IMPROVING KNOWLEDGE GRAPH UNDERSTANDING WITH CONTEXTUAL VIEWS

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

by

ANTREA CHRISTOU
B.S., University of Leeds, England, 2018

2024
Wright State University

Wright State University
COLLEGE OF GRADUATE PROGRAMS AND HONORS STUDIES

April 16, 2024

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Antrea Christou ENTITLED Improving Knowledge Graph Understanding with Contextual Views BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

Dr. Cogan Shimizu, Ph.D
Thesis Director

Dr. Thomas Wischgoll, Ph.D.
Interim Chair, Department of Computer Science and Engineering

Committee on
Final Examination

Dr. Cogan Shimizu

Dr. Krishnaprasad Thirunarayan,Ph.D.

Dr. Hugh P Salehi, Ph.D.

Paula Bubulya, Ph.D.
Interim Dean, College of
Graduate Programs and Honors Studies

ABSTRACT

Christou, Antrea. M.S., Department of Computer Science and Engineering, Wright State University, 2024. *Improving Knowledge Graph Understanding with Contextual Views.*

Knowledge Graphs (KGs) leverage structured data (entities and their relationships) to create a richly interconnected world. However, to fully explore these intricate connections, sophisticated exploration tools are essential as manual exploration can become overwhelming. Applications for KG exploration span many use-cases: social network analysis, corporate intelligence, and medical research.

This research improves the InK Browser (Interactive Knowledge Browser), a modular, web-based tool for KG exploration, facilitated by flexible views. The goal is to enhance user understanding and this is tested through a user study.

Flexible views are made possible by applying complex constraint definitions against data instances. When data points (and their relations) match a data shape, the flexible view provides an adaptive prespective of that data. The InK Browser already provides a flexible view for geospatial data (a map) and metadata (semantic and type information), as well as search functionality. This research has added a new functionality in Flexible Views , a KG summarization that is utilized within the InK Browser by the dynamic creation of SPARQL queries made from shortcuts of the used schema. This functionality aids the challenge of navigation and comprehension of KGs.

Contents

List of Figures

Acknowledgment

I would like to take this opportunity to extend my thanks to my advisor Dr. Cogan Shimizu, for being an amazing mentor and taking a leap of faith with me, always pushing me out of my comfort zone in the best way. Also, I would like to thank Dr Krishnaprasad Thirunarayan and Dr Hugh P. Salehi, for accepting to be in my committee, I really appreciate it. In addition, I would like to express my gratitude towards Evan Music, for his incredible input regarding the primary development of the underlying platform and help throughout this process. To my amazing family and friends back home and to the new ones I made in the KASTLE Lab, I cannot thank you enough for always being there to lean on while navigating this challenging process. Last but not least, thank you to my amazing partner, for being there for every tear, frustration and emense support.

To my grandma Androulla, my grandpa Michael, my mum Kalia and dad Christos, my sister Kristia, and Evan.

Introduction

Structured data can be understood through the use of complex networks, which are made up of entities and their relationships with one another. Knowledge graphs (KG) have become essential tools for a variety of tasks, including helping to understand social networks, corporate intelligence, and medical research [10]. However, their inherent complexity calls for innovative research tools to strengthen a given dataset's properties in a less time complex manner. The Interactive Knowledge Browser (InK Browser) emerges as a component-based virtual environment to answer that need [16].

The main unique feature of the InK Browser is its ability to apply complicated constraint definitions against data objects to enable flexible views. Flexible views dynamically adjust data displays to provide a customized view of information that aligns with predefined data structures. Users may now personalize their exploration experience.

This work contributed to the creation of an additional Flexible View, utilizing a dynamically constructed query to extract information of the dataset in a summarized manner. This is made possible when shortcuts from the schema are being extracted and arranged in a way that summarize the datasets information related to a specified entity. The user has a choice, either in the form of interactive - sequential windows. The Schema window displays the nodes extracted from the dataset, by clicking on a given node then the Type window populates with the data related to that node, and by the user clicking on one of the "type" data, the Focus window gets populated with all the information attached to that

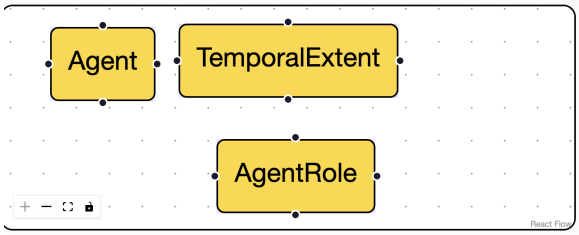


Figure 1.1: Nodes extracted from the datasets schema diagram.

specific instance.

Type
"Antrea"
"Brandon"
"Cogan"

Figure 1.2: Data related to the ”Agent” node if selected.

middle0Label	middle1Label	rhsLabel
GradStudent	2018-2021	CIIM
GradStudent	2021-present	WrightStateUniversity
Intern	June2018-December2018	BankofCyprus
UndergradStudent	2015-2018	UniversityofLeeds

Figure 1.3: Data related to the Type instance selection, in this case everything related to “Antrea” Agent.

Our Problem: The goal of the research is to improve the ability to navigate and understand structured material that is complex, especially KGs. The thesis focuses on statistically quantifying the impact of these flexible viewpoints through an upcoming user study, highlighting the goal of increasing navigation, discovery, and summarization in knowledge graph research. The ultimate goal is to showcase the significant advantages of the InK Browser as a driver for improved KG understanding and interpretation, offering a personalized and intuitive method for structured data exploration.

This research includes a user-study that evaluates the impact of flexible perspectives

quantitatively. InK Browser-based Linked Open Data exploration will be the primary focus of this project.Our objective is to showcase the advances in navigation, discovery, and summarization by incorporating flexible views and demonstrating the usefulness of the InK Browser for knowledge graph research [15].

Hypotheses :

1. The InK Browser's Flexible Views are expected to result in a quicker understanding of structured data resulting in successfully completing TaskA and TaskB.

2. The InK Browser's Flexible Views are expected to result in a more in-depth understanding of structured data resulting in successfully completing TaskA and TaskB.

In other words, flexible perspectives give personalization, which is a crucial aspect of user driven exploration. By providing users with more capability, we intend to show the considerable benefit of the InK Browser as a catalyst for improved KG comprehension and interpretation. Together, improved navigation, summarization, and discovery offer a thorough approach to make KG exploration efficient, user-friendly, and customized to meet particular needs.

1.1 Chapter Overview

Each Chapter is as follows.

Chapter 2 : Preliminaries contain background information that the reader needs go through before continuing reading this thesis.

Chapter 3 Related Work contains the literature review made in order to gain an understanding on how to solve the problem stated.

Chapter 4 Methodology states the way that the research contribution to the problem achieved.

Chapter 5 Evaluation states the Tool evaluation process.

Chapter 6 Results contains details on how the evaluation results were collected, cleaned, processed and interpreted.

Chapter 7 Conclusion discussion as to what the results mean in terms of the problem and hypotheses statements.Concludes the thesis by declaring any setbacks or challenges this research accumulated while also stating what can be build upon the tool and evaluation process.

Preliminaries

A thorough understanding of core principles is essential when exploring organized data and knowledge representation. Extensive information about Knowledge Graphs (KGs) & Schema Diagrams -2.1, what is an Ontology -2.2, Resource Description Framework (RDF) & Resource Description Framework Schema -2.3, Protégé & Ontology Design Pattern Language Annotator (OPLa) -2.4, Ontology Design Patterns (ODPs) -2.5, Modules -2.6, Triplestores & SPARQL -2.7, Interactive Knowledge Browser (InK Browser) -2.8, is provided in this section along with reasoning's as to why are these relevant to the research.

2.1 KGs & Schema Diagrams

KGs are organized databases that use entities, relationships, and characteristics to represent knowledge. Entities are things or ideas, and relationships are the links between these things. KGs utilize a network structure, where entities (representing real-world objects or concepts) are connected by edges (representing relationships between those entities). These connections, visualized as nodes and edges, allow KGs to capture intricate relationships within data such as product information, workplace database systems etc. This research takes the graph essence of the KG of a public materialized dataset, and by traversing through every node, we manage to extract shortcuts that we use to create the flexible views application for the InK Browser[17].

Databases relationships, and restrictions are all visually represented in a schema di-

agram. Schema diagrams demonstrate the ontology structure in connection to RDF and KGs; they represent the links between entities, the attributes that are attached to each entity, and the entities themselves. This graphic illustration facilitates comprehension of the KG's general architecture [11].

In the following schema example, the notion of a "prefix" is introduced as well. KGs use the namespace notion to give entities and relationships context and clarity. These namespaces serve as formal representations of a certain domain. Prefixes act as abbrivations of a certain domain or namespace. In this example "xsd" stands for XML Schema Definition, a language that defines structure of data in Extensible Markup Language (XML), that is used for documents that can be readable both from humans and machines. In this case, **"xsd:String"** stands as an instance of an **"xsd:String"** datatype, which can contain any characters including letters, numbers and symbols, it's colour and shape is also standardized for ease of reusability. **"PlaneID"** is connected to that instance for that very reason, it is a string [17].

Figure 2.1: Plane ID part of a larger schema for a Plane Ontology representing information regarding the PlaneID associated with a Plane as a string [5].

2.2 What is an Ontology

KGs are an important tool for integrating data, which is usually heterogeneous, in a formalized manner. This formalization, which constrains how data can or cannot be related, is called an ontology, and it acts as a schema for a knowledge graph. This guarantees a uniformity along the KG by using a set of "rules" called axioms. This research does not

particularly use axioms in an extent, but they form the foundation of the KG. Given the above example, a "PlaneID" is tied to only one instance of a plane.

Therefore that can be denoted with the following axiom : **owl:Plane** $\sqsubseteq \exists R.$**max 1 PlaneID**

And in natural language form : **hasPlaneID exactly 1 PlaneID** [25].

