WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

EFMD EQUIS ACCREDITED

Bachelor Thesis

Visualizing Event-Driven Dynamics in Wikidata: Leveraging Large Language Models for Entity Evolution Analysis

Gregor Vandák

Date of Birth: 05.10. 2001
Student ID: 12128763

Subject Area: Information Business

Studienkennzahl: UJ033560

Supervisor: Dr. Amin Anjomshoaa

Date of Submission: 12. 01. 2025

Department of Information Systems & Operations Management, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

DEPARTMENT FÜR WIRTSCHAFTSINFORMATIK UND OPERATIONS MANAGEMENT DEPARTMENT OF INFORMATION SYSTEMS AND OPERATIONS MANAGEMENT

Contents

1 Introduction **8**

1.1 Problem Statement . . . 8

1.2 Motivational Use-cases . . . 10

1.3 Research Questions and Scope . . . 11

1.4 Thesis Structure . . . 12

2 Background and Related Work **12**

2.1 Knowledge Graphs . . . 14

2.2 Wikidata . . . 15

2.3 Event Knowledge Graphs . . . 16

2.4 Large Language Models . . . 17

2.5 Visualization . . . 19

3 Methodology **21**

3.1 Wikidata Querying and Change Detection . . . 22

3.2 EventKG Querying . . . 24

3.3 Large Language Model Prompting . . . 26

3.4 Visualization Implementation . . . 32

4 Discussion and Results **33**

4.1 Revisiting Research Questions . . . 37

4.2 Limitations and Future Work . . . 38

5 Conclusion **39**

List of Figures

1 Showcase of OntoDiffGraph (ODG) [24] 19
2 Showcase of AberOWL [42] 20
3 Workflow Diagram 21
4 Visualization Tool Showcase 32
5 Changes Made to the Entity "Boston" Between 2019 and 2021 34
6 Changes Made to the Entity "Boston" Between 2023 and Time of Writing 35
7 Node Details of Boston's "postal code" Changes 35
8 Changes Made to the Entity "Elon Musk" Between 2021 and 2023 36
9 Node Details of Elon Musk's "owner of" Changes 37

List of Tables

1 Comparison of Event Data Across Knowledge Graphs [16] . . 17
2 Example Outgoing Statements of Douglas Adams in 2021 . . . 23
3 Statement Counts of Different Entities in 2023 23
4 Example Event of Boston Between 2019 and 2021 25
5 Example Temporal Relation of Boston Between 2019 and 2021 26

Listings

1 Wikidata Query 22
2 EventKG Events Query 24
3 EventKG Temporal Relations Query 25
4 LLM Prompt for Selecting Event out of Provided List 27
5 LLM Prompt for Generating Event 28
6 LLM Output When No Event Caused Statement Change 29
7 LLM Output When an Event Caused Statement Change 30
8 LLM Output Without Any Events Available 31
9 LLM Output When Statement Change Is Typo Fix 31

Abstract

Collaborative open knowledge graphs undergo continuous updates from a diverse community of users. Because of this, analyzing the details or the sources of these changes can often be difficult. This thesis aims to address these challenges by proposing a novel approach to visualizing the evolution of knowledge graphs, focusing specifically on Wikidata. By integrating large language models (LLMs) together with event knowledge graphs, the research explores the possibility of contextualizing changes within the graph and linking them to real-world events. For instance, a statement added to a Wikidata entity associated with a public figure might correspond to a recent newsworthy event involving that individual. By providing explanations for these changes, this work seeks to enhance our comprehension of knowledge graph evolution, bridging the gap between raw data updates and their broader implications. Ultimately, this research aims to contribute to a deeper understanding of the dynamics that shape the evolution of crowd sourced knowledge bases such as Wikidata.

1 Introduction

In the current digital age, structured knowledge bases have become indispensable tools for organizing and disseminating information. They offer a flexible data representation in graph form, which is both machine readable as well as easy to edit and expand with new data [20]. Wikidata, a collaborative knowledge base project of the Wikimedia Foundation, serves as a centralized repository of structured data supporting its Wikimedia sister projects such as Wikipedia, Wiktionary, and numerous others [58]. Its strength lies in its dynamic and editable nature, allowing it to evolve in response to new information, corrections, and contributions from a global community of editors [57]. This constant evolution reflects not only the fluidity of knowledge but also the influence of real-world events, technological advancements, and cultural phenomena. Along with knowledge bases, large language models (LLMs) have been recently getting a lot of attention [48]. Their ability to interpret natural language has already proved to be useful in many natural language processing tasks, where they achieve state-of-the-art performance [30, 31]. Furthermore, thanks to their extensive training on large datasets derived from real-world data, they are highly effective at retrieving information in a manner similar to a search engine [31]. These qualities make them perfect for complex tasks, requiring an understanding of language as well as reasoning and knowledge of the world. However, LLMs lack genuine comprehension and reasoning, which may lead to factually inaccurate or misleading responses [31]. This means the outputs have to be carefully verified and evaluated and can not be taken as a fact.

This thesis aims to explore how the temporal changes in Wikidata entities can be effectively visualized and understood. The primary focus is on developing visualization techniques that highlight the evolution of entity data over time and employing LLMs to connect these changes to significant real-world events gathered from event centric knowledge bases. By analyzing the correlation between entity updates and external phenomena, this research seeks to create tools that not only make data changes more interpretable but also uncover the broader narratives that drive them. Ultimately, this thesis aspires to contribute to the growing field of knowledge graph analysis by offering tools and insights that illuminate the dynamic interplay between structured data and the ever-changing world it seeks to represent.

