vi&VA* users. The complete map of all contents of this dissertation is shown in Figure 8.1. With this, we provide a glimpse of how *Vi&VA* tools can provide more value for users in their unquenchable search for knowledge.

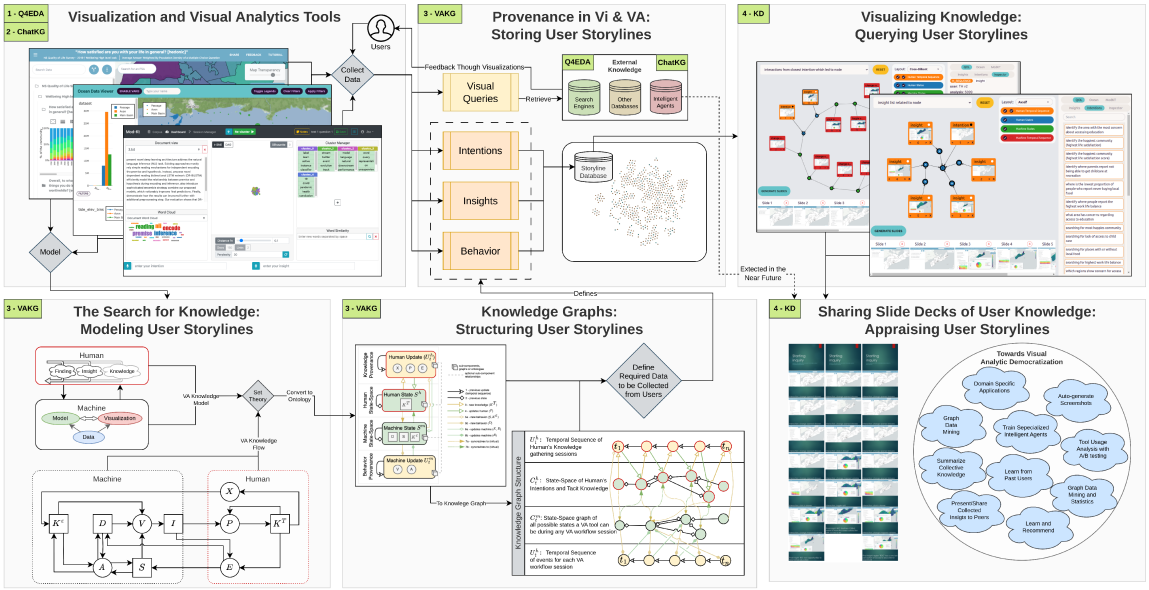


Figure 8.1: Summary of Results. First, I investigated how visual exploration and visual queries influence user *storyline* with *Q4EDA*. Then, I automate parts of the user's *storyline* with *ChatKG* though Intelligent Agents. I generalize the modeling process of user *storyline*s with *VAKG* and use *KG*s to structure user *storyline*s. I apply the *VAKG* process in *KD* by collecting user intentions, behavior, and insights. By populating a *KG* with user data, *KD* displays an interface for users to explore the intertwined *storyline*s of their tool and export it as slide decks.

Bibliography

[1] Julián N Acosta, Guido J Falcone, Pranav Rajpurkar, and Eric J Topol. Multimodal Biomedical AI. *Nature Medicine*, 28:1773–1784, 2022. doi: https://doi.org/10.1038/s41591-022-01981-2.

[2] Gulsum Alicioglu and Bo Sun. A survey of visual analytics for explainable artificial intelligence methods. *Computers & Graphics*, 102:502–520, 2022. doi: https://doi.org/10.1016/j.cag.2021.09.002.

[3] Mohammad Aljanabi, Mohanad Ghazi, Ahmed Hussein Ali, Saad Abas Abed, et al. ChatGpt: Open possibilities. *Iraqi Journal For Computer Science and Mathematics*, 4:62–64, 2023. doi: https://doi.org/10.52866/20ijcsm.2023.01.01.0018.

[4] Hussam Alkaissi and Samy I McFarlane. Artificial Hallucinations in ChatGPT: Implications in Scientific Writing. *Cureus*, 15, 2023. doi: https://doi.org/10.7759/cureus.35179.

[5] Ahmad Alshami, Moustafa Elsayed, Eslam Ali, Abdelrahman EE Eltoukhy, and Tarek Zayed. Harnessing the power of chatgpt for automating systematic review process: Methodology, case study, limitations, and future directions. *Systems*, 11(7):351, 2023. doi: https://doi.org/10.3390/systems11070351.

[6] Natalia Andrienko, Tim Lammarsch, Gennady Andrienko, Georg Fuchs, Daniel Keim, Silvia Miksch, and Andrea Rind. Viewing visual analytics as model building. In *Computer graphics forum*, volume 37 - 6, pages 275–299. Wiley Online Library, 2018. doi: https://doi.org/10.1111/cgf.13324.

[7] Francis J Anscombe. Graphs in Statistical Analysis. *The american statistician*, 27:17–21, 1973. doi: https://doi.org/10.2307/2682899.

[8] Grigoris Antoniou and Frank van Harmelen. *Web Ontology Language: OWL*, pages 67–92. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-24750-0. doi: https://doi.org/10.1007/978-3-540-24750-0_4.

[9] Henrique F Arruda, Luciano F Costa, and Diego R Amancio. Topic Segmentation via Community Detection in Complex Networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26, 2016. doi: https://doi.org/10.1063/1.4954215.

[10] Ashima Arya, Vikas Kuchhal, and Karan Gulati. Survey on data deduplication techniques for securing data in cloud computing environment. *Smart and Sustainable Intelligent Systems*, pages 443–459, 2021. doi: https://doi.org/10.1002/9781119752134.ch31.

[11] H Asghari, N Birner, A Burchardt, D Dicks, J Faßbender, N Feldhus, F Hewett, V Hofmann, Matthias C Kettemann, W Schulz, et al. What to explain when explaining is difficult? an interdisciplinary primer on xai and meaningful information in automated decision-making. *HIIG Impact Publication Series*, 2022. doi: https://doi.org/10.5281/zenodo.6375784.

[12] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007. doi: https://doi.org/10.1007/978-3-540-76298-0_52.

[13] Ömer Aydın and Enis Karaarslan. OpenAI ChatGPT Generated Literature Review: Digital Twin in Healthcare. *Available at SSRN 4308687*, 2022. doi: https://doi.org/10.2139/ssrn.4308687.

[14] Hiteshwar Kumar Azad and Akshay Deepak. Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56 / 6:1698–1735, 2019. doi: https://doi.org/10.1016/j.ipm.2019.05.009.

[15] Sriram Karthik Badam, Zhicheng Liu, and Niklas Elmqvist. Elastic documents: Coupling text and tables through contextual visualizations for enhanced document reading. *IEEE transactions on visualization and computer graphics*, 25(1):661–671, 2018. doi: https://doi.org/10.1109/tvcg.2018.2865119.

[16] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 31:606–660, 2017. doi: https://doi.org/10.1007/s10618-016-0483-9.

[17] Adithya Balaji and Alexander Allen. Benchmarking automatic machine learning frameworks. *arXiv preprint arXiv:1808.06492*, 2018. doi: https://doi.org/10.48550/arXiv.1808.06492.

[18] Oana Balalau, Helena Galhardas, Ioana Manolescu, Tayeb Merabti, Jingmao You, Youssr Youssef, et al. Graph integration of structured, semistructured and unstructured data for data journalism. *arXiv preprint arXiv:2007.12488*, 2020.

[19] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. *arXiv preprint arXiv:2302.04023*, 2023. doi: https://doi.org/10.48550/arXiv.2302.04023.

[20] Leilani Battle and Jeffrey Heer. Characterizing exploratory visual analysis: A literature review and evaluation of analytic provenance in tableau. In *Computer graphics forum*, volume 38 - 3, pages 145–159. Wiley Online Library, 2019. doi: https://doi.org/10.1111/cgf.13678.

[21] Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Dieter Fellner, and Michael Sedlmair. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE transactions on visualization and computer graphics*, 24(1):298–308, 2017.