2.3 RDF & RDFS

The Resource Description Framework (RDF) is key to creating semantic links in knowledge graphs (KGs). RDF serves as a standard language for data representation on the web, offering a solid framework for information organization, eventually enhancing the comprehension of online content. The essential KG building block, known as the triple, is used to achieve this. A triple consists of three components: subject, predicate, and object, and each triple conveys a statement about two particular instances within a domain along with their relationship.For example , to add on the above Plane instance KG, the subject is the **"Plane"** node, the predicate is the **"hasPlaneID"** edge and the object is the **"PlaneID"** node. This can be translated to natural language like this: **"This plane has plane ID of plane Id number"**, giving information regarding the unique ID of a specific plane.

With the introduction of ideas like classes, properties, and subClassOf relationships, Resource Description Framework (RDFS) offers an RDF schema. It permits basic inferencing and the definition of hierarchies. Consider building with LEGO. RDF gives the fundamental building pieces, or bricks, and can be linked together in any way that makes sense. Similar to a set of instructions, RDFS describes cars, wheels, and their connections so you may use those LEGO to build certain structures.

Both the framework and schema act as crucial components when extracting the dynamic shortcuts for the application of the flexible views as they set the skeleton of the dataset and it's properties this study is using [12].

2.4 Protégé & OPLa annotator

Working with knowledge graphs is made easier with Protégé, an open-source, free program. It serves as a knowledge management system in addition to an ontology editor. Protege functions as a behind-the-scenes architect in the context of knowledge graphs. This tool can be used to define and construct relationships between the concepts (such as “people”, “places”, or “events”) that are contained in the knowledge graph. serving as a guide for the KG, outlining the types of data that are present and how they relate to one another [19].

One standalone plugin for Protégé is the OPLa Annotator. opla:isNativeTo (where “opla” acts as the prefix for the Ontology Design Pattern Language Annotator namespace) annotations on ontological entities in an Web Ontology Language (OWL) file can be created guided by using this plug-in. This feature adds important context to the knowledge representation by enabling users to link ontological entities to their original domain or place of origin. Though CoModIDE includes this specific functionality, it can be a speedier solution when the imposed graphical structure is not wanted or necessary, as it does not require the construction of modules or the graphical canvas [22, 20, 14].

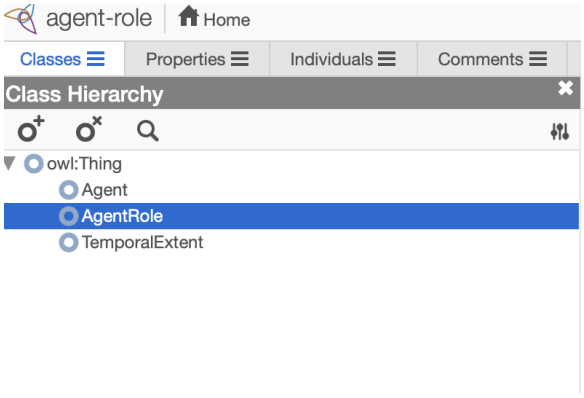


Figure 2.2: Protégé example with classes.

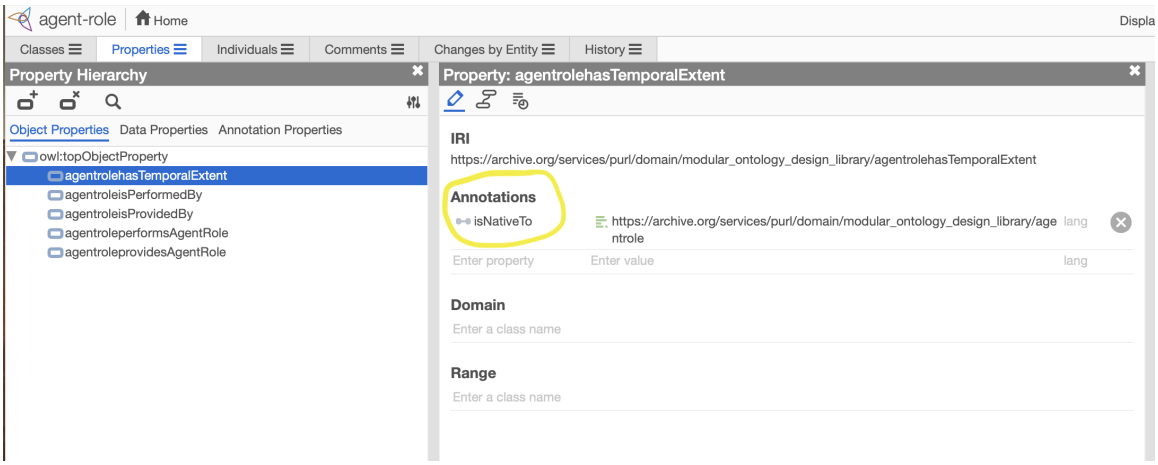


Figure 2.3: Protégé example with properties along with the OPLa annotator property.

2.5 ODPs

Provide reusable fixes for typical issues when it comes to reusability and consistency within ontology building. There are online repositories, however there are wide differences in the documentation and quality. In response, others suggest building machine-readable patterns using languages like the Ontology Pattern Language (OPLa) and curating libraries of thoroughly documented ODPs. The plan is to use an ODP as a template rather than implement it directly. This makes modification possible while guaranteeing that the final product retains the essential framework. This method encourages efficiency and uniformity in the ontology building process [13, 7].

2.6 Modules

Modules of ontologies serve as reusable components of knowledge hierarchies. A central idea and its connections to other ideas are captured in each module. The emphasis is on encapsulating related functionalities, even though shared semantics, ODPs, are frequently the foundation. The development context and domain semantics both influence the particular classes and attributes that make up a module. The way modules are defined is a reflection of the inherent ambiguity of the things they stand for. Nonetheless, modules are officially

designated using OPLa within the ontology data for clarity's sake. This makes it possible to thoroughly document every module, which encourages effective development and better comprehension of intricate knowledge systems[24].

2.7 Triplestores & SPARQL

Triplestores are databases specifically made to handle and store data in "triple" format. The relevant linking of data between triple stores made possible by this structure facilitates relationship analysis and information retrieval. A variety of Triplestore tools and resources are described under the W3C category [2].

Apache Jena-Fuseki

Instead of focusing solely on characters and symbols, this triplestore offers methods for working with data that reflect real-world meaning. Jena features a query language called SPARQL for retrieving and altering data, and it enables working with Semantic Web-specific data formats like RDF and OWL [1].

SPARQL

Information extraction from KGs heavily relies on the query language and protocol SPARQL, which is used to query RDF data. Users can examine relationships, access particular data points, and look for trends in the graph using SPARQL queries. Its RDF integration fits in perfectly with Knowledge Graph architecture[8].

This research is using Apache Jena-Fuseki as a triplestore to use as an endpoint to retrive the data and display within the InK Browser.

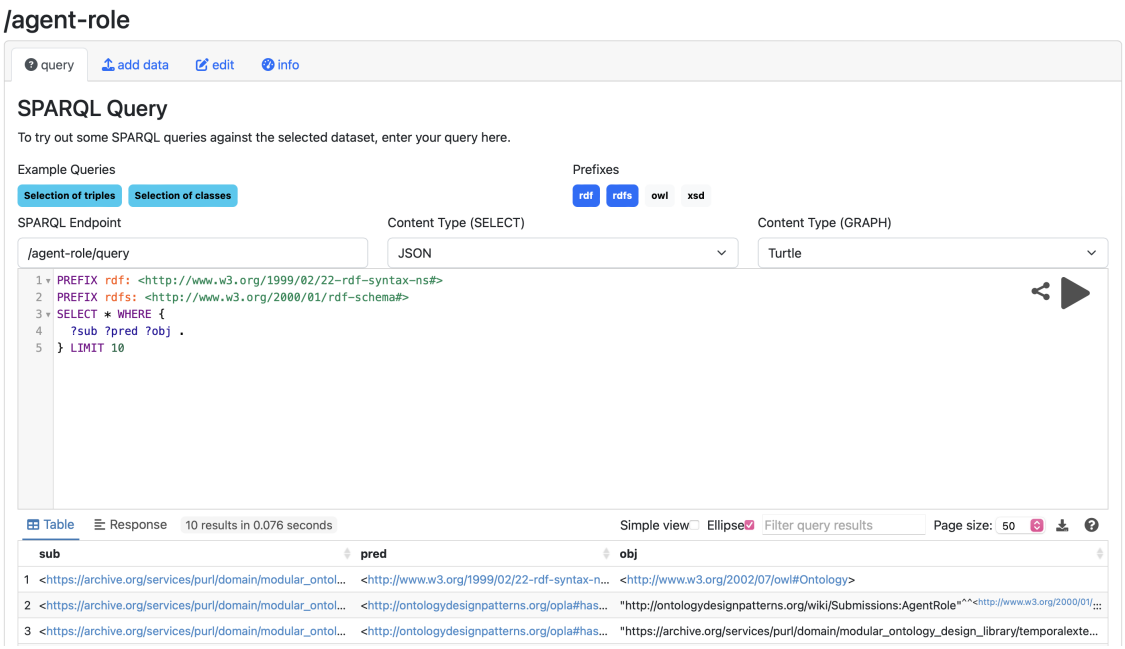


Figure 2.4: A screenshot of the triplestore Apache Jena-Fuseki hosting the AgentRole pattern materialized KG, retreiving the first 10 triples (subject, predicate, object).

2.8 InK Browser

The InK Browser depends on a React application (React App) and Apache Jena Fuseki running concurrently. The user interface, or React App, is made up of multiple JavaScript (.js) files. Each of these files reflects a distinct, modifiable view or component that enhances the overall presentation inside the application window and also represent the Flexible Views that are to be selected by each user when they use the tool.