1.1 Problem Statement

Wikidata is a dynamic, collaborative knowledge base that evolves continuously as users contribute, update, and refine its content. These updates can be as simple as correcting a typographical error or as complex as adding a new entity with extensive metadata. According to Wikistats [52], more than 2.3 billion edits contributed by an average of 8 thousand users have been committed to Wikidata since its inception. While these edits improve the quality of Wikidata, they also create a volatile environment where understanding the reasons behind specific changes is challenging. With so many contributions happening at every moment, it can prove difficult to extract specific information on the time when a specific change happened and even more so what was the rationale behind this change.

The quality of data contained within Wikidata has been the subject of plenty of research. The literature survey conducted by Piscopo and Simperl [39] looked at 28 papers that addressed this topic. Their survey showed that areas such as data completeness and consistency together with accuracy are among the most studied dimensions [39]. On the other hand, timeliness and trustworthiness are among the less researched qualities, with timeliness only being addressed by two of the surveyed papers [39]. According to Zaveri et al. [62], timeliness refers to the up-to-date nature of the data. This means it can be evaluated by the frequency of updates, as well as the ability to check the validity of information and the modification date [13]. The frequency of updates is high for Wikidata, with on average 16 million edits happening per month [52]. However, when it comes to the validity of information and the modification date, only the former can be checked using qualifiers, while the latter is missing from Wikidata [13]. The metadata only provides the last date of revision of the entity as a whole, and not each of the statements connected to it.

To address this challenge, our research introduces a novel approach that leverages LLMs and event knowledge graphs in a visualization tool to facilitate easier interpretation of the evolution of Wikidata entities. A method similar to the one described by Y. Jin and S. Shiramatsu [22] is employed to establish causal relations between events and entity changes. Real-world events connected to the changed entity are extracted, and then a LLM is used to link the events to the changed statements. Through this process, valid sources and rationale for each of the changes are obtained. Additionally, by visualizing the changes in statements done to an entity over time, the timeliness of these changes can be interpreted, since the time period each change took place is pinpointed. Our methods could provide insights into the dynamics of knowledge creation, the spread of information, and the impact of global events on collaborative knowledge base development. Overall, our approach aims to provide a tool which can be further utilized to explore entity evolution dynamics, what drives these changes, as well as a deeper understanding of the entity itself.

1.2 Motivational Use-cases

Understanding the temporal evolution of Wikidata entities and their sources can be incredibly beneficial across a range of disciplines. Not only would it expand and deepen our understanding of collaborative knowledge graphs and the mechanisms underlying their development but also aid in a lot of diverse fields.

For instance, in social sciences often times the evolution of culture and societal trends is pivotal. Researching how our society shifts focus on different issues and topics can be difficult in our fast-paced digital world. Given that Wikidata seeks to represent our current knowledge of the world, by analyzing edits over time we can interpret these shifts in culture in different periods. By applying our approach, researchers could examine for example the number and nature of edits related to women in STEM fields. From this analysis, patterns might emerge that would indicate a growing societal interest in equal gender representation within traditionally male-dominated fields. Furthermore, leveraging the LLM feature of the tool, the researchers could identify important contextual factors influencing these changes, such as a surge in edits following a female researcher receiving recognition through awards or mainstream media coverage. This kind of analysis would be helpful in deriving what factors shape societal perceptions and how they evolve over time.

Similarly, our approach offers great potential for historical research. After all, historical knowledge evolves over time in response to new discoveries and reinterpretations. Because Wikidata is collaboratively edited by a vast number of volunteers from across the world, it can represent the common general knowledge and understanding about historical subjects. By using our approach, historians could get a better understanding at how collective knowledge and memory of specific events or historical figures change over time. For example, the removal of confederate monuments and memorials in the United States represents a significant shift in perception of historical figures. With our proposed approach, historians could inspect when these perception shifts occurred, as well as identify the events that may have contributed to them. While some cases, like the Confederate monument debate, are well-documented, many others lack such clear timelines. Our approach simplifies the process of identifying the timing of these changes, while also aiding in uncovering their possible causes.

1.3 Research Questions and Scope

This thesis seeks to address the overarching question: Can analyzing temporal changes in Wikidata entities provide meaningful insights into the relationship between these changes and real-world events? More specifically, it explores several sub-questions: Is there a discernible relationship between editing activity in Wikidata and significant events? To what extent can LLMs effectively link statement changes in Wikidata to world events? And, ultimately, does this approach provide a viable framework for understanding the dynamics of collaborative knowledge creation? By focusing on these questions, this thesis aims to evaluate the feasibility and utility of creating a new tool for visualizing and analyzing changes between time points in Wikidata entites, incorporating temporal analysis and insights derived from LLMs and real-world events.

The scope of this thesis is intentionally designed as a proof of concept. Temporal changes will be analyzed over four time points: 2019, 2021, 2021 and the present. These time points should be sufficient to test the feasibility of our approach and investigate whether any connections between events and statement changes can be identified. When it comes to the data from Wikidata used for analysis, focus will be kept on entity statement triples of subject, predicate and object. While Wikidata provides data beyond these statements, such as qualifiers, which provide additional details for each statement, these should not be necessary for analyzing event connections to statement changes. Real world events will be extracted from EventKG [15], an open curated knowledge graph of events and temporal relations, due to its ease of access as well as vast amount of event data containing time when compared to other open multiple domain knowledge bases. The study will leverage only open-access LLMs to ensure accessibility and reproducibility of methods. Even though both Wikidata and EventKG support multiple languages, this thesis will only be working with data in English, mainly to help our LLM generations be more consistent as well as simplifying any necessary pre-processing.