[22] Jürgen Bernard and Mennatallah El-Assady. The Future of Interactive Data Analysis and Visualization. In Ingrid Hotz and H.-J. Schulz, editors, *EuroVis 2023 - Panel*. The Eurographics Association, 2023. ISBN 978-3-03868-227-1. doi: https://doi.org/10.2312/evt.20231119.

[23] Jagdev Bhogal, Andrew MacFarlane, and Peter Smith. A review of ontology based query expansion. *Information processing & management*, 43(4):866–886, 2007.

[24] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[25] Ali Borji. Generated Faces in the Wild: Quantitative Comparison of Stable Diffusion, Midjourney and Dall-e 2. *arXiv preprint arXiv:2210.00586*, 2022. doi: http://dx.doi.org/10.48550/arXiv.2210.00586.

[26] David Borland, Wenyuan Wang, Jonathan Zhang, Joshua Shrestha, and David Gotz. Selection bias tracking and detailed subset comparison for high-dimensional data. *IEEE transactions on visualization and computer graphics*, 26: 429–439, 2019. doi: https://doi.org/10.1109/tvcg.2019.2934209.

[27] David Borland, Jonathan Zhang, Smiti Kaul, and David Gotz. Selection-bias-corrected visualization via dynamic reweighting. *IEEE Transactions on Visualization and Computer Graphics*, 2020. doi: https://doi.org/10.1109/tvcg.2020.3030455.

[28] Matthew Brehmer and Tamara Munzner. A multi-level typology of abstract visualization tasks. *IEEE transactions on visualization and computer graphics*, 19 (12):2376–2385, 2013. doi: https://doi.org/10.1109/tvcg.2013.124.

[29] Ana Carolina M Brito, Maria Cristina F Oliveira, Osvaldo N Jr Oliveira, Filipi N Silva, and Diego R Amancio. Network Analysis and Natural Language Processing to Obtain a Landscape of the Scientific Literature on Materials Application. *ACS Applied Materials & Interfaces*, 15:27437–27446, 2023. doi: https://doi.org/10.1021/acsami.3c01632.

[30] Peter J Brockwell and Richard A Davis. *Introduction to time series and forecasting*. springer, 2016. doi: https://doi.org/10.1007/0-387-21657-x_1.

[31] Chris Bryan, Kwan-Liu Ma, and Jonathan Woodring. Temporal summary images: An approach to narrative visualization via interactive annotation generation and placement. *IEEE transactions on visualization and computer graphics*, 23(1):511–520, 2017. doi: https://doi.org/10.1109/tvcg.2016.2598876.

[32] IntroToIS BYU. Tableau practice problems, 2016. URL `https://www.youtube.com/embed/B3jKKQrhTko?start=0&end=255`.

[33] Eric M Cabral, Evangelos E Milios, and Rosane Minghim. Visual analysis of interactive document clustering streams. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pages 1–3, 2020. doi: https://doi.org/10.1145/3399715.3399962.

[34] Michael J Cafarella and Oren Etzioni. A search engine for natural language applications. In *Proceedings of the 14th international conference on World Wide Web*, pages 442–452, 2005. doi: https://doi.org/10.1145/1060745.1060811.

[35] Jiazhen Cai, Xuan Chu, Kun Xu, Hongbo Li, and Jing Wei. Machine Learning-Driven New Material Discovery. *Nanoscale Advances*, 2:3115–3130, 2020. doi: https://doi.org/10.1039/d0na00388c.

[36] Steven P Callahan, Juliana Freire, Emanuele Santos, Carlos E Scheidegger, Cláudio T Silva, and Huy T Vo. Vistrails: visualization meets data management. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 745–747, 2006. doi: https://doi.org/10.1145/1142473.1142574.

[37] Ewen Callaway. What's Next for the AI Protein-Folding Revolution. *Nature*, 604:234–238, 2022. doi: https://doi.org/10.1038/d41586-022-00997-5.

[38] Nil Goksel Canbek and Mehmet Emin Mutlu. On the Track of Artificial Intelligence: Learning with Intelligent Personal Assistants. *Journal of Human Sciences*, 13:592–601, 2016. doi: https://doi.org/10.14687/ijhs.v13i1.3549.

[39] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, 44(1):1–50, 2012. doi: https://doi.org/10.1145/2071389.2071390.

[40] Dylan Cashman, Shenyu Xu, Subhajit Das, Florian Heimerl, Cong Liu, Shah Rukh Humayoun, Michael Gleicher, Alex Endert, and Remco Chang. Cava: A visual analytics system for exploratory columnar data augmentation using knowledge graphs. *IEEE Transactions on Visualization and Computer Graphics*, 2020. doi: https://doi.org/10.1109/tvcg.2020.3030443.

[41] Remco Chang, Caroline Ziemkiewicz, Tera Marie Green, and William Ribarsky. Defining insight for visual analytics. *IEEE Computer Graphics and Applications*, 29(2):14–17, 2009. doi: https://doi.org/10.1109/mcg.2009.22.

[42] Shuo Chang, Peng Dai, Lichan Hong, Cheng Sheng, Tianjiao Zhang, and Ed H Chi. Appgrouper: Knowledge-graph-based interactive clustering tool for mobile app search results. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 348–358, 2016. doi: https://doi.org/10.1145/2856767.2856783.

[43] Min Chen and David S. Ebert. An ontological framework for supporting the design and evaluation of visual analytics systems. *Computer Graphics Forum*, 38(3):131–144, 2019. doi: https://doi.org/10.1111/cgf.13677. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13677`.

[44] Min Chen, Georges Grinstein, Chris R Johnson, Jessie Kennedy, and Melanie Tory. Pathways for theoretical advances in visualization. *IEEE computer graphics and applications*, 37(4):103–112, 2017. doi: https://doi.org/10.1109/mcg.2017.3271463.

[45] Xiaojun Chen, Shengbin Jia, and Yang Xiang. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141:112948, 2020. doi: https://doi.org/10.1016/j.eswa.2019.112948.

[46] Zhutian Chen, Yun Wang, Qianwen Wang, Yong Wang, and Huamin 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. doi: https://doi.org/10.1109/tvcg.2019.2934810.

[47] Zhutian Chen, Wai Tong, Qianwen Wang, Benjamin Bach, and Huamin Qu. Augmenting static visualizations with paparvis designer. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020. doi: https://doi.org/10.1145/3313831.3376436.

[48] Wonbong Choi, Rigoberto C Advincula, H Felix Wu, and Yijie Jiang. Artificial Intelligence and Machine Learning in the Design and Additive Manufacturing of Responsive Composites. *MRS Communications*, pages 1–11, 2023. doi: https://doi.org/10.1557/s43579-023-00473-9.

[49] L. Christino. Example of the provenance flow on kd, 2022. URL `https://www.youtube.com/watch?v=qMeVf_K16VQ`.

[50] Leonardo Christino. Wellbeing mapping tool, 2020. URL `https://datatool.nsqualityoflife.ca/`.

[51] Leonardo Christino. Christinoleo/kds: 0.0.2, November 2023. URL `https://doi.org/10.5281/zenodo.10150663`.

[52] Leonardo Christino and Fernando V. Paulovich. ChatKG: Visualizing Temporal Patterns as Knowledge Graph. In Marco Angelini and Mennatallah El-Assady, editors, *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2023. ISBN 978-3-03868-222-6. doi: https://doi.org/10.2312/eurova.20231090.

[53] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. An overview of end-to-end entity resolution for big data. *ACM Comput. Surv.*, 53(6), dec 2020. ISSN 0360-0300. doi: https://doi.org/10.1145/3418896.

[54] Dan Clark. *Introducing Power BI*, pages 1–20. Apress, Berkeley, CA, 2020. ISBN 978-1-4842-5620-6. doi: https://doi.org/10.1007/978-1-4842-5620-6_1.

[55] Brian Clifton. *Advanced web metrics with Google Analytics*. John Wiley & Sons, 2012.

[56] Alessandro Comai. Decision-making support: the role of data visualization in analyzing complex systems. *World Future Review*, 6(4):477–484, 2014. doi: https://doi.org/10.1177/1946756715569233.

[57] Debby RE Cotton, Peter A Cotton, and J Reuben Shipway. Chatting and Cheating: Ensuring Academic Integrity in the Era of ChatGPT. *Innovations in Education and Teaching International*, pages 1–12, 2023. doi: https://doi.org/10.1080/14703297.2023.2190148.