These files communicate with a SPARQL endpoint in Apache Jena Fuseki in order to obtain pertinent information from the materialized KG that is uploaded and stays persistent within the triplestore. Therefore, the data do not need to be hosted within the React App for the Flexible Views to display the customized material.

While Jena Fuseki is operating at localhost:3030, the React App, available at localhost:3000, can execute SPARQL queries to obtain the required data from the KG, allowing users to interactively explore the dataset [3].

Related Work

This research thoroughly studied a number of articles to have a deeper grasp of knowledge graph analysis and navigation. Each source helped to shape the overall study approach by offering insightful information and useful techniques.

3.1 KG Navigation

The problem of navigating complex knowledge graphs was addressed in "Leveraging Schema Information for Improved Knowledge Graph Navigation". The suggested method used attributes and relationships along with schema information to lead users around the graph. The study acknowledged difficulties linked to schema availability while discussing prospective applications such as recommendation systems and search engines, in addition to presenting a promising method for knowledge graph navigation.

The creation of a schema-aware navigation algorithm forms the basis of the suggested methodology. The quickest route between two points is just given priority in typical shortest path algorithms, but this technique does more. As an alternative, it uses schema information to direct users down paths that make sense semantically and fit the KG's natural structure.

First, based on how closely the users follow the schema, the algorithm rates several possible navigation routes. Routes adhering to established connections inside the schema, such as a professor-course"teaches" relationship, would score higher. In the knowledge

domain, this directs users toward pathways that have semantic value rather than away from connections that are meaningless. Schema information is used as heuristics in the second strategy to affect the search. Depending on the schema specifications, these heuristics may prioritize entities belonging to the same class or prevent pathways from going beyond predetermined schema requirements.

Through user studies, this approach gets evaluated by comparing user performance while using the schema navigation approach versus a traditional shortest path algorithm having positive results.

Further exploration statements lead this research into developing a shortcut extraction method from a given schema resulting in a SPARQL query creation that later gets utilized in adding more Flexible Views of the InK Browser, essentially giving the customized visualization of a KG [9].

3.2 InK Browser – The Interactive Knowledge Browser

The web-based tool InK Browser, which enables interactive knowledge graph exploration, was introduced in ”InK Browser – The Interactive Knowledge Browser”. Users could examine clusters and graph structures thanks to the browser's functionality for filtering, sorting, and searching. Case studies showed how useful it is in scientific papers and genetic databases. The paper covered the architectural and technical difficulties and gave a thorough rundown of what InK Browser is capable of. The conclusion emphasized its potential applications in a variety of domains, including as social network analysis and biomedical research [16].

In contrast to conventional browsers that concentrate on discrete data points, InK Browser offers a comprehensive perspective that transforms the exploration of knowledge graphs. A schema diagram, a visual map that reveals the graph's underlying structure, helps it accomplish this. Using this roadmap, users may browse by clicking on topics to

learn more and see how different parts fit together to form a larger image. With features like entity annotation (pop-up details), class hierarchy (concept linkages), and text search, InK Browser further enhances comprehension and turns knowledge graph research into an exciting voyage of discovery.

As mentioned above, these features the InK Browser offers, are what we also call Flexible views, or Flexible perspectives. These unique features, let users personalize their exploring journey. Flexible views enable dynamic adjustment to deliver individualized perspectives by utilizing intricate constraint definitions, which improves user understanding of the KG. Such views can be a map, a table containing summarized information, a table with statistical information, an interactive schema and so on. Depending on the dataset, the flexible views can be adjusted in a way that makes sense for it. With shortcut extraction from a publicly materialized dataset that will be subsequently detailed in the Methodology section, new flexible views are generated, giving the user the chance to click on nodes they are interested in, giving them any relevant information associated with them.

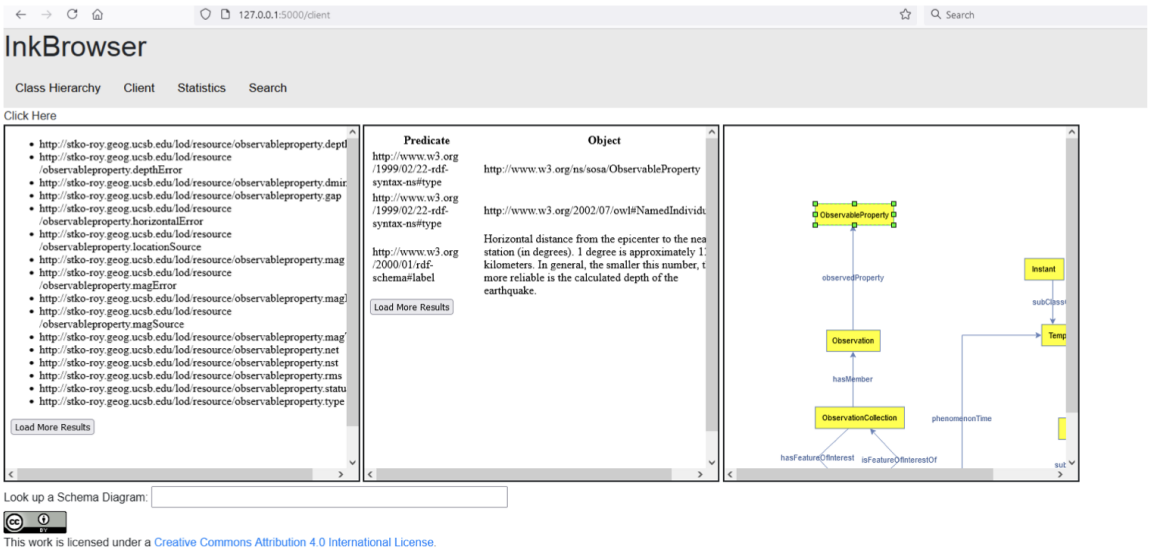


Figure 3.1: An early version of the InK Browser from [16] (used with permission).

3.3 Interesting Paths in KGs

The paper ”WiSP: Weighted Shortest Paths for RDF Graphs” presented WiSP [26] , a technique for calculating weighted shortest pathways in RDF graphs. When tested on real-world datasets, WiSP performed comparably in terms of efficiency and accuracy. In addition to highlighting the algorithm's benefits for knowledge graph exploration, query optimization, and semantic similarity computing, the paper also described the algorithm's implementation and possible uses.

This thesis investigated further optimal graph navigation techniques in order to finally conclude in using the Breadth First Search (BFS) alsogithm for the shortcut retrieval. When navigating graphs in an organized fashion, BFS works effectively because it makes sure that every node within a certain radius of the source node gets investigated before moving on to nodes farther away.

3.4 SPARQL compatability with web tools

”Phuzzy.link: A SPARQL-powered Client-Sided Extensible Semantic Web Browser” introduced Phuzzy.link, a client-side Semantic Web browser. Phuzzy.link provided a user-friendly interface with cutting-edge web technology by using SPARQL for querying and displaying connected data. Predefined query templates, flexibility for tweaks, and direct SPARQL execution were among the features. According to the study, Phuzzy.link offers a strong and adaptable tool for handling linked data, making it easier to conduct creative data exploration and analysis. As a community-driven project that extracts structured data from Wikipedia, DBpedia's notes described its purpose. Natural language processing and machine learning are just two of the uses for DBpedia's , which is kept in a publicly available database. Connected with additional datasets as part of the Linked Open Data effort, DBpedia allows complex applications utilizing numerous sources to be created.[6]

Although direct SPARQL execution is provided by Phuzzy.link, users who are not familiar with the query language may find this to be a hurdle. Hence the choice of the InK Browser for the utilization of the Flexible Views since the React App can effectively use the triplestore Apache Jena-Fuseki as an endpoint to retrive inquired data. [6].

3.5 Modular ontology modeling

Ontologies simply digital representations of knowledge, and reusing them might be difficult. Several issues contribute to this challenge: ontologies and their intended applications frequently lack alignment in terms of detail; reusable ontologies often lack clear definitions; and there is an absence of tools to promote and facilitate reuse during the development process. The Modular Ontology Modeling (MOMo) approach and its companion program, CoModIDE [20], were developed by researchers in order to overcome these problems. By using a lot of visual representations to assist gather information from experts, MOMo improves on current design techniques while introducing a critical component. By encouraging the reuse of current knowledge for new applications, this strategy seeks to increase accessibility to ontology development [21].

Moreover, the research project's goals were precisely aligned with MOMo's emphasis on modularity and reusability. The AgentRole pattern, which formed the basis of the ontology being developed, had already been conceptualized and therefore became seamlessly integrated with real-world data to create a comprehensive ontology. This secured the practical applicability of the ontology used.

Methodology

When KG Flexible Views are dynamically created from module metadata, KG visual representations based on details about modules are automatically created. This aids in organizing and comprehending the modules' relationships with one another. We can map extracted information to KG nodes and connections by extracting features such as module names (e.g., "**AgentRole**"), descriptions (e.g., "defines roles played by agents"), characteristics (e.g., "includes properties for **hasRole, providedBy, hasTemporalExtent**"), and relationships to other modules (e.g., "related to **Person** module"). This makes it simpler to deal with and manage complicated systems because it makes it possible to quickly identify and analyze the links and organizational structure between various modules. For instance, a KG node might represent the "**AgentRole**" module, with connections to other nodes representing related modules like ""**Person**" (entities that can have roles) or "**Task**" (activities for which roles are assigned). These connections would be based on the information extracted from the module metadata, such as the "**hasRole**" property linking "**AgentRole**" to "**Person**".

A case study was performed using the InK Browser with flexible views, using a publicly available knowledge graph dataset. We have evaluated the effectiveness of the InK Browser and flexible views using metrics such as time to complete tasks and accuracy of results.