The deliverable for this research will include a tool released as an open-source web application as well as a repository on GitHub, providing a foundation for future exploration and development by the community. This tool will be designed to visualize entity changes, link them to corresponding real-world events, and facilitate deeper analysis. By offering this resource, the thesis aims to not only address the research questions but also contribute a scalable framework for investigating the evolution of collaborative knowledge bases and their relationship with the world they document.

1.4 Thesis Structure

To start, the background of the topics that are crucial to our approach are introduced. The ground is set by defining knowledge graphs and examining the structure of Wikidata, event databases such as the EventKG, as well as a brief discussion about LLMs and possible visualization techniques. In the Methodology section, the process for creating the proposed approach is detailed. Our queries for Wikidata and EventKG are introduced, as well as our change detection process. LLM prompts that connect the statement changes to events are showcased, also including some example outputs . On top of that, the features and development of our visualization tool are presented. Lastly, example use cases for our approach are displayed and the limitations and potential continuations of our work are discussed.

2 Background and Related Work

Before delving into the specifics of the proposed approach, it is essential to establish the foundational context of the research domain. This section explores knowledge graphs, which form the foundation of the research. Additionally, it dives deeper into the history and structure of Wikidata as well as introducing event centric knowledge graphs, focusing on EventKG. Since LLMs are utilized to connect events to statement changes, an introduction to the fundamentals of LLMs is also provided. Furthermore, various methods for visualizing ontology changes and evolution are discussed, which will be useful for developing our tool. Firstly, the work related to our goals will be examined.

An important starting point for our related work discussion is Wikidated 1.0 [47], where the authors present a method for constructing an evolving knowledge graph of Wikidata using its revision history. Their approach utilizes Wikidata dumps rather than queries, enabling the extraction of the complete historical dataset. The paper details their methodology and provides tools for generating this dataset. While the findings of the Wikidated 1.0 team are compelling, their approach lies outside the scope of this thesis. Nevertheless, their method for change detection proves highly valuable, and a similar strategy is adopted in this thesis. Notably, the authors focus on categorizing RDF triple additions and removals rather than performing a detailed analysis, which aligns well with our objectives, as it avoids the risk of making incorrect assumptions about the nature of edits.

The master thesis by N. Krenn [23] also explores the evolution of Wikidata and presents a related perspective. In this work, the author investigates the feasibility of achieving topic-wise modularization of Wikidata by clustering users based on their utilized vocabulary. By extracting the complete edit history of Wikidata from dumps and performing extensive pre-processing, the author creates a valuable resource for subsequent research. Preliminary network analyses are conducted through clustering, revealing that clustering based on users' entity similarity yields coherent results, whereas clustering based on users' vocabulary does not produce clear outcomes. The thesis offers valuable insights into the underlying dynamics of collaborative knowledge graphs and establishes a foundational framework for further research in this area. However, it primarily focuses on the evolution of Wikidata's schemas and the behavior of its contributors rather than the changes to the content itself. In contrast, our approach centers on analyzing and contextualizing the changes made to the statements within Wikidata, rather than its editors. While this work provides a helpful reference, it does not directly address the specific challenges targeted in this research.

The work of Y. Jin and S. Shiramatsu [22] has been highly influential for our methodology, as it addresses a comparable challenge in linking events to entities. In their paper, the authors present an approach that leverages Generative Pre-Trained Transformer 3 (GPT-3) [4], a type of LLM pre-trained on a large dataset and fine-tuned for specific tasks [41]. This approach aims to extract causal relationships of news to entities from news texts and further infer the impact of these events on related entities. Our approach builds on this concept but introduces enhancements by sourcing a curated list of events from EventKG rather than extracting event data directly from news texts. Additionally, these events are linked to specific statement changes within Wikidata rather than associating them with entities more generally. This refinement allows us to achieve a more detailed understanding of how individual events influence specific statement changes and contribute to the broader evolution of the entity. This approach aims to provide a richer contextualization of the dynamic interplay between events and entity evolution. The insights and methods presented in Y. Jin and S. Shiramatsu [22] have been valuable as a point of reference, enabling us to refine and adapt their approach to address the specific objectives of our research.

C. Sarasua et al. [46] conducted an in-depth analysis of the editing behavior of different types of Wikidata contributors, yielding notable insights. The study revealed that most edits are performed by a small group of highly active editors, called power editors, while the majority of users contribute only a few edits. Interestingly, the findings also highlighted that there is no direct correlation between an editor's lifespan on the platform and their volume of edits. Additionally, the research identified that the average duration of an edit session on Wikidata is approximately four times longer than on Wikipedia. Another intriguing observation was the difference in editing patterns between power editors and standard editors. Power editors tend to contribute at a steady pace, whereas standard editors display fluctuating activity levels, with periods of increase and decrease. This dynamic may partially relate to our approach, as it is hypothesized that real-world events could influence these non-linear editing patterns. This research offers compelling insights that could inform future studies on the drivers of Wikidata changes. Its exploration of editing dynamics serves as a useful foundation for further investigations into the interplay between editor activity and the evolution of Wikidata content.

2.1 Knowledge Graphs

Knowledge graphs have become integral to the organization and utilization of large-scale, interconnected data. A. Hogan et al. define knowledge graphs as: *"a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities"* [20]. Knowledge graphs provide a framework for capturing and linking diverse information in a way that is both machine-readable and contextually meaningful [20]. Unlike traditional relational databases, which primarily store data in rigid tabular formats, knowledge graphs adopt a flexible, semantic structure that enables dynamic query generation and integration of heterogeneous datasets [1].