[58] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading, 2010.

[59] Weiwei Cui, Xiaoyu Zhang, Yun Wang, He Huang, Bei Chen, Lei Fang, Haidong Zhang, Jian-Guan Lou, and Dongmei 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. doi: https://doi.org/10.1109/tvcg.2019.2934785.

[60] Edward Curry. *Dataspaces: Fundamentals, Principles, and Techniques*, pages 45–62. Springer, 2020. doi: https://doi.org/10.1007/978-3-030-29665-0_3.

[61] Edward Curry. *Real-time linked dataspaces: Enabling data ecosystems for intelligent systems*. Springer Nature, 2020. doi: https://doi.org/10.1007/978-3-030-29665-0_4.

[62] Sérgio Manuel Serra da Cruz, Maria Luiza M Campos, and Marta Mattoso. Towards a taxonomy of provenance in scientific workflow management systems. In *2009 Congress on Services-I*, pages 259–266. IEEE, 2009. doi: https://doi.org/10.1109/services-i.2009.18.

[63] Paulo Pinheiro da Silva, Silva Deborah, Deborah L McGuinness, and Rob Mccool. Knowledge provenance infrastructure. *Data Engineering Bulletin*, 24 - 4:26–32, 2003.

[64] Sarah Dahir and Abderrahim El Qadi. A query expansion method based on topic modeling and dbpedia features. *International Journal of Information Management Data Insights*, 1 / 2:100043, 2021. doi: https://doi.org/10.1016/j.jjimei.2021.100043.

[65] Sahraoui Dhelim, Huansheng Ning, and Nyothiri Aung. Compath: User interest mining in heterogeneous signed social networks for internet of people. *IEEE Internet of Things Journal*, 8(8):7024–7035, 2020. doi: https://doi.org/10.1109/jiot.2020.3037109.

[66] Rui Ding, Shi Han, Yong Xu, Haidong Zhang, and Dongmei Zhang. Quickinsights: Quick and automatic discovery of insights from multi-dimensional data. In *Proceedings of the 2019 International Conference on Management of Data*, pages 317–332, 2019. doi: https://doi.org/10.1145/3299869.3314037.

[67] Ugur Dogrusoz, Erhan Giral, Ahmet Cetintas, Ali Civril, and Emek Demir. A layout algorithm for undirected compound graphs. *Information Sciences*, 179(7):980–994, 2009. doi: https://doi.org/10.1016/j.ins.2008.11.017.

[68] Mennatallah El-Assady and Caterina Moruzzi. Which biases and reasoning pitfalls do explanations trigger? decomposing communication processes in human–ai interaction. *IEEE Computer Graphics and Applications*, 42(6):11–23, 2022. doi: https://doi.org/10.1109/mcg.2022.3200328.

[69] Inc. Element Labs. Lmstudio, accessed in 2023-11-06. `https://lmstudio.ai/`.

[70] Nik Everett. Loading wikipedia's search index for testing, 2016 (accessed February, 2020). URL `https://www.elastic.co/blog/loading-wikipedia`. `https://www.elastic.co/blog/loading-wikipedia`.

[71] Dirk Fahland. Process mining over multiple behavioral dimensions with event knowledge graphs. In *Process Mining Handbook*, pages 274–319. Springer, 2022. doi: https://doi.org/10.1007/978-3-031-08848-3_9.

[72] Paolo Federico, Markus Wagner, Alexander Rind, Albert Amor-Amorós, Silvia Miksch, and Wolfgang 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)*, pages 92–103, 2017. doi: https://doi.org/10.1109/VAST.2017.8585498.

[73] Christiane Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer, 2010. doi: https://doi.org/10.1007/978-90-481-8847-5_10.

[74] Dieter Fensel, Umutcan Şimşek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, and Alexander Wahler. Introduction: what is a knowledge graph? In *Knowledge Graphs*, pages 1–10. Springer, 2020. doi: https://doi.org/10.1007/978-3-030-37439-6_1.

[75] Alfio Ferrara, Andriy Nikolov, and François Scharffe. Data linking for the semantic web. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 7(3):46–76, 2011. doi: https://doi.org/10.4018/jswis.2011070103.

[76] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in neural information processing systems*, pages 2962–2970, 2015.

[77] Richard Feynman. Ebnf: A notation to describe syntax. *http://www. ics. uci. edu/˜ pattis/misc/ebnf2. pdf*, 2016.

[78] Mark S Fox and Jingwei Huang. Knowledge provenance. In *Advances in Artificial Intelligence: 17th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2004, London, Ontario, Canada, May 17-19, 2004. Proceedings 17*, pages 517–523. Springer, 2004. doi: https://doi.org/10.1007/978-3-540-24840-8_47.

[79] Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. Cypher: An evolving query language for property graphs. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1433–1445, 2018. doi: https://doi.org/10.1145/3183713.3190657.

[80] Michael Franklin, Alon Halevy, and David Maier. From databases to dataspaces: a new abstraction for information management. *ACM Sigmod Record*, 34(4):27–33, 2005.

[81] Max Franz, Christian T Lopes, Dylan Fong, Mike Kucera, Manfred Cheung, Metin Can Siper, Gerardo Huck, Yue Dong, Onur Sumer, and Gary D Bader. Cytoscape. js 2023 update: a graph theory library for visualization and analysis. *Bioinformatics*, 39(1):btad031, 2023. doi: https://doi.org/10.1093/bioinformatics/btad031.

[82] Takanori Fujiwara, Tarik Crnovrsanin, and Kwan-Liu Ma. Concise provenance of interactive network analysis. *Visual Informatics*, 2(4):213–224, 2018. doi: https://doi.org/10.1016/j.visinf.2018.12.002.

[83] Elisa Gabbert. Keywords vs. search queries: What's the difference?, accessed in 2022-05-06. `https://www.wordstream.com/blog/ws/2011/05/25/keywords-vs-search-queries`.

[84] Tong Gao, Jessica R Hullman, Eytan Adar, Brent Hecht, and Nicholas Diakopoulos. Newsviews: an automated pipeline for creating custom geovisualizations for news. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 3005–3014, 2014. doi: https://doi.org/10.1145/2556288.2557228.

[85] Sukhpal Singh Gill and Rupinder Kaur. Chatgpt: Vision and challenges. *Internet of Things and Cyber-Physical Systems*, 3:262–271, 2023. doi: https://doi.org/10.1016/j.iotcps.2023.05.004.

[86] David F Gleich. Pagerank beyond the web. *siam REVIEW*, 57(3):321–363, 2015. doi: https://doi.org/10.1137/140976649.

[87] Sultan Global. Tableau tutorial - global superstore performance dashboard, 2019. URL `https://www.youtube.com/embed/kIZDb_pHvX0?start=187&end=452`.

[88] Satish Gojare, Rahul Joshi, and Dhanashree Gaigaware. Analysis and design of selenium webdriver automation testing framework. *Procedia Computer Science*, 50:341–346, 2015. doi: https://doi.org/10.1016/j.procs.2015.04.038.

[89] Behzad Golshan, Alon Halevy, George Mihaila, and Wang-Chiew Tan. Data integration: After the teenage years. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI symposium on principles of database systems*, pages 101–106, 2017. doi: https://doi.org/10.1145/3034786.3056124.

[90] Steven R Gomez, Hua Guo, Caroline Ziemkiewicz, and David H Laidlaw. An insight-and task-based methodology for evaluating spatiotemporal visual analytics. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 63–72. IEEE, 2014. doi: https://doi.org/10.1109/vast.2014.7042482.

[91] Rodrigo Gonçalves and Carina Friedrich Dorneles. Automated expertise retrieval: a taxonomy-based survey and open issues. *ACM Computing Surveys (CSUR)*, 52(5):1–30, 2020. doi: https://doi.org/10.1145/3331000.

[92] Simon Gottschalk and Elena Demidova. Eventkg+ tl: creating cross-lingual timelines from an event-centric knowledge graph. In *European Semantic Web Conference*, pages 164–169. Springer, 2018. doi: https://doi.org/10.1007/978-3-319-98192-5_31.

[93] Danny Graham, 2023. URL `https://engagenovascotia.ca/`.