Flexible views are a feature of the InK Browser that allow users to customize the display of the knowledge graph to meet their specific needs. With flexible views, users

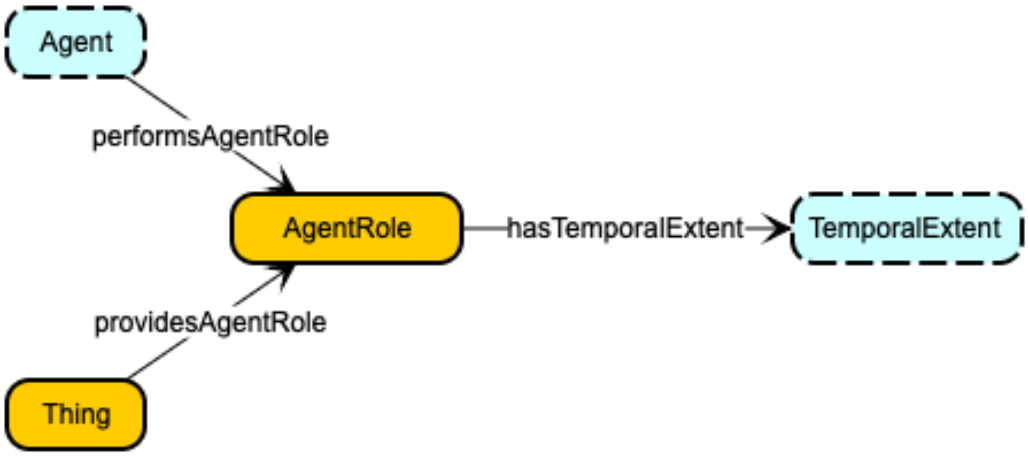


Figure 4.1: AgentRole Pattern Schema [23].

can modify the graph's layout, labeling, and coloring to highlight important information, improve readability, and better understand complex relationships between entities.

For example, users can choose to display only certain types of relationships between entities, filter entities based on their attributes, or arrange the graph in a hierarchical or clustered layout.

4.1 Shortcut Generation & Shortcut Extraction

4.1.1 Keywords

Agent: The person that performs a role.

AgentRole: The role being perfomed by an Agent.

Provider: The organization that provides the Role.

Temporal Extent: The time period that the Agent is/was performing the AgentRole. The AgentRole Pattern Shortcut Extraction which the role is being provided by the provider/role is being performed by the Agent.

4.1.2 Objective

Extracting/Deriving shortcuts from information we already have in our graph.

This process eventually creates a shortcut between multiple attributes of the agent-role pattern and merges them together to provide full information on each part of the graph.

4.1.3 Shortcut Generation and implementation code steps

The shortcut extraction is done using the GrampML file of the AgentRole schema diagram.

A dictionary namespace is defined to map XML namespace prefixes to their corresponding URLs.

The GraphML file named agent-role.graphml is parsed using xml.etree.ElementTree library, and its root element is obtained.

The script searches for an XML node labeled as "**Agent**" inside the GraphML file using XPath expressions with the defined namespace. The agent-node variable is assigned the "**Agent**" node if found. This is because our pattern is specifically dependent on Agent and its relations.

A breadth-first search (BFS) is performed on the graph starting from the "**Agent**" node. Following that, it investigates every entity that is directly connected to that node and it visits every one of these neighbors before moving on to investigate the neighbors of those neighbors, and so forth, layer by layer, revealing the intricate network of data inside the KG.

Then the paths are being printed with outgoing edges and paths with incoming edges to the console.

Then the shortcut string is being added to the agent-role RDF Triple Language (ttl) file (a file format used to express RDF data) along with the other materialized information. In this case the data consists of three Agents: Antrea, Brandon and Cogan, along with their respective roles and providers or those roles.

```
Agent Node ID: n0
Paths with outgoing edges:
1. performsAgentRole
2. performsAgentRole -> hasTemporalExtent

Paths with incoming edges:
1. performsAgentRole -> providesAgentRole
```

Figure 4.2: AgentRole schema's paths extracted using the BFS algorithm.

Note that the shortcut can be modified, the orders of the elements can change and strings can change as well. It depends on the path extracted from the graphml file and what makes sense to the given pattern.

4.1.4 Output

The code outputs the agent-role ttl file that is stored under "data" in the github repo that will be attached below, and it outputs a new ttl file with the materialized shortcut extraction.

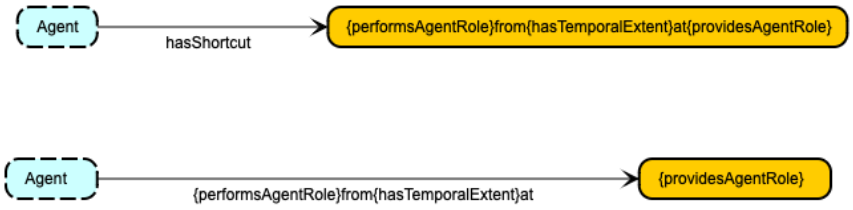


Figure 4.3: Agent-Role Shortcut Schema with generated string template.

4.2 Shortcut Materiliazation - Query Generation

- The code continues with parsing the agent-role turtle (TTl) file with respect to the **hasShortcut** predicate, then it matches that object aka the shortcut string, with the predicates of the agent-role TTl file.

- If the elements match, then based on each person and their roles, the elements of the shortcut string are being replaced with the objects of the predicates that were matched with, producing shortuts of the roles of each person.

- The script explores namespaces as it initializes an RDF graph, which makes complex web addresses easier to understand and makes the code easier to comprehend. Deeper exploration can be undertaken with the foundation of this namespace resolution, which supports the coherent representation of RDF data.

- The script reveals shortcuts; condensed versions of intricate relationships, through methodical exploration. Regular expressions are used to decipher the underlying pathways and types that are encoded within these shortcuts.

- The script dynamically creates SPARQL queries, which are the preferred query language for RDF data. By serving as a link between the script and the RDF graph, these queries make it possible to retrieve important information about resource relationships.

Appendix A contains the described query.

Evaluation

We have evaluated the tool by constructing a qualitative and quantitative analysis. Qualitative will be constructed by giving out surveys and interviewing the people that will be part of our case study for some feedback regarding the flexible views and what they thought on the InK Browser. The feedback will consist of their likes/dislikes, suggestions, their perception of usability and their overall impression. Quantitative analysis will be constructed based on the time-frame, accuracy and ease of use. After gathering that data we can construct a statistical analysis among the participants and compare the difference between use of flexible views and without flexible views.

5.1 Expected Results

We expect to find that the InK Browser is a useful tool for exploring knowledge graphs, and that the flexible view feature enhances its usability by allowing users to customize the display of the graph to meet their needs. We anticipate that the case study will demonstrate the effectiveness of the InK Browser with flexible views, and that users will find it to be a valuable tool for knowledge graph exploration.

5.2 Survey - Experiment Protocol

In this experiment, we evaluate how well the modular web-based KG exploration tool, the InK Browser, aids in a more rapid and in-depth comprehension of structured data. We review the experiences of participants that will complete two set of tasks wither using or not using a tool: TaskA, which only uses the ttl file format, and TaskB, which has access to the sophisticated features of the InK Browser. Specifically, we predict that participants completing Tasks A and B using the too will perform better on comprehension and navigation of the competency questions than only completing both tasks without any tool.

5.2.1 Hypotheses

Hypotheses :

1. The InK Browser's Flexible Views are expected to result in a quicker understanding of structured data resulting in successfully completing TaskA and TaskB.

2. The InK Browser's Flexible Views are expected to result in a more in-depth understanding of structured data resulting in successfully completing TaskA and TaskB.

Recruitment included participants from the Department of Computer Science and Engineering. Backrounds included people that work in Computer Science but are not really familiar with KGs, people that work in Computer Science that do. People from the Graduate School that are not in Computer Science. People that know how to navigate basic web tools like Google Chrome and any other web-based tool that includes the navigation while using the "mouse" and it's buttons to click where they like. The number of 14 people can be sufficient and since they will be completing the tasks twice (with and without the tool),

the statistical number of participants will result in 28. The experiment took place in Wright State University, Joshi Room 392. All eight members of the KASTLE Lab participated, along with another 6 that were recruited through an email sent to the Computer Science Department directory of students.

TaskA: Given a simple Knowledge Graph (KG) with relationships between items **A, B, C**, how does **A** relate to **B**?

TaskB: Given the same KG, how are items **A** and **B** connected through **C** ? These tasks will be performed twice, while using a tool and without it. In this study the tool is the InK Browser and it's flexible views navigation. No tool in this case is answering the questions by simply looking at the triples of the KG.

The order matters and has an effect on the outcome on wherether the participant uses the tool first or second but it is not important. The data collected consists of the answers given by the participant with respect to the time the participant took answering them. Participants used a PC to look at the KG's data when they are completing their task without a tool and they had to interact with the PC trough the cursor when they were completing the task with a tool. They were be given a form with simple questions related to the KG with space to answer them in both instances, after using the tool and after they just complete the task without it.

Example competency questions to be asked :

1. When was Antrea an Undergrad Student ?

2. Where was Cogan a Graduate Student ?

3. How many years was Antrea's Internship ?

4. How many roles has Antrea had ?

The study did not take more than an hours time for each individual. It took approximately a week to complete all fourteen surveys.