This adaptability has led to their widespread application in many fields and companies [32]. When utilized by a company, they are referred to as enterprise knowledge graphs,which are typically private to the company and used for commercial applications [32]. Notable industries leveraging enterprise knowledge graphs include commerce, web search, finance and social networks among others [20]. On the other hand, open knowledge graphs are publicly available for everyone to use. Some examples of open knowledge graphs include DBpedia [26], YAGO [19], and Wikidata [58]. The listed graphs span multiple domains and languages, however domain specific knowledge graphs also exist, focusing on for example government, life sciences or world events.

A core feature of knowledge graphs is their reliance on graph-based data models. Nodes in the graph represent entities, such as people, places, or concepts, while edges denote the relationships between these entities, forming a web of interconnected data. This design not only reflects the complexity of real-world information but also allows for intuitive visualizations and deep analytics. It does not require hierarchical organization of the data, which allows for modeling of complex and interconnected relationships more flexibly than hierarchical structures like trees [20]. Additionally, it provides the added benefit of flexibility for incorporating new data sources, unlike the standard relational model, which requires a predefined schema that must be adhered to throughout the process [20].

A widely used data model that captures this structure is the Resource Description Framework (RDF) [9]. In the RDF model, data is represented as a triple of a subject, a predicate and an object, where we can imagine that the subject is connected to the object via the predicate [9]. This means that for knowledge graphs, subject and object will denote entities or nodes, while predicate will stand for relationships or edges between them. To access data stored in RDF format remotely, SPARQL endpoints are commonly used. SPARQL is a semantic query language designed for databases that store data in RDF format [36]. It allows data to be retrieved, filtered, or matched against specified patterns using SPARQL [36]. A basic SPARQL query typically includes prefixes, which declare namespaces for compact URIs, a SELECT clause that specifies the variables to return, and a WHERE clause that defines the graph patterns to match. A SPARQL endpoint refers to an access point that enables clients to send queries to the database.

With such a big focus on flexibility, it is no surprise knowledge graphs go through a lot of changes. Over time, they evolve as new information is added, existing relationships are refined, and outdated data is corrected or removed. This dynamic nature reflects the continual development of human understanding and real-world events, but it also poses significant challenges for those seeking to analyze and interpret this evolution. These changes are either automatically extracted from outside sources [19, 26], or they are built and kept by a collaborating volunteers [58]. Understanding why specific changes occur, whether due to emerging discoveries, societal shifts, or the resolution of controversies, requires contextual awareness that goes beyond the graph's immediate data.

2.2 Wikidata

Wikidata is a collaboratively edited open knowledge base operated by the Wikimedia Foundation, designed to serve as a structured data repository supporting Wikipedia and other Wikimedia projects [58]. Launched in 2012 as a small project, it aimed to centralize the management of factual data, ensuring consistency and reducing redundancy across various Wikimedia platforms [57]. At the end of 2014, the other major collaborative knowledge base named Freebase [2] developed by Google was shut down in favor of Wikidata due to its success [35]. As part of the shut down, Google agreed to transfer the data of Freebase to Wikidata [59]. This resulted in a great data migration effort, where issues such as data quality and a lack of references, which are crucial to the community of Wikidata had to be addressed [35]. In the end, 14 million new statements were added to Wikidata, which constituted a 21% increase in its size at the time [35].

Currently, Wikidata is crucial for the Wikimedia ecosystem, as well as many further applications, which rely on its machine-readable data [59]. For example, the different pages of Wikidata have been viewed a total of 279 million times during 2023 alone [52]. At the time of writing in January, 2025, it contains almost 115 million data items, which total 13 billion words [52]. This content has been managed by a total of 6.8 million unique users, 23.8 thousand of whom are still currently active [52]. This data has been updated a lot throughout the life span of Wikidata. Specifically, there have been nearly 2.3 billion edits committed, which translates to roughly 19 edits per page on average [52].

The structure of Wikidata mostly follows knowledge graph logic, so data inside it is represented by a RDF triple of subject, predicate and object [12]. The exception to this would be more complex data, that is not possible to represent in such a simple manner. For this data, subordinate property value pairs are used called "qualifiers" [33]. These qualifiers contain contextual information about given basic statements. Users all around the world can add and edit these statements [34]. This structured data format allows for the integration of diverse information across different languages and disciplines, promoting a multilingual and universal knowledge ecosystem [64]. Most importantly, all of this data is accessible for free public use by anyone. It can be extracted directly in the form of weekly JSON dumps of the entire knowledge base that can then be further processed. Alternatively, it can be accessed via SPARQL endpoints, which allow for extensive querying of the knowledge graph and opens the door for a plethora of uses ranging from enhancing search engines [44] to improving automatic text generation [45, 6].

2.3 Event Knowledge Graphs

Event knowledge graphs are simply domain specific knowledge graphs that focus on events. They are usually formed by extracting existing event data from other more general knowledge graphs or other sources of data such as Wikipedia [17]. Another approach for creating event knowledge graphs is by scraping news websites, where each article is automatically examined for relevant events and entities and added to the knowledge graph in real time [17]. As events encapsulate occurrences in the real world, it is evident that this information holds significant value across a wide range of fields. Because of this, they are often further specified by only focusing on events from concrete domains, such as tourism [27], finance [10] or sports [43]. Nonetheless, more general event centric knowledge graphs are still a very valuable resource for plenty of applications, such as search [61, 44], text generation [8] or recommendation [60].