[94] M. Grootendorst. Keybert: Keyword extraction with bert, 2020 (accessed November, 2020). `https://github.com/MaartenGr/KeyBERT`.

[95] growhackscale. What is a search query? (definition) - seo glossary, accessed in 2022-05-06. `https://growhackscale.com/glossary/search-queries`.

[96] Christoph Gröger, Holger Schwarz, and Bernhard Mitschang. The deep data warehouse: Link-based integration and enrichment of warehouse data and unstructured content. In *2014 IEEE 18th International Enterprise Distributed Object Computing Conference*, pages 210–217, 2014. doi: https://doi.org/10.1109/EDOC.2014.36.

[97] Ken Gu, Madeleine Grunde-McLaughlin, Andrew M McNutt, Jeffrey Heer, and Tim Althoff. How Do Data Analysts Respond to AI Assistance? a wizard-of-oz study. *arXiv preprint arXiv:2309.10108*, 2023. doi: https://doi.org/10.48550/arXiv.2309.10108.

[98] Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Yutao Zeng, Xiaolong Jin, and Jiafeng Guo. What is event knowledge graph: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022. doi: https://doi.org/10.1109/tkde.2022.3180362.

[99] Hua Guo, Steven R Gomez, Caroline Ziemkiewicz, and David H Laidlaw. A case study using visualization interaction logs and insight metrics to understand how analysts arrive at insights. *IEEE transactions on visualization and computer graphics*, 22(1):51–60, 2016. doi: https://doi.org/10.1109/tvcg.2015.2467613.

[100] Vishu Gupta, Kamal Choudhary, Yuwei Mao, Kewei Wang, Francesca Tavazza, Carelyn Campbell, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. MPpredictor: an Artificial Intelligence-Driven Web Tool for Composition-Based Material Property Prediction. *Journal of Chemical Information and Modeling*, 63:1865–1871, 2023. doi: https://doi.org/10.1021/acs.jcim.3c00307.

[101] Isabelle Guyon, Kristin Bennett, Gavin Cawley, Hugo Jair Escalante, Sergio Escalera, Tin Kam Ho, N ú ria Macia, Bisakha Ray, Mehreen Saeed, Alexander Statnikov, et al. Design of the 2015 chalearn automl challenge. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2015.

[102] Isabelle Guyon, Imad Chaabane, Hugo Jair Escalante, Sergio Escalera, Damir Jajetic, James Robert Lloyd, N ú ria Maci à, Bisakha Ray, Lukasz Romaszko, Mich è le Sebag, et al. A brief review of the chalearn automl challenge: any-time any-dataset learning without human intervention. In *Workshop on Automatic Machine Learning*, pages 21–30, 2016.

[103] Wentao Han, Youshan Miao, Kaiwei Li, Ming Wu, Fan Yang, Lidong Zhou, Vijayan Prabhakaran, Wenguang Chen, and Enhong Chen. Chronos: a graph engine for temporal graph analysis. In *Proceedings of the Ninth European Conference on Computer Systems*, pages 1–14, 2014. doi: https://doi.org/10.1145/2592798.2592799.

[104] Hossein Hassani and Emmanuel Sirmal Silva. The role of chatgpt in data science: how ai-assisted conversational interfaces are revolutionizing the field. *Big data and cognitive computing*, 7(2):62, 2023. doi: https://doi.org/10.3390/bdcc7020062.

[105] Helen Ai He, Jagoda Walny, Sonja Thoma, Wesley J. Willett, and Sheelagh Carpendale. Discussing open energy data and data visualizations with canadians. *University of Calgary, Faculty of Science, Department of Computer Science, Calgary, AB. 1-61*, 2019.

[106] Xing He, Rui Zhang, Rubina Rizvi, Jake Vasilakes, Xi Yang, Yi Guo, Zhe He, Mattia Prosperi, Jinhai Huo, Jordan Alpert, et al. Aloha: developing an interactive graph-based visualization for dietary supplement knowledge graph through user-centered design. *BMC medical informatics and decision making*, 19(4):1–18, 2019. doi: https://doi.org/10.1186/s12911-019-0857-1.

[107] Jeffrey Heer, Jock Mackinlay, Chris Stolte, and Maneesh Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE transactions on visualization and computer graphics*, 14(6):1189–1196, 2008. doi: https://doi.org/10.1109/tvcg.2008.137.

[108] Gladys M Hilasaca, Wilson E Marcílio-Jr, Danilo M Eler, Rafael M Martins, and Fernando V Paulovich. Overlap removal of dimensionality reduction scatterplot layouts. *arXiv preprint arXiv:1903.06262*, 2019. doi: https://doi.org/10.48550/arXiv.1903.06262.

[109] Elisa L Hill-Yardin, Mark R Hutchinson, Robin Laycock, and Sarah J Spencer. A Chat (GPT) about the Future of Scientific Publishing. *Brain Behav Immun*, 110:152–154, 2023. doi: https://doi.org/10.1016/j.bbi.2023.02.022.

[110] Harry Hochheiser and Ben Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004. doi: https://doi.org/10.1057/palgrave.ivs.9500061.

[111] Orland Hoeber, Xue-Dong Yang, and Yiyu Yao. Visualization support for interactive query refinement. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pages 657–665. IEEE, 2005. doi: https://doi.org/10.1109/wi.2005.158.

[112] Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. Applying pragmatics principles for interaction with visual analytics. *IEEE transactions on visualization and computer graphics*, 24(1):309–318, 2018. doi: https://doi.org/10.1109/tvcg.2017.2744684.

[113] Dandan Huang, Melanie Tory, Bon Adriel Aseniero, Lyn Bartram, Scott Bateman, Sheelagh Carpendale, Anthony Tang, and Robert Woodbury. Personal visualization and personal visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 21(3):420–433, 2015. doi: https://doi.org/10.1109/tvcg.2014.2359887.

[114] Jiangjie Huang, Xiaoyu Zhang, and Yongchuan Tang. Using Linkography to Quantitatively Analyze the Design Ideation of AI Agents. In *2023* 15^{th} *International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, pages 241–244. IEEE, 2023. doi: https://doi.org/10.1109/IHMSC58761.2023.00063.

[115] Jessica Hullman, Nicholas Diakopoulos, and Eytan Adar. Contextifier: automatic generation of annotated stock visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2707–2716, 2013. doi: https://doi.org/10.1145/2470654.2481374.

[116] Filip Ilievski, Daniel Garijo, Hans Chalupsky, Naren Teja Divvala, Yixiang Yao, Craig Rogers, Rongpeng Li, Jun Liu, Amandeep Singh, Daniel Schwabe, et al. Kgtk: a toolkit for large knowledge graph manipulation and analysis. In *International Semantic Web Conference*, pages 278–293. Springer, 2020. doi: https://doi.org/10.1007/978-3-030-62466-8_18.

[117] Woojeong Jin, He Jiang, Meng Qu, Tong Chen, Changlin Zhang, Pedro Szekely, and Xiang Ren. Recurrent event network : Global structure inference over temporal knowledge graph, 2020. URL `https://openreview.net/forum?id=SyeyF0VtDr`.

[118] Zhihua Jin, Yong Wang, Qianwen Wang, Yao Ming, Tengfei Ma, and Huamin Qu. Gnnlens: A visual analytics approach for prediction error diagnosis of graph neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 29(6):3024–3038, 2023. doi: https://doi.org/10.1109/TVCG.2022.3148107.

[119] Kushal Kafle, Robik Shrestha, Scott Cohen, Brian Price, and Christopher Kanan. Answering questions about data visualizations using efficient bimodal fusion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1498–1507, 2020. doi: https://doi.org/10.1109/wacv45572.2020.9093494.

[120] Yvonne Kammerer and Maja Bohnacker. Children's web search with google: the effectiveness of natural language queries. In *proceedings of the 11th International Conference on Interaction Design and Children*, pages 184–187, 2012. doi: https://doi.org/10.1145/2307096.2307121.

[121] Larry Kaufman and John Ågren. CALPHAD, First and Second Generation–Birth of the Materials Genome. *Scripta Materialia*, 70:3–6, 2014. doi: https://doi.org/10.1016/j.scriptamat.2012.12.003.