The study did not interfere in any way with the participants rights or wellfare. There was no risk associated with the questions to be asked, it is simply a questionaire to be

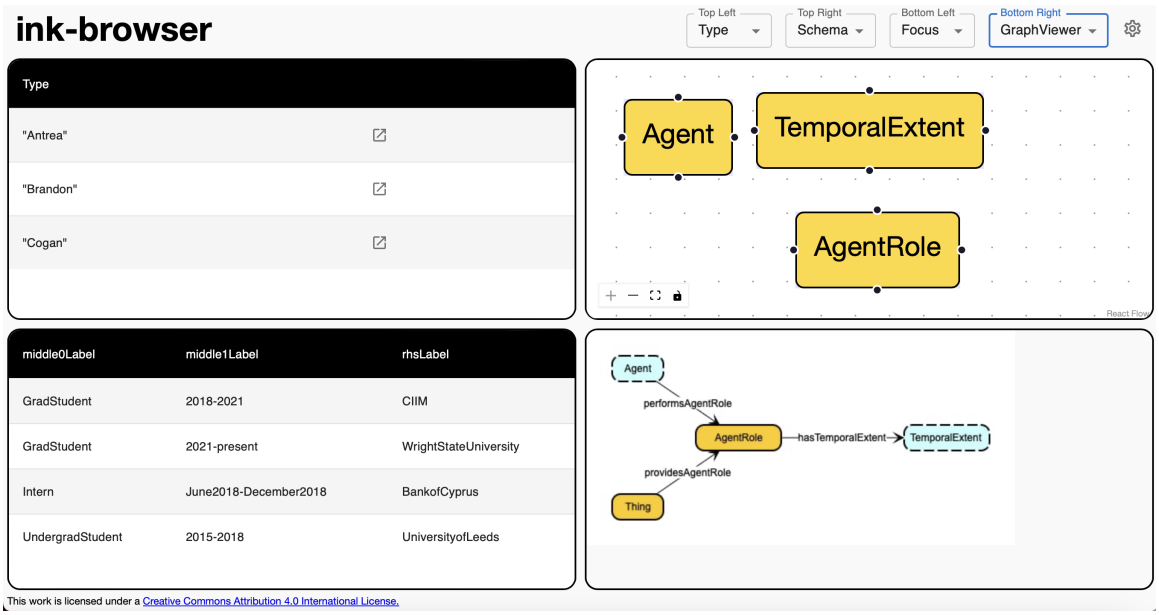


Figure 5.1: InK Browser updated Flexible Views

answered by them like they would answer in a class oriented test/quiz. A personal laptop was sufficient for the participants to complete their tasks assigned. The Kastle Lab is provided with a space within Wright State University in Joshi Research Center Room 391. It is equiped with chairs, tables, desks, monitors, paper and printing devices. There was access to eight potential participants and all of them were recruited. There were verbal instructions read to the participants before their session starts following with them going through it themselves. For the recruitment via email a description of the experiment was included along with any details the participant needs. The personal laptop had Apache Jena-Fuseki running on localhost:3030 while the Agent-Role materialized dataset was loaded. After that, the InK Browser started running on localhost:3000, displaying the different windows.

Results

In an effort to assess the usefulness of InK Browser, 14 individuals were given two tasks (**TaskA** and **TaskB**) to complete in two iterations, once while not using the tool and once while using it, with measurements made of their accuracy and time. A pre-defined rubric later on that outlines exactly what an accurate response looks like, will be attached below.

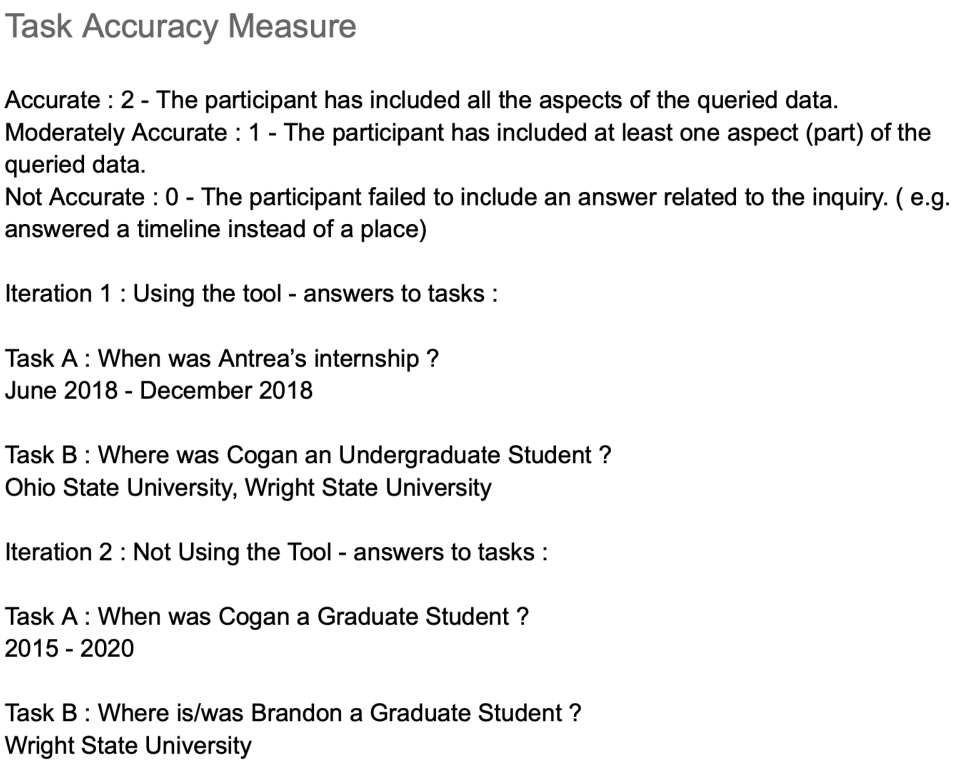
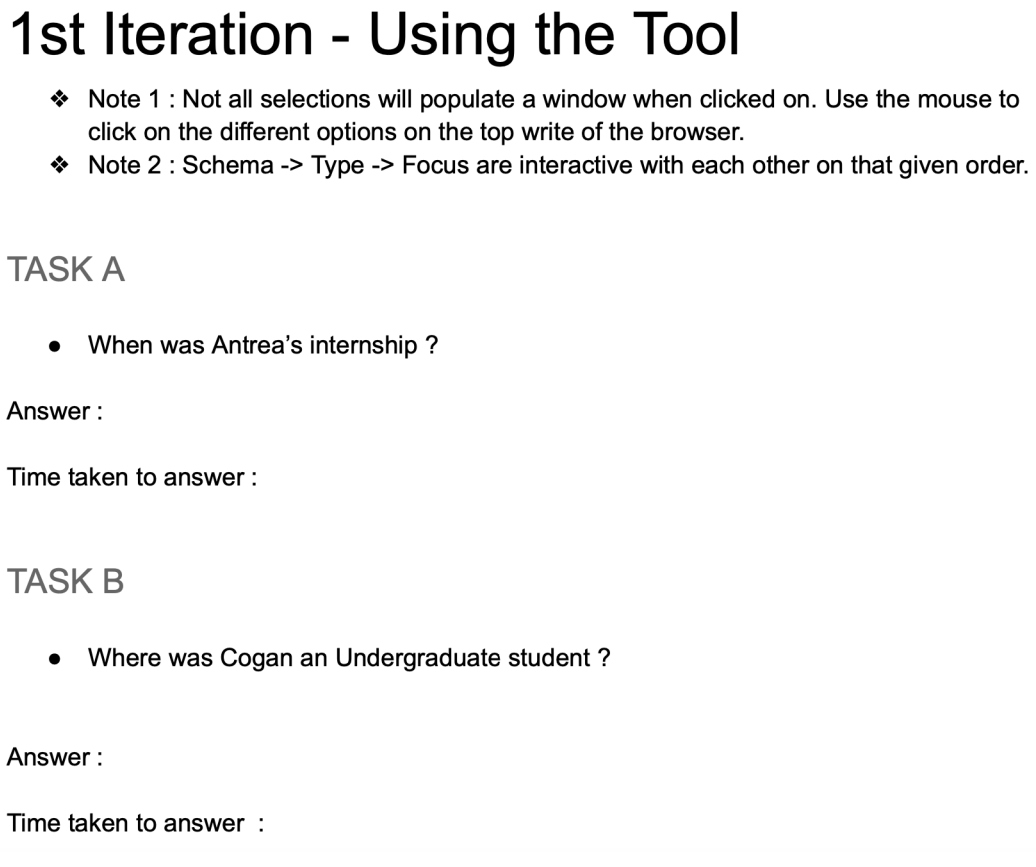


Figure 6.1: Accuracy Rubric

6.1 Data Collection

The data was selected using google sheets where the participants of the survey were using to type their answers to TaskA and TaskB in two iterations, either while using the tool in the first iteration and looking at raw data for the second iteration and vice versa.

In between task completion, using a stopwatch, the PI measured the time taken to complete **TaskA** and **TaskB** in both iterations.


1st Iteration - Using the Tool

- Note 1 : Not all selections will populate a window when clicked on. Use the mouse to click on the different options on the top write of the browser.
- Note 2 : Schema -> Type -> Focus are interactive with each other on that given order.

TASK A

- When was Antrea's internship ?

Answer :

Time taken to answer :

TASK B

- Where was Cogan an Undergraduate student ?

Answer :

Time taken to answer :

Figure 6.2: Survey - Iteration 1 - Using the Tool

2nd Iteration - Not using the Tool

- ❖ Note : Do not interact with the file. Only scroll if needed.

TASK A

- When was Cogan a Graduate student ? (aka every role after completing an Undergraduate role)

Answer :

Time taken to answer:

TASK B

- Where is/ was Brandon a Graduate Student ?

Answer :

Time taken to answer :

Figure 6.3: Survey - Iteration 2 - Not Using the Tool

1st Iteration - Not using the Tool

- ❖ Note : Do not interact with the file, just scroll if needed.

TASK A

- When was Antrea's internship ?

Answer :

Time taken to answer :

TASK B

- Where was Cogan an Undergraduate student ?