For the purposes of our work, EventKG [15] will be utilized. EventKG is a multilingual event-centric temporal knowledge graph developed to facilitate a global view on events and temporal relations [14]. The inclusion of both events and temporal relations make EventKG a fusion of structured and semi-structured data in one knowledge graph [14]. It was created by extracting event data from existing knowledge bases, namely Wikidata [58], DBpedia [26] and YAGO [19] as well as sources like the Wikipedia Current Events Portal [50] and Wikipedia event lists [51] in 15 languages [14]. Table 1 shows that this leads to EventKG containing more events than any of its sources, while also expanding on their details such as the time of the event. Note that for semi-structured sources like Wikipedia event lists and Wikipedia Current Events Portal, locations were not extracted [15]. Furthermore, the entire knowledge graph is available for public use, either by downloading the entire graph or through the official SPARQL endpoint [15]. All these qualities make EventKG perfect for our proof-of-concept work. It is pivotal for our approach to have reliable, open source event data which contains temporal information so that real-world events could be linked to Wikidata statement changes. EventKG fulfills this role for our time points, while providing sourced data for not just events, but also temporal relations which could also be linked to statement changes.

Knowledge Graph	**Events**	**Known Location**	**Known Time**
EventKG	1,348,561	496,537	1,151,734
Wikidata	715,286	480,616	498,930
YAGO	262,286	63,051	89,276
DBpedia (en)	302,274	13,784	31,845
Wikipedia event lists (en)	208,727	0	208,727

Table 1: Comparison of Event Data Across Knowledge Graphs [16]

2.4 Large Language Models

Large Language Models (LLMs) are a class of machine learning models designed to process and generate natural language. In the context of this thesis, LLMs are employed to analyze and select events that are connected to Wikidata statement changes. Although this thesis primarily focuses on knowledge graphs and their visualization, a foundational understanding of LLMs is necessary.

LLMs are built upon the transformer architecture, which was introduced by Vaswani et al. [55]. These models are pre-trained on extensive datasets comprising textual data from diverse sources such as books and the internet [63]. The training of LLMs occurs in two main stages: pre-training and fine-tuning. Pre-training involves exposing the model to vast amounts of text to learn general linguistic patterns and representations [55]. During this stage, the model develops a foundational understanding of language, which it applies broadly across different contexts. Fine-tuning is a subsequent step, where the model is further trained on task-specific datasets to refine its parameters [31]. This process ensures that the model is tailored to address particular domains or perform specific tasks more effectively.

The training process begins with tokenization, a technique that segments text into smaller units called tokens. Tokens can represent words, subwords, or individual characters depending on the tokenizer employed [29]. The transformer processes input tokens in parallel while maintaining their positional information through positional encodings [55]. A key component of the transformer architecture is the attention mechanism, which assigns dynamic weights to tokens based on their contextual relevance within a sequence [55]. This mechanism enables the model to capture relationships between tokens, regardless of their position in the text. During training, LLMs learn to adjust billions of parameters, which encapsulate the knowledge extracted from the training data [55]. These parameters govern the model's behavior and serve as the foundation for generating outputs. When generating outputs, the user's input is tokenized, and the model predicts the most likely continuation of the input based on its learned parameters [31]. This process involves selecting tokens iteratively, with each selection informed by the preceding context, thereby constructing coherent and contextually appropriate responses.

Despite their remarkable capabilities, LLMs have notable limitations. They often reflect biases inherent in their training data, which can lead to biased or inappropriate outputs [63]. Additionally, while LLMs generate outputs that are contextually plausible, they lack true understanding or reasoning, which can result in factually incorrect or misleading responses [31]. These challenges show why it is important to always evaluate and verify responses generated by LLMs, especially when accuracy and validity is critical, as is the case in this thesis.

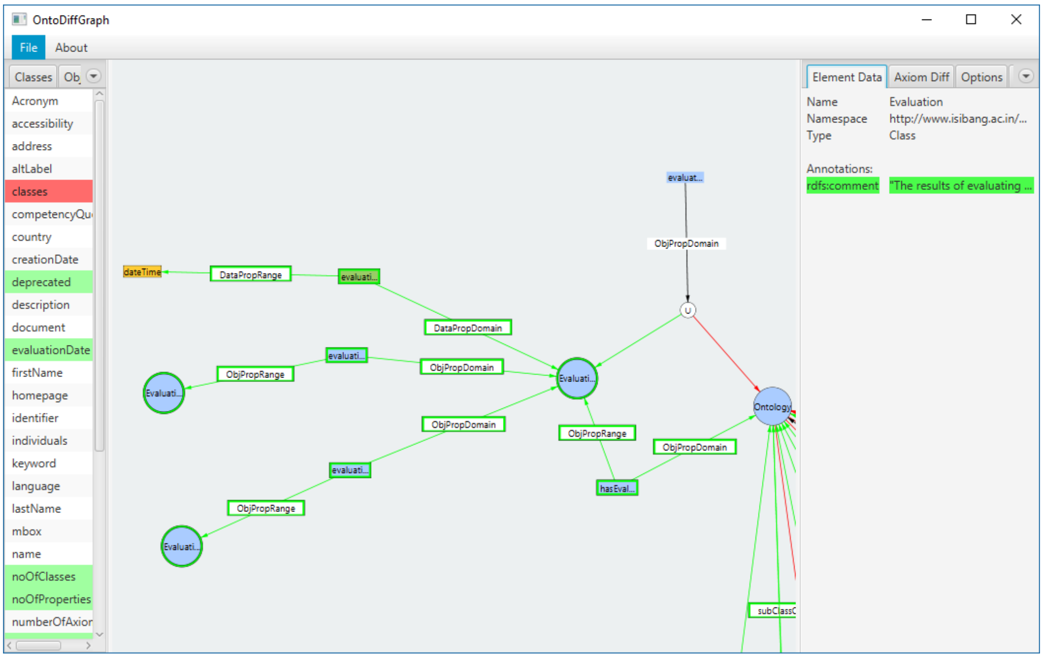

Figure 1: Showcase of OntoDiffGraph (ODG) [24]