[122] Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. Mastering the information age: solving problems with visual analytics. *VisMaster - Eurographics Association*, 2010. doi: https://doi.org/10.2312/14803.

[123] Daniel A Keim, Florian Mansmann, and Jim Thomas. Visual analytics: how much visualization and how much analytics? *ACM SIGKDD Explorations Newsletter*, 11(2):5–8, 2010. doi: https://doi.org/10.1145/1809400.1809403.

[124] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386, 2005. doi: https://doi.org/10.1007/s10115-004-0154-9.

[125] Taraneh Khazaei and Orland Hoeber. Supporting academic search tasks through citation visualization and exploration. *International Journal on Digital Libraries*, 18(1):59–72, 2017. doi: https://doi.org/10.1007/s00799-016-0170-x.

[126] Dae Hyun Kim, Enamul Hoque, Juho Kim, and Maneesh Agrawala. Facilitating document reading by linking text and tables. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 423–434, 2018. doi: https://doi.org/10.1145/3242587.3242617.

[127] Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. Answering questions about charts and generating visual explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020. doi: https://doi.org/10.1145/3313831.3376467.

[128] Cole Nussbaumer Knaflic. *Storytelling with data: A data visualization guide for business professionals*. John Wiley & Sons, 2015.

[129] Oleksii Kononenko, Olga Baysal, Reid Holmes, and Michael W. Godfrey. Mining modern repositories with Elasticsearch. In *11th Working Conference on Mining Software Repositories, MSR 2014 - Proceedings*, pages 328–331, New York, New York, USA, 5 2014. Association for Computing Machinery, Inc. doi: https://doi.org/10.1145/2597073.2597091.

[130] Tim Kraska. Northstar: An interactive data science system. *Proceedings of the VLDB Endowment*, 11(12):2150–2164, 2018. doi: https://doi.org/10.14778/3229863.3240493.

[131] Krazydawg. Country to continent. `https://www.kaggle.com/datasets/statchaitya/country-to-continent`, 2017.

[132] Yngve Sekse Kristiansen and Stefan Bruckner. Visception: An interactive visual framework for nested visualization design. *Computers & Graphics*, 92: 13–27, 2020. doi: https://doi.org/10.1016/j.cag.2020.08.007.

[133] Bum Chul Kwon, Florian Stoffel, Dominik Jäckle, Bongshin Lee, and Daniel Keim. Visjockey: Enriching data stories through orchestrated interactive visualization. In *Poster Compendium of the Computation+ Journalism Symposium*, volume 3, page 3, 2014.

[134] Arash Habibi Lashkari, Fereshteh Mahdavi, and Vahid Ghomi. A boolean model in information retrieval for search engines. In *2009 International Conference on Information Management and Engineering*, pages 385–389. IEEE, 2009. doi: https://doi.org/10.1109/icime.2009.101.

[135] Fernando Paulovich Leonardo Christino. Vakg sample implementation, dec 2021. URL `https://doi.org/10.5281/zenodo.8124217`.

[136] Haotian Li, Yong Wang, Songheng Zhang, Yangqiu Song, and Huamin Qu. Kg4vis: A knowledge graph-based approach for visualization recommendation. *IEEE Transactions on Visualization and Computer Graphics*, 2021. doi: https://doi.org/10.1109/tvcg.2021.3114863.

[137] Harry Li, Gabriel Appleby, Camelia Daniela Brumar, Remco Chang, and Ashley Suh. Knowledge graphs in practice: Characterizing their users, challenges, and visualization opportunities. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):584–594, 2024. doi: https://doi.org/10.1109/TVCG.2023.3326904.

[138] Miaoran Li, Baolin Peng, and Zhu Zhang. Self-Checker: Plug-and-Play Modules for Fact-Checking with Large Language Models. *arXiv preprint arXiv:2305.14623*, 2023. doi: https://doi.org/10.48550/arXiv.2305.14623.

[139] Quan Li, Kristanto Sean Njotoprawiro, Hammad Haleem, Qiaoan Chen, Chris Yi, and Xiaojuan Ma. Embeddingvis: A visual analytics approach to comparative network embedding inspection. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 48–59. IEEE, 2018. doi: https://doi.org/10.1109/vast.2018.8802454.

[140] Allen Yilun Lin, Joshua Ford, Eytan Adar, and Brent Hecht. Vizbywiki: mining data visualizations from the web to enrich news articles. In *Proceedings of the 2018 World Wide Web Conference*, pages 873–882, 2018. doi: https://doi.org/10.1145/3178876.3186135.

[141] Heather Richter Lipford, Felesia Stukes, Wenwen Dou, Matthew E Hawkins, and Remco Chang. Helping users recall their reasoning process. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pages 187–194. IEEE, 2010. doi: https://doi.org/10.1109/vast.2010.5653598.

[142] Google LLC. Google search, accessed in 2022-05-06. `https://www.google.com/`.

[143] Edward Loper and Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, page 63–70, USA, 2002. Association for Computational Linguistics. doi: https://doi.org/10.3115/1118108.1118117.

[144] Brady D Lund and Ting Wang. Chatting about ChatGPT: how may ai and gpt impact academia and libraries? *Library Hi Tech News*, 2023. doi: https://doi.org/10.2139/ssrn.4333415.

[145] Yuyu Luo, Xuedi Qin, Nan Tang, Guoliang Li, and Xinran Wang. Deepeye: Creating good data visualizations by keyword search. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1733–1736, 2018. doi: https://doi.org/10.1145/3183713.3193545.

[146] Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. Chatgpt as a factual inconsistency evaluator for abstractive text summarization. *arXiv preprint arXiv:2303.15621*, 2023. doi: https://doi.org/10.48550/arXiv.2303.15621.

[147] Meinard M ü ller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007. doi: https://doi.org/10.1007/978-3-540-74048-3_4.

[148] Karthic Madanagopal, Eric D Ragan, and Perakath Benjamin. Analytic provenance in practice: The role of provenance in real-world visualization and data analysis environments. *IEEE Computer Graphics and Applications*, 39 (6):30–45, 2019. doi: https://doi.org/10.1109/mcg.2019.2933419.

[149] Paula Maddigan and Teo Susnjak. Chat2vis: Generating data visualizations via natural language using chatgpt, codex and gpt-3 large language models. *IEEE Access*, 11:45181–45193, 2023. doi: https://doi.org/10.1109/ACCESS.2023.3274199.

[150] Amgad Madkour, Walid G Aref, Faizan Ur Rehman, Mohamed Abdur Rahman, and Saleh Basalamah. A survey of shortest-path algorithms. *arXiv preprint arXiv:1705.02044*, 2017. doi: https://doi.org/10.48550/arXiv.1705.02044.

[151] Jorge Martinez-Gil. Automated knowledge base management: A survey. *Computer Science Review*, 18:1–9, 2015. doi: https://doi.org/10.31219/osf.io/gyft8.

[152] Andreas Mathisen, Tom Horak, Clemens Nylandsted Klokmose, Kaj Grønbæk, and Niklas Elmqvist. Insideinsights: Integrating data-driven reporting in collaborative visual analytics. In *Computer Graphics Forum*, volume 38, pages 649–661. Wiley Online Library, 2019. doi: https://doi.org/10.1111/cgf.13717.

[153] Atif Memon, Ishan Banerjee, and Adithya Nagarajan. What test oracle should i use for effective gui testing? In *18th IEEE International Conference on Automated Software Engineering, 2003. Proceedings.*, pages 164–173. IEEE, 2003. doi: https://doi.org/10.1109/ase.2003.1240304.

[154] Ronald Metoyer, Qiyu Zhi, Bart Janczuk, and Walter Scheirer. Coupling story to visualization: Using textual analysis as a bridge between data and interpretation. In *23rd International Conference on Intelligent User Interfaces*, pages 503–507, 2018. doi: https://doi.org/10.1145/3172944.3173007.

[155] Daniel Z Meyer and Leanne M Avery. Excel as a qualitative data analysis tool. *Field methods*, 21(1):91–112, 2009. doi: https://doi.org/10.1177/1525822x08323985.

[156] Aditya Mogadala, Marimuthu Kalimuthu, and Dietrich Klakow. Trends in integration of vision and language research: A survey of tasks, datasets, and methods. *Journal of Artificial Intelligence Research*, 71:1183–1317, 2021. doi: https://doi.org/10.1613/jair.1.11688.