Answer :
Time taken to answer :

Figure 6.4: Survey - Iteration 1 - Not Using the Tool

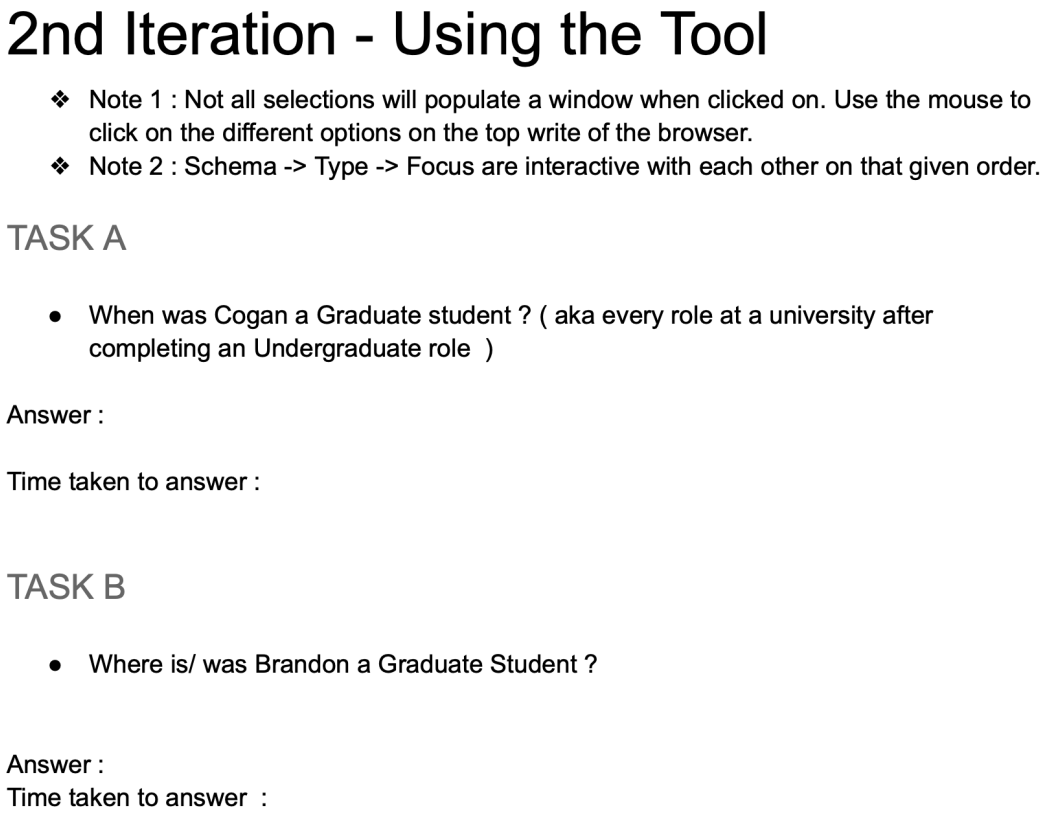


Figure 6.5: Survey - Iteration 2 - Using the Tool

6.2 Data Cleanup

The Principal Investigator (PI) made sure the data had the following variables by loading it into a comma-separated values (CSV) file in accordance with the specified rubric.

Variables

Participant: The number of the participant followed by the task completed by him (a or b).

Time Tool: Time to complete task (indicated by the letter next to the number of the participant) while using the tool.

Accuracy Tool: Accuracy of the answer given by the participant of the given task (indicated by the letter next to the number of the participant) while using the tool.

Time No Tool: Time to complete task (indicated by the letter next to the number of the participant) without using the tool.

Accuracy No Tool: Accuracy of the answer given by the participant of the given task (indicated by the letter next to the number of the participant) without using the tool.

Participant	Time_Tool	Accuracy_Tool	Time_NoTool	Accuracy_NoTool
1a	1:13	2	1:17	2
1b	1:39	2	0:58	2
2a	2:48	2	0:41	0
2b	0:20	0	0:18	2
3a	2:07	1	1:06	2
3b	0:23	2	0:28	0
4a	0:55	2	0:41	1
4b	0:41	1	0:15	0
5a	3:31	2	1:58	2
5b	0:17	0	0:06	0

Figure 6.6: Top 5 Data Entries from the CSV file.

6.3 Data Analysis

Using Python, the CSV file hosting the clean data was loaded. With using Python libraries such as **spicy.stats**, **pandas**, **numpy**,**seaborn**, and **matplotlib**, the PI first loaded the csv into the program. Then, the time related variables had to be converted from strings into minutes for ease of analysis. Next, basic statistical information was extracted using the

	Participant	Accuracy_Tool	Accuracy_NoTool	Time_Tool_Minutes	Time_NoTool_Minutes
count	28.0	28.0	28.0	28.0	28.0
mean	7.5	1.4285714285714300	1.4285714285714300	1.1928571428571400	0.9607142857142860
std	4.1051007115358100	0.7901510136288030	0.7901510136288030	0.8609190774099060	0.6593926355777740
min	1.0	0.0	0.0	0.11666666666666700	0.1
25%	4.0	1.0	1.0	0.6500000000000000	0.5041666666666670
50%	7.5	2.0	2.0	0.925	0.7833333333333330
75%	11.0	2.0	2.0	1.475	1.2083333333333300
max	14.0	2.0	2.0	3.5166666666666700	2.7333333333333300

Figure 6.7: Data Collected Statistics Table

"describe" function, and that information was tabulated in a table.

Also, the distribution of Time Taken by Accuracy Levels was plotted against each other.

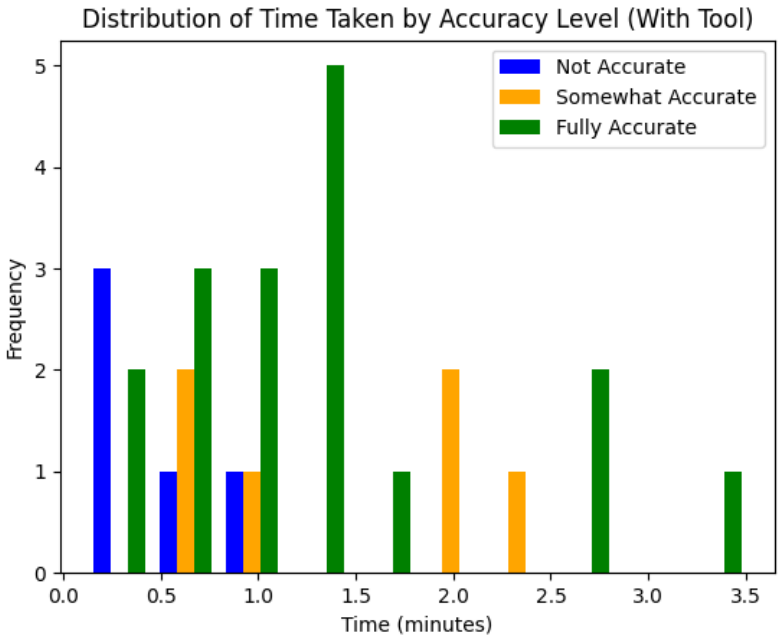


Figure 6.8: Time VS Accuracy Level

6.3.1 Student T-Test

Following up, a Student T-Test was conducted for the Time_Tool and Time_NoTool variables. A two sample Student T-Test, aims to examine if two variables are statistically different, in our case we want to examine whether the time taken to complete a task while using the tool is different than the time taken to complete the same task without the tool. And respectively examine whether the task accuracy while using the tool is statistically different than the task accuracy with the tool.

The standard deviation, which measures the distribution of data within each group, is used to determine whether the observed mean difference is the result of pure chance or a real influence.

This test was set up while considering we have two independent groups, in this case Time_Tool variable versus the Time_NoTool variable [4].

Null Hypothesis H0 : The means of the two variables are the same.

Alternative Hypothesis H1 : The means of the two variables are not equal.

Subsequently, the Sample Statistics were calculated using the following equations :

Let *X1...Xn* represent the Time with Tool Usage variables and let *Y1...Yn* represent the Time with no Tool Usage variables.

Average and variance of each variable :

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{6.1}$$

$$s_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \overline{Y})^2 \tag{6.2}$$

$$\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \tag{6.3}$$

$$s_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \overline{Y})^2 \tag{6.4}$$

Next step is calculating the pooled variance that will take both variables variance in order to finally calculate the test statistic that will determine whether we accept or reject the null hypothesis.

$$s_p^2 = \frac{(n_1-1)s_X^2 + (n_2-1)s_Y^2}{n_1+n_2-2} \tag{6.5}$$

$$t = \frac{\overline{X_1} - \overline{Y_2}}{\sqrt{s_p^2\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}} \tag{6.6}$$

6.3.2 T-Test Results for Time_Tool and Time_NoTool pair

T-statistic t= 1.52.

P-value p: 0.14 So, there is no significant difference between the time taken with and without the tool.

Similarly, a Student T-Test was conducted for the Accuracy_Tool and Accuracy_NoTool variables using the same equations as above.

6.3.3 T-Test Results for Accuracy_Tool and Accuracy_NoTool pair

T-statistic: 0.0 .

P-value: 1.0 . So, there is is no significant difference between the accuracy with and without the tool.

6.3.4 Wilcoxon Test

In addition, the Wilcoxon test was applied on the Time and Accuracy related variables respectively.

A non-parametric statistical test called the Wilcoxon test for paired data is used to compare two sets of paired data. The Wilcoxon test concentrates on the rankings of the differences between matched samples, in contrast to other tests that depend on presumptions about data normality. Because of this, it is a reliable choice in cases of skewed or non-normal data distributions.

The Wilcoxon Test assists in determining if the tool usage produced a statistically significant change within the group by comparing the ranks of the differences between pre- and post-intervention measures [18].

Since this and the Student T-Test were not expensive to use as tests, we used both to examine the effect of the tool in time and accuracy.

Null Hypothesis H0 : The median difference between the pairs of variables is zero.

Alternative Hypothesis H1 : The median difference between the two paired variables is not zero.

$$d_i = X_i - Y_i \quad \text{for } i = 1, 2, \ldots, n \tag{6.7}$$

Here, d_i represents the difference between the ith paired data points X_i and Y_i, and n is the total number of pairs.