2.5 Visualization

There is extensive literature on the visualization of changes, particularly when generalizing the topic to ontology visualization. The poster developed by K. Chung et al. [7] provides a good baseline introduction to the topic. In their work, the authors present three dominant methods for visualization of ontology change: graphical notation, list visualizations, and abstraction networks [7]. These methods are evaluated based on their change representation, scalability and contextual informativeness. Of particular interest to our research is the graphical notation approach, exemplified in the poster by the OntoDiffGraph (ODG) [24]. The ODG organizes changes in a hierarchical structure, with each state differentiated through a color scheme [24]. Figure 1 shows an example of how this method looks. While this approach lacks scalability, this limitation is not a problem for our purposes, as our focus is on visualizing only the changes within the ontology rather than highlighting the changes within the entire ontology itself. Furthermore, the ODG does not support the visualization of complex changes, but this is also not a concern for our work, as our focus is limited to the addition and removal of statements. The hierarchical representation provided by the ODG aligns well with our objective of presenting statement changes in a clear and organized manner, making it a suitable choice for our visualization needs.

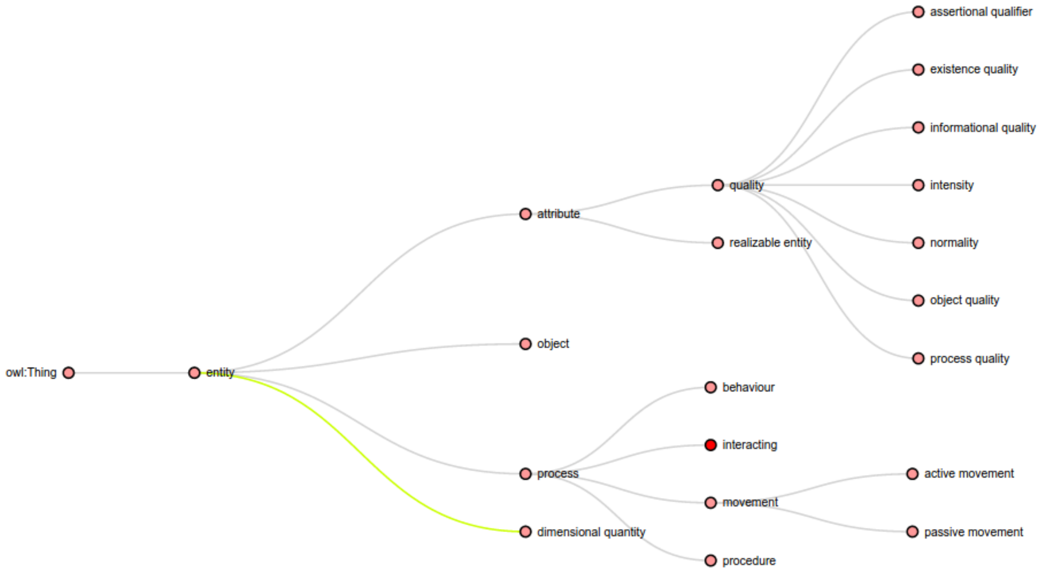

Figure 2: Showcase of AberOWL [42]

The literature review conducted by R. Pernish et al. [38] serves as a highly valuable resource for our selection of a visualization method. The authors perform a systematic review, evaluating 28 ontology visualization tools, which they categorize into five primary groups. These groups include list visualizations, graph visualizations, visualizations combining lists and graphs, visualizations combining lists, statistics and graphs, and mixed visualizations [38]. The paper goes on to analyze trends in the field, noting, for instance, that ontology visualization research peaked in 2008 and 2017 [38]. Additionally, the focus has shifted from visualizing changes to capturing evolutions, with a slight preference for interactive, graph methods emerging in recent years [38, 40, 37]. Once again, graph visualization stood out to us as the best choice for our purposes based on the analysis of the survey. Among the tools reviewed, the approach employed by AberOWL [42] was of particular interest, a showcase of which is in figure 2. It utilizes a straightforward yet effective method of visualizing direct changes. Changes are distinguished through color differences within a hierarchical structure, making this method especially relevant to our goals.

Based on the reviewed resources, an interactive graph visualization approach was ultimately selected. This method offers the advantage of producing clear and intuitive visualizations, enabling even non-experts to comprehend the underlying structure of ontologies with ease. Its limitations, such as a lack of scalability and the inability to represent complex changes,