[157] Shayan Monadjemi, Mengtian Guo, David Gotz, Roman Garnett, and Alvitta Ottley. Human-Computer Collaboration for Visual Analytics: an Agent-based Framework. *Computer Graphics Forum*, 2023. ISSN 1467-8659. doi: https://doi.org/10.1111/cgf.14823.

[158] Michalis Mountantonakis and Yannis Tzitzikas. Large-scale semantic integration of linked data: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–40, 2020. doi: https://doi.org/10.1145/3345551.

[159] Daniel G Murray. *Tableau your data!: fast and easy visual analysis with tableau software*. John Wiley & Sons, 2013.

[160] Igor Myroshnichenko and Marguerite C Murphy. Mapping er schemas to owl ontologies. In *2009 IEEE International Conference on Semantic Computing*, pages 324–329. IEEE, 2009. doi: https://doi.org/10.1109/icsc.2009.61.

[161] United Nations. United nations datasets, accessed in 2022-05-06. `https://data.un.org/`.

[162] Neo4j. Neo4j - the world's leading graph database, 2012. URL `http://neo4j.org/`.

[163] Mário Popolin Neto and Fernando V. Paulovich. Explainable matrix - visualization for global and local interpretability of random forest classification ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 27(2): 1427–1437, 2021. doi: https://doi.org/10.1109/TVCG.2020.3030354.

[164] Hoang Long Nguyen, Dang Thinh Vu, and Jason J Jung. Knowledge graph fusion for smart systems: A survey. *Information Fusion*, 61:56–70, 2020. doi: https://doi.org/10.1016/j.inffus.2020.03.014.

[165] Phong H Nguyen, Kai Xu, Andy Bardill, Betul Salman, Kate Herd, and BL William Wong. Sensemap: Supporting browser-based online sensemaking through analytic provenance. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 91–100. IEEE, 2016. doi: https://doi.org/10.1109/vast.2016.7883515.

[166] Osvaldo N Oliveira Jr, David Beljonne, Stanislaus S Wong, and Kirk S Schanze. Forum on Artificial Intelligence/Machine Learning for Design and Development of Applied Materials. *ACS Applied Materials & Interfaces*, 13:53301–53302, 2021. doi: https://doi.org/10.1021/acsami.1c18225.

[167] Randal S Olson, Nathan Bartley, Ryan J Urbanowicz, and Jason H Moore. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 485–492. ACM, 2016. doi: https://doi.org/10.1145/2908812.2908918.

[168] Jessie Ooi, Xiuqin Ma, Hongwu Qin, and Siau Chuin Liew. A survey of query expansion, query suggestion and query refinement techniques. In *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*, pages 112–117. IEEE, 2015. doi: https://doi.org/10.1109/icsecs.2015.7333094.

[169] OpenAI. Chatgpt, 2023 (accessed September, 2023). URL `https://openai.com/chatgpt`. `https://openai.com/chatgpt`.

[170] Charles Packer, Vivian Fang, Shishir G Patil, Kevin Lin, Sarah Wooders, and Joseph E Gonzalez. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*, 2023. doi: https://doi.org/10.48550/arXiv.2310.08560.

[171] Deokgun Park, Mohamed Suhail, Minsheng Zheng, Cody Dunne, Eric Ragan, and Niklas Elmqvist. Storyfacets: A design study on storytelling with visualizations for collaborative data analysis. *Information Visualization*, 21(1):3–16, 2022. doi: https://doi.org/10.1177/14738716211032653.

[172] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023. doi: https://doi.org/10.1145/3586183.3606763.

[173] Fernando V Paulovich, Maria Cristina F De Oliveira, and Osvaldo N Oliveira Jr. A future with Ubiquitous Sensing and Intelligent Systems. *ACS sensors*, 3:1433–1438, 2018. doi: https://doi.org/10.1021/acssensors.8b00276.

[174] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. doi: https://doi.org/10.3115/v1/d14-1162.

[175] Jan Polowinski and Martin Voigt. Viso: A shared, formal knowledge base as a foundation for semi-automatic infovis systems. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 1791–1796. Association for Computing Machinery, 2013. doi: https://doi.org/10.1145/2468356.2468677.

[176] Xiafei Qiu, Wubin Cen, Zhengping Qian, You Peng, Ying Zhang, Xuemin Lin, and Jingren Zhou. Real-time constrained cycle detection in large dynamic graphs. *Proceedings of the VLDB Endowment*, 11(12):1876–1888, 2018. doi: https://doi.org/10.14778/3229863.3229874.

[177] Eric D Ragan, Alex Endert, Jibonananda Sanyal, and Jian Chen. Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes. *IEEE transactions on visualization and computer graphics*, 22(1):31–40, 2015. doi: https://doi.org/10.1109/tvcg.2015.2467551.

[178] Hooman H Rashidi, Nam Tran, Samer Albahra, and Luke T Dang. Machine learning in health care and laboratory medicine: General overview of supervised learning and auto-ml. *International Journal of Laboratory Hematology*, 43:15–22, 2021. doi: https://doi.org/10.1111/ijlh.13537.

[179] Partha Pratim Ray. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 2023. doi: https://doi.org/10.1016/j.iotcps.2023.04.003.

[180] General Data Protection Regulation. General data protection regulation (gdpr). *Intersoft Consulting, Accessed in October*, 24(1), 2018.

[181] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[182] Michele Reilly and Santi Thompson. Reverse image lookup: assessing digital library users and reuses. *Journal of Web Librarianship*, 11(1):56–68, 2017. doi: https://doi.org/10.1080/19322909.2016.1223573.

[183] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. doi: https://doi.org/10.18653/v1/d19-1410.

[184] Sima Rezaeipourfarsangi, Ningyuan Pei, Ehsan Sherkat, and Evangelos Milios. Interactive clustering and high-recall information retrieval using language models. In *Proceedings of the 2022 International Conference on Advanced Visual Interfaces*, pages 1–5, 2022. doi: https://doi.org/10.1145/3531073.3531174.

[185] Anna Rosling Ronnlund and Ola Rosling. Free software for a world in motion. In *Proceedings. Second International Conference on Creating, Connecting and Collaborating through Computing, 2004.*, pages 14–19. IEEE, 2004. doi: https://doi.org/10.1109/c5.2004.1314363.

[186] Hans Rosling. Data - gapminder.org, 2012.

[187] Hans Rosling. Geography related dataset from gapminder, 2018 (accessed February, 2020). `https://www.gapminder.org/data/geo/`.

[188] Hans Rosling. Gapminder - life expectancy vs income bubble-chart, accessed in 2022-05-06. `https://tinyurl.com/gapminderbubblechart`.

[189] Hans Rosling. Gapminder - usa's life expectancy line-chart, accessed in 2022-05-06. URL `tinyurl.com/gapminderlinechart`.

[190] Ola Rosling Rosling, Hans and Anna Rosling Rönnlund. *Factfulness: Ten Reasons We're Wrong About the World - and Why Things Are Better Than You Think*. New York: Flatiron Books, 2018.

[191] Prasan Roy, Mukesh Mohania, Bhuvan Bamba, and Shree Raman. Towards automatic association of relevant unstructured content with structured query results. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 405–412. Association for Computing Machinery, 2005. doi: https://doi.org/10.1145/1099554.1099676.

[192] Tony Russell-Rose and Philip Gooch. 2dsearch: A visual approach to search strategy formulation. *Design of Experimental Search and Information REtrieval Systems (DESIRES 2018)*, 2018.

[193] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A Keim. Knowledge generation model for visual analytics. *IEEE transactions on visualization and computer graphics*, 20(12):1604–1613, 2014. doi: https://doi.org/10.1109/tvcg.2014.2346481.

[194] Dominik Sacha, Ina Boesecke, Johannes Fuchs, and Daniel A. Keim. Analytic Behavior and Trust Building in Visual Analytics. In Enrico Bertini, Niklas Elmqvist, and Thomas Wischgoll, editors, *EuroVis 2016 - Short Papers*. The Eurographics Association, 2016. ISBN 978-3-03868-014-7. doi: 10.2312/eurovisshort.20161176.