Ranks are assigned based on the absolute values ($|d_i|$) of the differences, from smallest to largest. If multiple differences have the same absolute value, they receive the same rank.

$$T+ = \sum_{j:d_j>0} R(j) \tag{6.8}$$

Here, Σ (sigma) represents the summation, j iterates over the data points where the difference (d_j) is positive, and $R(j)$ represents the rank assigned to the jth difference.

$$T- = \sum_{j:d_j<0} R(j) \tag{6.9}$$

This is used only for a one-tailed test and calculates the sum of the ranks ($R(j)$) assigned to data points with negative differences ($d_j < 0$).

- Two-tailed test: $\min\left(T+, \frac{n(n+1)}{2} - T+\right)$
- One-tailed test (greater than): $T+$
- One-tailed test (less than): $T-$

Here, $\min$ represents the minimum value, n is the sample size, and $\frac{n(n+1)}{2}$ is the sum of ranks from 1 to n. The decision statistic compares the sum of positive ranks ($T+$) with either the total possible ranks or the sum of negative ranks ($T-$) depending on the test type.

6.3.5 Wilcoxon Results for Time_Tool and Time_NoTool pair

Wilcoxon Signed-Rank statistic: 140.0

P-value: 0.16 . There is no significant difference between the time taken with and without the tool.[18]

6.3.6 Wilcoxon Results for Accuracy_Tool and Accuracy_NoTool pair

Wilcoxon Signed-Rank statistic: 93.5

P-value: 0.95 There is no significant difference between the accuracy with and without the tool.

Following up, the data was grouped by Accuracy_Tool and Time Difference (Time_Tool-Time_NoTool), calculating that mean. Then that mean was plotted against the accuracy level.

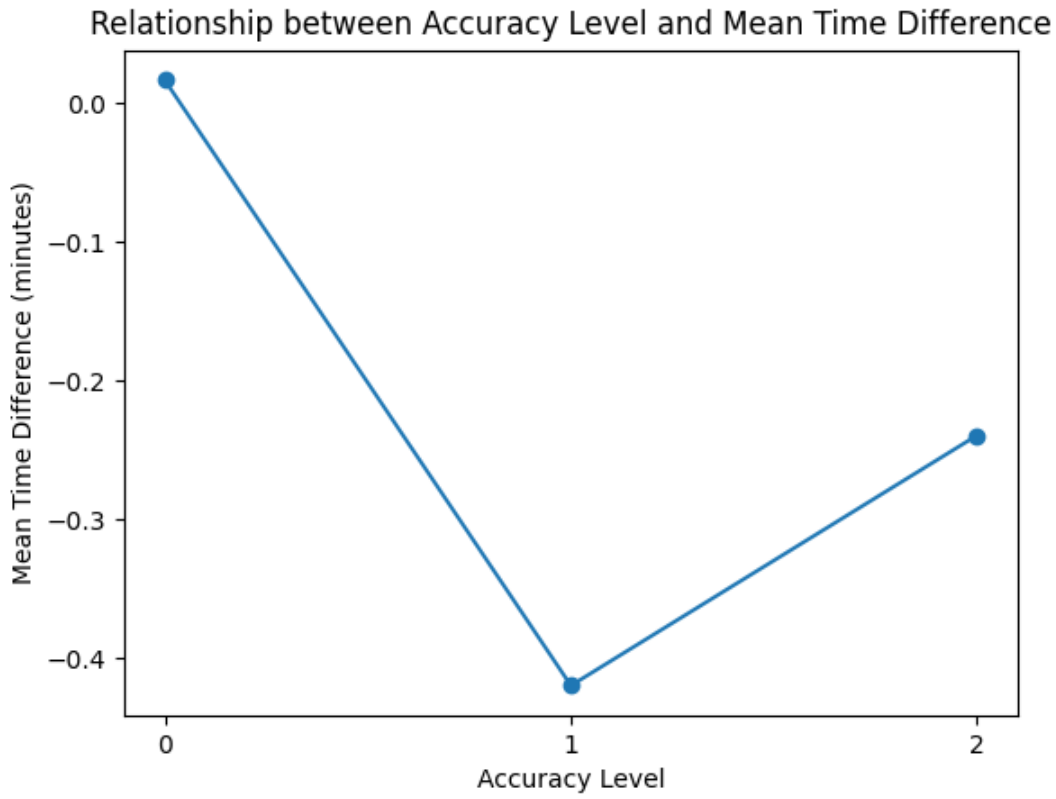


Figure 6.9: Accuracy VS Time Difference

Last, the data where the accuracy was larger with tool usage than no tool usage was found, and with calculating the mean of the accuracy of that data, they were plotted against each other.

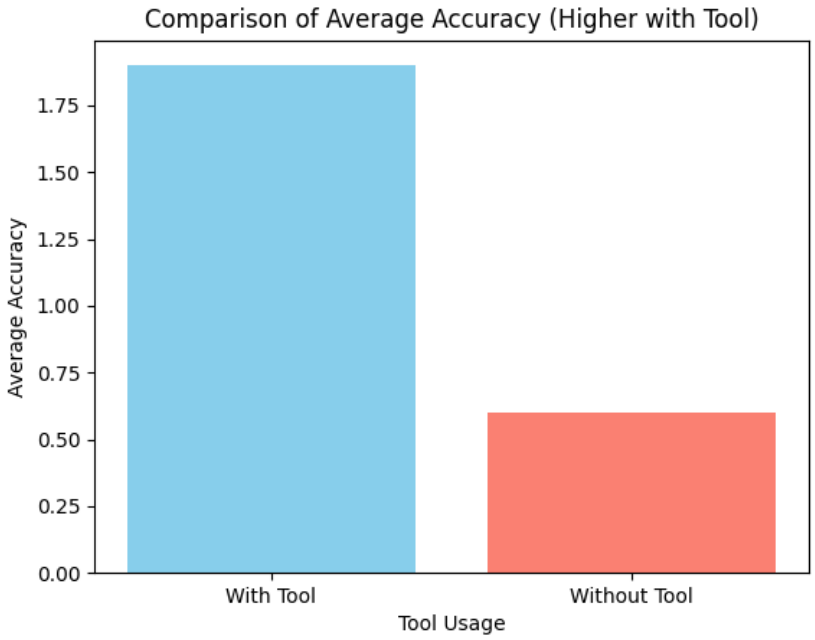


Figure 6.10: Average Accuracy VS Tool

6.4 Result Interpretation

The Statistics Table 6.7 gives the initial scope on how the variables are distributed. Unfortunately, from the nature of the Accuracy_Tool and Accuracy_NoTool variables being on a scale like form, the mean is identical as expected. The same goes for the Time_Tool and Time_NoTool variables where their mean difference is not ideal when the goal is to show the impact of the tool, i.e. if the mean of the Time_NoTool was smaller than the mean of the Time_Tool variable then the conclusion that the tool has indeed an impact would be noticable straight away.

As far as the density table 6.8, the results are somewhat more ambiguous. There's an interesting interaction between the variables when examining the graph. When the tool it used, we can see that at the distribution of time taken by accuracy level, for tasks labelled

as “Not Accurate” and “Somewhat Accurate”, data points cluster at lower time values (0.5 to 1.5 minutes), but those labeled as “Fully Accurate” have a higher concentration at bigger time values (2.5 to 3.5 minutes). This indicates that there is a trade-off—that is, utilizing the tool more frequently will result in higher accuracy. All of the accuracy variable scales do, however, exhibit considerable variability; some individuals reach high accuracy rapidly, while others require more time. Furthermore, the data points appear to be concentrated more in the “Somewhat Accurate” category, indicating that the tool may often produce findings that are fairly accurate, with fewer instances of very high or very low accuracy.

Following up with the Student T-Test and Wilcoxon test, all the P-values calculated respectively for each test and pairs, do not indicate any significant difference regarding tool usage impacting time to complete the tasks and their accuracy. We assume that is due to the low density of our dataset, causing the values to not be so spread out.

The data should ideally show a normal distribution, which is frequently represented as a bell curve, in order for statistical tests such as the T-Test to perform as well as possible. The distribution of data points around a mean, or central value, is implied by this bell curve. Generally, a more pronounced bell curve with a broader spread would result from a denser dataset with more data points. The tests would have been able differentiate between true tool impacts and random variation more effectively thanks to this dispersion.

The small size and low density of our dataset probably makes it more difficult for statistical testing to identify meaningful tool influence. Because of the low density in this case, data points can be grouped too closely together. The act of clustering reduces the range of possible data points, thereby increasing the difficulty of tests such as the Wilcoxon signed-rank test (which presumes independence within paired data) and the Student’s T-Test (which relies on normality) in detecting underlying patterns or meaningful differences between the groups that used and did not use the tool. Therefore, the tests may have misinterpreted the confined variation as being solely random chance, leading to the reported non-significant p-values.

In order to explain further how a larger and more complex dataset would indeed show the effectiveness of the tool usage we have to think of every data point as a combination of two factors, **e1**: positive impact of tool and **e2**: the negative impact of the complexity of the dataset, in this case, the complexity of going through the raw dataset by scrolling or by text search. When it comes to a less dense and less complex dataset, these factors can be small, and also being masked by random noise. As the dataset becomes larger and more complex, those factors accumulate when taking the sum of performance for every participant with :

$$Pi = \sum_{i:0} b + e1 + e2 \tag{6.10}$$

Where ***pi*** is performance for every participant $i > 0$, and ***b*** is base performance, hence the effects will be much more noticeable.

Higher degrees of unpredictability or randomness in the observations are intrinsically introduced by small datasets. Real effects are more difficult to discern from random noise when there is substantial variability, which is what most statistical tests seek for when looking for consistent patterns among fluctuation. This may go against the StudentT-Test or Wilcoxon assumptions of independence or normalcy, which further jeopardizes the validity and reliability of the findings.

In the Future Work section , some concrete next steps are defined that further measure the effect of the InK Browser.