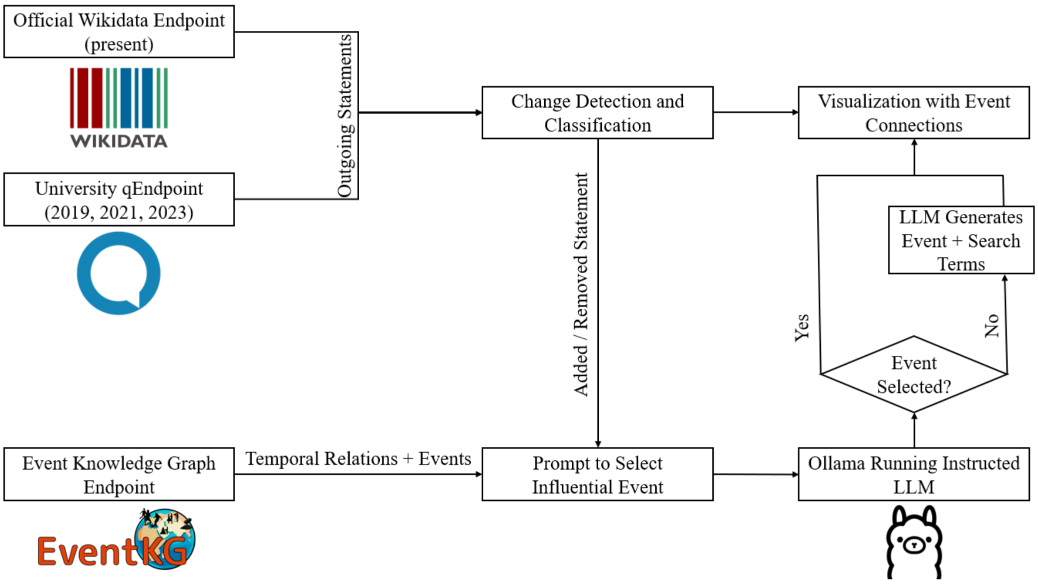

Figure 3: Workflow Diagram

are not significant concerns for our specific objectives. Hence, it was determined that the most effective solution would be a interactive webgraph featuring color differentiation for the different states of nodes, similar to the approach showcased by AberOWL [42]. To enhance usability and reduce visual clutter, a supplementary side panel displaying detailed information about selected nodes will also be incorporated. This design ensures that the visualization remains both accessible and comprehensive without sacrificing critical details.

3 Methodology

To provide a clearer understanding of the proposed method, our workflow is outlined in figure 3. The process begins with the collection of two Wikidata datasets containing the outgoing statements of a specific entity at two distinct time points. Simultaneously, EventKG is queried to retrieve the temporal relations and events associated with the entity. Next, change detection is performed on the two Wikidata datasets to identify statements modified between the selected time points. Finally, an instructed LLM is utilized to analyze the detected changes and events, in order to select an event connected to the statement change. If no event is selected, either due to no events being available or simply no event fitting the statement change, the LLM generates an event that could explain the change instead. The results are subsequently integrated into an interactive webgraph visualization, providing a comprehensive representation of the entity's evolution over time.

3.1 Wikidata Querying and Change Detection

The first step in our approach is gathering Wikidata outgoing statements from two different time points. This requires extracting not only the human-readable labels of the statement triples but also their corresponding Wikidata codes, which are essential for conducting the change detection.

To obtain the required datasets, the challenge of accessing historical data must first be addressed. While the SPARQL endpoint provides access to current data without difficulty, historical data retrieval relies on the availability of weekly database dumps provided by Wikidata. For this thesis, historical data was sourced from the Wikidata endpoints offered by the Vienna University of Economics and Business using qEndpoint SPARQL endpoints, which includes database dumps from 2019, 2021, and 2023. As the university endpoint also uses SPARQL, the same query developed for the official Wikidata endpoint can be utilized without modification, streamlining the data extraction process. A sample query for Wikidata Entity Q42 (Douglas Adams) which we use in our proposed solution is presented in listing 1.

```
PREFIX bd: <http://www.bigdata.com/rdf#>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wikibase: <http://wikiba.se/ontology#>
SELECT ?subject ?subjectLabel ?predicate ?realpredicateLabel ?object ?objectLabel
WHERE {
  #Set searched for entity as subject to get outgoing statements
  BIND(wd:Q42 AS ?subject).
  #Get the predicate and object
  ?subject ?predicate ?object.
  #Resolving direct claims
  ?realpredicate wikibase:directClaim ?predicate
  #Getting the labels in english
  SERVICE wikibase:label {bd:serviceParam wikibase:language "en".}
  }
```

Listing 1: Wikidata Query

An example of the data gathered can be seen in table 2. However, not all Wikidata statements are relevant for tracking meaningful changes. One such case would be the various identifier codes of entities in various databases, such as the International Standard Name Identifier (ISNI) or KBpedia. While this information is essential for maintaining data integrity, it is less significant for

Subject	Object	Predicate
Q42	P123	Q1
Q42	P456	Q10
Q42	P321	Q999
SubjectLabel	**ObjectLabel**	**PredicateLabel**
Douglas Adams	field of work	science fiction
Douglas Adams	notable work	Dirk Gently series
Douglas Adams	spouse	Jane Belson

Table 2: Example Outgoing Statements of Douglas Adams in 2021

Subject	Total Statements	Statements with IDs removed
Douglas Adams	294	80
Boston	311	193
World War II	193	84

Table 3: Statement Counts of Different Entities in 2023

event-driven change analysis. To address this issue and also reduce clutter in the visualization later on, an option was implemented to filter out all statements that include "ID" using regular expression. It should be noted that not all identifiers are removed this way, since some of them do not conform to the "ID" nomenclature (e.g. ISNI). The importance of this filtering step is evident in table 3, which demonstrates how many statements get filtered out for various example entities. This filtering process enhances the clarity of the visualizations, enabling a sharper focus on statement changes that can potentially be linked to real-world events.

The next step in the methodology involves change detection and classification. The topic of classification is more complex of the two. There can be a lot of reasons for edits, such as correcting typographical errors, adding new information, elaborating on existing information, or removing inaccurate statements. To fully capture this complexity, a large number of classes would need to be created. Capturing this level of detail would require the creation of numerous change categories, which could introduce ambiguity given the limited context provided by the statement triples alone. Therefore, an approach inspired by Wikidated 1.0 [47] was adopted, in which changes are classified into two broad categories of statement additions and removals. This approach ensures accuracy by avoiding potential misclassification of changes while leaving the interpretation of specific change types to the user analyzing the changes. To identify changes between two time points, all predicates and objects connected to a given subject at each time point are compared. Since the subject is fixed for all statements, the comparison focuses on determining whether the same predicate-object pair exists in both datasets. The comparison is performed using the Wikidata identifiers rather than labels to ensure consistency in format. A statement found only in the newer dataset is classified as an addition, while one appearing exclusively in the older dataset is classified as a removal. This streamlined classification process ensures clarity and precision in the detection of changes.