[195] Dominik Sacha, Matthias Kraus, Daniel A Keim, and Min Chen. Vis4ml: An ontology for visual analytics assisted machine learning. *IEEE transactions on visualization and computer graphics*, 25(1):385–395, 2018. doi: https://doi.org/10.1109/tvcg.2018.2864838.

[196] Amardeo Sarma. Hans rosling brought data to life, showed our misconceptions about the world. *Skeptical Inquirer*, 41(4):9–10, 2017.

[197] David S. Sawicki and William J. Craig. The democratization of data: Bridging the gap for community groups. *Journal of the American Planning Association*, 62(4):512–523, 1996. ISSN 01944363. doi: https://doi.org/10.1080/01944369608975715.

[198] Harrisen Scells and Guido Zuccon. Searchrefiner: A query visualisation and understanding tool for systematic reviews. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1939–1942, 2018. doi: https://doi.org/10.1145/3269206.3269215.

[199] Roger Schneider. Survey of peaks/valleys identification in time series. *Department of Informatics, University of Zurich, Switzerland*, 2011.

[200] Dennis Schoeneborn. The pervasive power of powerpoint: How a genre of professional communication permeates organizational communication. *Organization Studies*, 34(12):1777–1801, 2013. doi: https://doi.org/10.1177/0170840613485843.

[201] Ehsan Sherkat, Seyednaser Nourashrafeddin, Evangelos E Milios, and Rosane Minghim. Interactive document clustering revisited: A visual analytics approach. In *23rd International Conference on Intelligent User Interfaces*, pages 281–292, 2018. doi: https://doi.org/10.1145/3172944.3172964.

[202] Gao Shu, Nick J Avis, and O Rana. Investigating visualization ontologies. In *Proceedings of the UK e-Science All Hands Meeting*, volume 2006, 2006.

[203] communities Silva, FilipiNetwork, topics for the period N, Diego R Amancio, Maria Bardosova, Luciano da F Costa, and Osvaldo N Oliveira Jr. Using Network Science and Text Analytics to Produce Surveys in a Scientific Topic. *Journal of Informetrics*, 10:487–502, 2016. doi: https://doi.org/10.1016/j.joi.2016.03.008.

[204] Yogesh L Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *ACM Sigmod Record*, 34(3):31–36, 2005. doi: http://dx.doi.org/10.1145/1084805.1084812.

[205] Simeon Simoff, Michael H Böhlen, and Arturas Mazeika. *Visual data mining: theory, techniques and tools for visual analytics*, volume 4404. Springer Science & Business Media, 2008. doi: https://doi.org/10.1007/978-3-540-71080-6_1.

[206] Melvyn L Smith, Lyndon N Smith, and Mark F Hansen. The Quiet Revolution in Machine Vision- a State-of-the-Art Survey Paper, including Historical Review, Perspectives, and Future Directions. *Computers in Industry*, 130: 103472, 2021. doi: https://doi.org/10.1016/j.compind.2021.103472.

[207] Michael E Smoot, Keiichiro Ono, Johannes Ruscheinski, Peng-Liang Wang, and Trey Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27(3):431–432, 2011. doi: https://doi.org/10.1093/bioinformatics/btq675.

[208] Amílcar Soares, Jordan Rose, Mohammad Etemad, Chiara Renso, and Stan Matwin. Vista: A visual analytics platform for semantic annotation of trajectories. In *Proceedings of the 22nd international conference on extending database technology (EDBT)*, 2019. doi: https://doi.org/10.5441/002/edbt.2019.58.

[209] Thilo Spinner, Udo Schlegel, Hanna Schäfer, and Mennatallah El-Assady. Explainer: A visual analytics framework for interactive and explainable machine learning. *IEEE transactions on visualization and computer graphics*, 26 (1):1064–1074, 2019. doi: https://doi.org/10.1109/tvcg.2019.2934629.

[210] Arjun Srinivasan and John Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE transactions on visualization and computer graphics*, 24(1):511–521, 2017. doi: https://doi.org/10.1109/tvcg.2017.2745219.

[211] Arjun Srinivasan, Steven M Drucker, Alex Endert, and John Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE transactions on visualization and computer graphics*, 25(1):672–681, 2019. doi: https://doi.org/10.1109/tvcg.2018.2865145.

[212] Robert St. Amant and Paul R Cohen. Intelligent support for exploratory data analysis. *Journal of Computational and Graphical Statistics*, 7(4):545–558, 1998. doi: https://doi.org/10.2307/1390682.

[213] Ashley Suh, Yilan Jiang, Ab Mosca, Eugene Wu, and Remco Chang. A grammar for hypothesis-driven visual analysis. *arXiv preprint arXiv:2204.14267*, 2022. doi: https://doi.org/10.48550/arXiv.2204.14267.

[214] Nicole Sultanum, Michael Brudno, Daniel Wigdor, and Fanny Chevalier. More text please! understanding and supporting the use of visualization for clinical text overview. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–13, 2018. doi: https://doi.org/10.1145/3173574.3173996.

[215] Nigar M Shafiq Surameery and Mohammed Y Shakor. Use chat gpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290*, 3(01):17–22, 2023. doi: https://doi.org/10.55529/ijitc.31.17.22.

[216] Shree Hari Sureshbabu, Manas Sajjan, Sangchul Oh, and Sabre Kais. Implementation of Quantum Machine Learning for Electronic Structure Calculations of Periodic Systems on Quantum Computing Devices. *Journal of Chemical Information and Modeling*, 61:2667–2674, 2021. doi: https://doi.org/10.1021/acs.jcim.1c00294.

[217] szrlee. Kaggle dataset - djia 30 stock time series, 2017. URL `https://www.kaggle.com/datasets/szrlee/stock-time-series-20050101-to-20171231`.

[218] Tableau. Tableau training dataset - global superstore, 2016. URL `http://www.tableau.com/sites/default/files/training/global_superstore.zip`.

[219] Bo Tang, Shi Han, Man Lung Yiu, Rui Ding, and Dongmei Zhang. Extracting top-k insights from multi-dimensional data. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1509–1524, 2017. doi: https://doi.org/10.1145/3035918.3035922.

[220] NYC Taxi and Limousine Commission (TLC). Kaggle dataset - new york city taxt trip duration, 2016. URL `https://www.kaggle.com/competitions/nyc-taxi-trip-duration/data`.

[221] Elastic Documentation Team. Simple query string query — elasticsearch reference [7.6] — elastic, 2018 (accessed February 15, 2020). `https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-simple-query-string-query.html`.

[222] Wikipedia Team. Wikimedia downloads, 2020 (accessed February, 2020). URL `https://dumps.wikimedia.org/other/cirrussearch/`.

[223] Wikipedia Team. Wikimedia downloads, 2020 (accessed February, 2020). URL `https://dumps.wikimedia.org/other/cirrussearch/`.

[224] Wikipedia Team and Contributors. Wikipedia - the free encyclopedia, accessed in 2022-05-06. URL `https://en.wikipedia.org/wiki/Main_Page`.

[225] Celonis, Inc. Make, 2023 (accessed September, 2023). URL https://www.make.com/. https://www.make.com/.

[226] LangChain, Inc. AgentGPT, 2022 (accessed September, 2023). URL https://www.langchain.com/. https://www.langchain.com/.

[227] Leonardo Interactive Pty Ltd. Leonardo.ai, 2023 (accessed September, 2023). URL https://leonardo.ai/. https://leonardo.ai/.

[228] Reworkd AI, Inc. AgentGPT, 2023 (accessed September, 2023). URL https://reworkd.ai/. https://reworkd.ai/.

[229] Wolfram Alpha LLC. Wolfram Alpha, 2016 (accessed September, 2023). URL http://www.wolframalpha.com/input/?i=2%2B2. http://www.wolframalpha.com/input/?i=2%2B2.

[230] @yoheinakajima. BabyAGI, 2023 (accessed September, 2023). URL https://github.com/yoheinakajima/babyagi. https://github.com/yoheinakajima/babyagi.

[231] James J Thomas and Kristin A Cook. A visual analytics agenda. *IEEE computer graphics and applications*, 26(1):10–13, 2006. doi: https://doi.org/10.1109/mcg.2006.5.