When grouping the data by Accuracy_Tool usage and Time Difference, we got promising results. By calculating that mean and plotting the variables against the accuracy levels in 6.9, it is clear from the plot that the average to complete a task with a tool versus without the tool is in opposite relation. This indicates that when using the tool, task completion takes less time while the accuracy increases.

Last but not least, by viewing the latest plot 6.10, it can be concluded that the Average Accuracy while using the tool is significantly larger than without it.

Conclusion

This work addresses the problem of effectively examining and understanding organized complicated data represented with KGs. As stated in our hypotheses, we can say that the innovative configurable views feature of the InK Browser, when combined with its flexible limitations, enable a deeper and more effective analysis of structured data.

By using our preliminary work as foundation in ontology patterns while approaching them as a graph, we were able to dynamically construct the dataset's shortcuts.

Having the introductory InK Browser as groundwork from Dr Shimizu's work et al, we have managed to build on it and expand the notion of flexible views while using those shortcuts.

Having first used the BFS algorithm to successfully find every path in our testing dataset of the Agent-Role, we managed to extract the shortcut based on our baseline, the Agent. Given that, we then successfully constructed the SPARQL query that eventually queries our materialized dataset and gives us all the information related to our Agent in only one line. This runs concurrently along the InK Browser, showcasing that information to the users in a more flexible manner, giving them the change to traverse the dataset in that manner.

With our user study, even though our P-values could not show any significant difference in accuracy and time for tool usage versus examining raw data, through the mean we can come to the conclusion that indeed the usage of the InK Browser can improve accuracy in a shorter duration.

It is fair to say that the initial testing of the tool on a small public dataset is promising, leading the way into more applications.

7.1 Future Work

Given the lack of significant findings from our study, there are concrete steps for a better evaluation of the tool.

1. the AgentRole pattern should be materializing taking into consideration more Agents, the whole University for example. Raw data for 3 people as opposed to 1000 for example, would make the completion of tasks without the tool even more challenging, as the raw file would be much larger for the participants to scroll through. This will give the opportunity for the dataset to grow and be more distributed as this would enhance identifying patterns of impact of tool usage in time and accuracy of time completion.

2. Increased enrollment would result in a richer dataset, which may be achieved by extending the recruitment period. When thinking about the idea of growth rate, this can have a significant impact.

Take a look at the following formula, where the "Final Value" denotes the optimal participant count required for a definitive evaluation of the tool's efficacy. The number of participants in the current study is referred to as the "Initial Value". The following formula can be used to get a growth rate:

$$\begin{aligned}\text{Growth rate (\%)} &= \frac{\text{Final value} - \text{Initial value}}{\text{Initial value}} \times 100\% \\ &= \frac{20 - 14}{14} \times 100\% \\ &\approx 42.86\%\end{aligned}$$

Eventually with an additional 6 recruits , the growth of the participants could go up by

approximately 43% , aiding also to the performance effect discussed in 6.4.

3. There could be a potential addition to the Flexible Views of the InK Browser. It could be programmed to detect the nature of each dataset, customizing each view depending on that. For example, when the endpoint is populated with a dataset of geospatial form, then the InK Browser could automatically generate a map. These automation and customization of the views could give the users an even more superior experience when it comes to exploring complex and vast KGs.

Bibliography

[1] Apache jena. https://www.w3.org/2001/sw/wiki/Apache_Jena.

[2] Category:triple store. https://www.w3.org/2001/sw/wiki/Category:Triple_Store.

[3] ink-browser. https://github.com/kastle-lab/ink-browser.

[4] Student test - encyclopedia of mathematics, 2024.

[5] Sydney Woods Antrea Christou, Erin Rogers. Cs7810-metadata-representation languages - group3 project, 2023.

[6] Krzysztof Janowicz Blake Regalia and Gengchen Mai. Phuzzy.link: A sparql-powered client-sided extensible semantic web browser. *Computer Science*, page 44, 2017.

[7] Eva Blomqvist, Karl Hammar, and Valentina Presutti. Engineering Ontologies with Patterns – The eXtreme Design Methodology. In Pascal Hitzler, Aldo Gangemi, Krzysztof Janowicz, Adila Krisnadhi, and Valentina Presutti, editors, *Ontology Engineering with Ontology Design Patterns – Foundations and Applications*, volume 25 of *Studies on the Semantic Web*, pages 23–50. IOS Press, 2016.

[8] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui XIao. Ontop: Answering

SPARQL queries over relational databases. *Semantic Web*, 2017. To appear, available from http://www.semantic-web-journal.net.

[9] Rama Someswar Chittella. Leveraging schema information for improved knowledge graph navigation. *Computer Science*, page 62, 2019.

[10] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[11] Ramanathan V. Guha, Dan Brickley, and Steve Macbeth. Schema.org: evolution of structured data on the web. *Commun. ACM*, 59(2):44–51, 2016.

[12] Patrick Hayes and Peter Patel-Schneider, editors. *RDF 1.1 Semantics*. W3C Recommendation 25 February 2014, 2014. Available from http://www.w3.org/TR/rdf11-mt/.

[13] Pascal Hitzler, Aldo Gangemi, Krzysztof Janowicz, Adila Krisnadhi, and Valentina Presutti, editors. *Ontology Engineering with Ontology Design Patterns: Foundations and Applications*, volume 25 of *Studies on the Semantic Web*. IOS Press, Amsterdam, 2016.

[14] Pascal Hitzler, Aldo Gangemi, Krzysztof Janowicz, Adila Alfa Krisnadhi, and Valentina Presutti. Towards a simple but useful ontology design pattern representation language. In Eva Blomqvist, Óscar Corcho, Matthew Horridge, David Carral, and Rinke Hoekstra, editors, *Proceedings of the 8th Workshop on Ontology Design and Patterns (WOP 2017) co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017.*, volume 2043 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.

[15] Prateek Jain, Pascal Hitzler, Amit P. Sheth, Kunal Verma, and Peter Z. Yeh. Ontology alignment for Linked Open Data. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang 0007, Jeff Z. Pan, Ian Horrocks, and Birte Glimm, editors, *The Semantic Web – ISWC 2010 – 9th International Semantic Web Conference,*

ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I, volume 6496 of *Lecture Notes in Computer Science*, pages 402–417. Springer, 2010.

[16] Cogan Shimizu Joseph Zalewski, Lu Zhou and Pascal Hitzler. Ink browser – the interactive knowledge browser. *Computer Science*, page 5, 2021.

[17] M. Kejriwal, C.A. Knoblock, and P. Szekely. *Knowledge Graphs: Fundamentals, Techniques, and Applications*. Adaptive Computation and Machine Learning series. MIT Press, 2021.

[18] J.H. McDonald. Wilcoxon signed-rank test, 2014.

[19] Mark A. Musen. The Protégé project: a look back and a look forward. *AI Matters*, 1(4):4–12, 2015.

[20] Cogan Shimizu, Karl Hammar, and Pascal Hitzler. Modular graphical ontology engineering evaluated. In Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez, editors, *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, volume 12123 of *Lecture Notes in Computer Science*, pages 20–35. Springer, 2020.

[21] Cogan Shimizu, Karl Hammar, and Pascal Hitzler. Modular ontology modeling. *Semantic Web*, 14(3):459–489, 2023.

[22] Cogan Shimizu, Quinn Hirt, and Pascal Hitzler. A protégé plug-in for annotating OWL ontologies with opla. In Aldo Gangemi, Anna Lisa Gentile, Andrea Giovanni Nuzzolese, Sebastian Rudolph, Maria Maleshkova, Heiko Paulheim, Jeff Z. Pan, and Mehwish Alam, editors, *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers*, volume 11155 of *Lecture Notes in Computer Science*, pages 23–27. Springer, 2018.

[23] Cogan Shimizu, Quinn Hirt, and Pascal Hitzler. MODL: A modular ontology design library. In Krzysztof Janowicz, Adila Alfa Krisnadhi, María Poveda-Villalón, Karl Hammar, and Cogan Shimizu, editors, *Proceedings of the 10th Workshop on Ontology Design and Patterns (WOP 2019) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 27, 2019*, volume 2459 of *CEUR Workshop Proceedings*, pages 47–58. CEUR-WS.org, 2019.

[24] Cogan Shimizu, Quinn Hirt, and Pascal Hitzler. Modl: A modular ontology design library. Technical report, Wright State University, Dayton, Ohio, April 2019.

[25] Cogan Shimizu, Pascal Hitzler, and Adila Krisnadhi. Modular ontology modeling: A tutorial. In Giuseppe Cota, Marilena Daquino, and Gian Luca Pozzato, editors, *Applications and Practices in Ontology Design, Extraction, and Reasoning*, volume 49 of *Studies on the Semantic Web*, pages 3–20. IOS Press, 2020.

[26] Hogan Aidan Tartari Gonzalo. Wisp: Weighted shortest paths for rdf graphs. *Computer Science*, page 52, 2018.

Appendix A
Extracting Shortcuts

```
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX kastle-lab: <http://kastle-lab.org/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX momo:<https://archive.org/services/purl/domain/modular_onto

SELECT ?lhsLabel ?middle0Label ?middle1Label ?rhsLabel
    WHERE {
            ?lhs a momo:Agent ;
                rdfs:label ?lhsLabel .

            ?lhs kastle-lab:performsAgentRole ?middle0.
                ?middle0 rdfs:label ?middle0Label .

            ?lhs kastle-lab:performsAgentRole ?middle0.
            ?middle0 rdfs:label ?middle0Label .
            ?middle0 kastle-lab:hasTemporalExtent ?middle1.
            ?middle1 rdfs:label ?middle1Label .

            ?lhs kastle-lab:performsAgentRole ?middle0.
```

```
        ?middle0 rdfs:label ?middle0Label .
        ?middle0 ^kastle-lab:providesAgentRole ?rhs.
        ?rhs rdfs:label ?rhsLabel .
}
```