[232] Jim Thomas and Joe Kielman. Challenges for visual analytics. *Information Visualization*, 8(4):309–314, 2009. doi: https://doi.org/10.1057/ivs.2009.26.

[233] Wil MP Van der Aalst. Extracting event data from databases to unleash process mining. *BPM-Driving innovation in a digital world*, pages 105–128, 2015. doi: https://doi.org/10.1007/978-3-319-14430-6_8.

[234] Wil MP Van der Aalst and Anton JMM Weijters. Process mining: a research agenda. *Computers in industry*, 53(3):231–244, 2004. doi: https://doi.org/10.1016/j.compind.2003.10.001.

[235] Richard Van Noorden and Jeffrey M Perkel. Ai and science: what 1,600 researchers think. *Nature*, 621(7980):672–675, 2023. doi: https://doi.org/10.1038/d41586-023-02980-0.

[236] Jarke J Van Wijk. The value of visualization. In *VIS 05. IEEE Visualization, 2005.*, pages 79–86. IEEE, 2005. doi: https://doi.org/10.1109/VISUAL.2005.1532781.

[237] Jarke J van Wijk. Evaluation: A challenge for visual analytics. *Computer*, 46 (7):56–60, 2013. doi: https://doi.org/10.1109/mc.2013.151.

[238] Yuli Vasiliev. *Natural Language Processing with Python and SpaCy: A Practical Introduction*. No Starch Press, 2020.

[239] Giannis Vassiliou, Nikolaos Papadakis, and Haridimos Kondylakis. Summarygpt: Leveraging chatgpt for summarizing knowledge graphs. In *European Semantic Web Conference*, pages 164–168. Springer, 2023. doi: https://doi.org/10.1007/978-3-031-43458-7_31.

[240] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020. doi: https://doi.org/10.1038/s41592-019-0686-2.

[241] Tatiana von Landesberger, Sebastian Fiebig, Sebastian Bremm, Arjan Kuijper, and Dieter W Fellner. Interaction taxonomy for tracking of user actions in visual analytics applications. In *Handbook of Human Centric Visualization*, pages 653–670. Springer, 2014. doi: https://doi.org/10.1007/978-1-4614-7485-2_26.

[242] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Michal Walczak, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. Informed machine learning – a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2023. doi: https://doi.org/10.1109/TKDE.2021.3079836.

[243] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 417–426, 2018. doi: https://doi.org/10.1145/3269206.3271739.

[244] Jinjiang Wang, Yilin Li, Robert X Gao, and Fengli Zhang. Hybrid Physics-Based and Data-Driven Models for Smart Manufacturing: Modelling, Simulation, and Explainability. *Journal of Manufacturing Systems*, 63:381–391, 2022. doi: http://dx.doi.org/10.1016/j.jmsy.2022.04.004.

[245] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017. doi: https://doi.org/10.1109/tkde.2017.2754499.

[246] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*, 2023. doi: https://doi.org/10.48550/arXiv.2302.10205.

[247] Wikipedia contributors. Wikipedia, the free encyclopedia, 2004. [Online; accessed 22-July-2004].

[248] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang (Eric) Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv.2308.08155*, August 2023. doi: https://doi.org/10.48550/arXiv.2308.08155.

[249] Sean Wu, Michael Koo, Lesley Blum, Andy Black, Liyo Kao, Fabien Scalzo, and Ira Kurtz. A Comparative Study of Open-Source Large Language Models, GPT-4 and Claude 2: Multiple-Choice Test Taking in Nephrology. *arXiv preprint arXiv:2308.04709*, 2023. doi: https://doi.org/10.48550/arXiv.2308.04709.

[250] Kai Xu, Simon Attfield, TJ Jankun-Kelly, Ashley Wheat, Phong H Nguyen, and Nallini Selvaraj. Analytic provenance for sensemaking: A research agenda. *IEEE computer graphics and applications*, 35(3):56–64, 2015. doi: https://doi.org/10.1109/mcg.2015.50.

[251] Kai Xu, Alvitta Ottley, Conny Walchshofer, Marc Streit, Remco Chang, and John Wenskovitch. Survey on the analysis of user interactions and visualization provenance. In *Computer Graphics Forum*, volume 39 - 3, pages 757–783. Wiley Online Library, 2020. doi: https://doi.org/10.1111/cgf.14035.

[252] Chao Yang, Zengyou He, and Weichuan Yu. Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis. *BMC Bioinformatics*, 10:4, 1 2009. ISSN 14712105. doi: https://doi.org/10.1186/1471-2105-10-4.

[253] Hui Yang, Sifu Yue, and Yunzhong He. Auto-GPT for Online Decision Making: Benchmarks and Additional Opinions. *arXiv preprint arXiv:2306.02224*, 2023. doi: https://doi.org/10.48550/arXiv.2306.02224.

[254] Peipei Yi, Byron Choi, Sourav S Bhowmick, and Jianliang Xu. Autog: a visual query autocompletion framework for graph databases. *The VLDB Journal*, 26 (3):347–372, 2017. doi: https://doi.org/10.1007/s00778-017-0454-9.

[255] B. Yu and C. T. Silva. Flowsense: A natural language interface for visual data exploration within a dataflow system. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1–11, 2020. doi: https://doi.org/10.1109/tvcg.2019.2934668.

[256] Jing Yu, Weifeng Zhang, Yuhang Lu, Zengchang Qin, Yue Hu, Jianlong Tan, and Qi Wu. Reasoning on the relation: Enhancing visual representation for visual question answering and cross-modal retrieval. *IEEE Transactions on Multimedia*, 22(12):3196–3209, 2020. doi: https://doi.org/10.1109/tmm.2020.2972830.

[257] Jing Yu, Zihao Zhu, Yujing Wang, Weifeng Zhang, Yue Hu, and Jianlong Tan. Cross-modal knowledge reasoning for knowledge-based visual question answering. *Pattern Recognition*, 108:107563, 2020. doi: https://doi.org/10.1016/j.patcog.2020.107563.

[258] Reng Zeng, Xudong He, and W.M.P. van der Aalst. A method to mine workflows from provenance for assisting scientific workflow composition. In *2011 IEEE World Congress on Services*, pages 169–175, 2011. doi: https://doi.org/10.1109/services.2011.55.

[259] Chaoning Zhang, Chenshuang Zhang, Chenghao Li, Yu Qiao, Sheng Zheng, Sumit Kumar Dam, Mengchun Zhang, Jung Uk Kim, Seong Tae Kim, Jinwoo Choi, et al. One Small Step for Generative AI, One Giant Leap for AGI: A Complete Survey on ChatGPT in AIGC Era. *arXiv preprint arXiv:2304.06488*, 2023. doi: https://doi.org/10.48550/arXiv.2304.06488.

[260] Jiuling Zhang, Beixing Deng, and Xing Li. Concept based query expansion using wordnet. In *2009 International e-Conference on Advanced Science and Technology*, pages 52–55. IEEE, 2009. doi: https://doi.org/10.1109/ast.2009.24.

[261] Leishi Zhang, Andreas Stoffel, Michael Behrisch, Sebastian Mittelstadt, Tobias Schreck, René Pompl, Stefan Weber, Holger Last, and Daniel Keim. Visual analytics for the big data era—a comparative review of state-of-the-art commercial systems. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 173–182. IEEE, 2012. doi: https://doi.org/10.1109/vast.2012.6400554.

[262] Zhuoxun Zheng, Baifan Zhou, Dongzhuoran Zhou, Ahmet Soylu, and Evgeny Kharlamov. Exekg: Executable knowledge graph system for user-friendly data analytics. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 5064–5068, 2022. doi: https://doi.org/10.1145/3511808.3557195.

[263] Zhilan Zhou, Ximing Wen, Yue Wang, and David Gotz. Modeling and leveraging analytic focus during exploratory visual analysis. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. doi: https://doi.org/10.1145/3411764.3445674.

[264] Sujia Zhu, Guodao Sun, Qi Jiang, Meng Zha, and Ronghua Liang. A survey on automatic infographics and visualization recommendations. *Visual Informatics*, 4(3):24–40, 2020. doi: https://doi.org/10.1016/j.visinf.2020.07.002.