3.2 EventKG Querying

To utilize event data for connecting or explaining statement changes, it is essential to gather as much data as possible. This process involves collecting both temporal relations and events. The EventKG endpoint uses SPARQL, so it has some similarities to our Wikidata query, however there are quite a few differences as well. By default, EventKG selects the highest rank source for an event when multiple sources are available. To ensure the capture of all possible event information, data from all available sources is retrieved, with duplicates filtered out afterward. This approach ensures that any details captured by one source but omitted by another are preserved. Additionally, only events occurring within the specified time scope are of interest. For this purpose, the starting time of events is filtered directly within the query. However, some events, especially ongoing ones, lack an end time and filtering for their end time within the query would lead to their exclusion. To address this, filtering for event end times is done to the output stage of the query, alongside the duplicate filtering process. The SPARQL query used for extracting events is provided in listing 2. Textual events are presented in the form of descriptions, typically as short paragraphs. An example output can be observed in table 4.

```
SELECT DISTINCT ?beginTime ?endTime STR(?description) AS
?description ?graph
WHERE {
    ?actor owl:sameAs dbr:Douglas_Adams .
    #choose only textual events
    ?event rdf:type eventkg-s:TextEvent .
    #choose events where Douglas Adams is involved
    ?event sem:hasActor ?actor.
    #get source of the information
    GRAPH ?graph {?event sem:hasBeginTimeStamp ?startTime . }
    #get description of the event in english
    ?event dcterms:description ?description .
    FILTER(LANG(?description) = "en") .
```

beginTime	endTime	description	source
2020-06-09	2020-06-09	Statues of Christopher Columbus are beheaded in Boston, Massachusetts, and knocked over in Richmond, Virginia in support of Native American rights.	Wikipedia_en

Table 4: Example Event of Boston Between 2019 and 2021

```
    #if there is a start date, filter based on our needs
    ?event sem:hasBeginTimeStamp ?beginTime .
    OPTIONAL { ?event sem:hasEndTimeStamp ?endTime .}
    FILTER (?beginTime >= "{start}-01-01"^^xsd:date)
}
ORDER BY ?beginTime
```

Listing 2: EventKG Events Query

The temporal relations have the form of a triple the same as with Wikidata. Since no comparisons between datasets are required for this task, only the human-readable labels of the relations are extracted. The query for retrieving temporal relations is shown in listing 3, with an example output provided in table 5.

```
SELECT DISTINCT ?beginTime ?endTime (STR(?propertyLabel)
AS ?propertyLabel) (STR(?objectLabel) AS ?objectLabel) ?graph
WHERE {
    ?actor owl:sameAs dbr:Douglas_Adams .
    #choose temporal relations connected to Douglas
   Adamschange
    ?relation rdf:subject ?actor .
    ?relation rdf:object ?object .
    #get the source of the relation
    GRAPH ?graph {?object rdfs:label ?objectLabel .}
    ?relation sem:roleType ?roleType .
    #get the property label of the relation
    ?roleType rdfs:label ?propertyLabel .
    #make sure we only get english labels
    FILTER(LANG(?propertyLabel) = "en") .
    FILTER(LANG(?objectLabel) = "en") .
    #if there is a start date, filter based on our needs
    ?relation sem:hasBeginTimeStamp ?beginTime .
    OPTIONAL {?relation sem:hasEndTimeStamp ?endTime .}
    FILTER (?beginTime >= "2019-01-01"^^xsd:date)
}
```

beginTime	endTime	propertyLabel	objectLabel	source
2021-03-22	2021-11-16	head of government	Kim Janey	wikidata

Table 5: Example Temporal Relation of Boston Between 2019 and 2021

```
ORDER BY ?beginTime
```

Listing 3: EventKG Temporal Relations Query

3.3 Large Language Model Prompting

The implementation of the large language model (LLM) begins with addressing the issue of model selection. As outlined in the project scope, only open-source models are considered. This limitation does not significantly restrict the available options, as numerous state-of-the-art models are accessible. Four models were selected for evaluation: LLama3.1 with 8 billion parameters [11], Gemma2 with 9 billion parameters [49], Mistral with 7 billion parameters [21], and Falcon2 with 11 billion parameters [28].

To determine the most suitable model, extensive testing was conducted during the development of prompts. Each prompt iteration was applied to all four models to evaluate their responses to changes in prompt structure and style. Ultimately, LLama3.1 was selected for its superior consistency and reasoning capabilities for our task. Following this decision, the prompts were refined further, with the fine tuning process continuing until the outputs demonstrated satisfactory consistency and logical coherence.

The main task of the LLM is to analyze the statement change that occurred as well as all the events connected to the entity and determine whether one of the events could plausibly cause the change to have happened. An additional prompt is required for cases where no relevant events are identified within the specified time period or where none of the provided events appear to be connected to the change. This alternative prompt instructs the LLM to rely on its internal knowledge base, leveraging its understanding of the entity and its historical context to propose an influential event that could account for the change. This dual-prompt approach ensures that, regardless of the availability or quality of event data, a potential link between the statement change and an event can be established. Importantly, the outputs generated by the prompts will require user validation, since there is a chance the model hallucinates an event which does not exist, or making a wrong selection.

To optimize the performance of the LLM, various prompt engineering techniques were employed, informed by academic literature [3, 5, 18, 56] and insights from LLM forums. The prompts utilize a standard temperature of 0.8 and a top-p value of 40, as these prompt settings yielded the most favorable results during testing. Additionally, a seed is applied to ensure reproducibility, making the outputs consistent across repeated generations. The first prompt used in the implementation is presented in detail in listing 4.

```
0. Write out your entire process as listed here.
1. You will receive an added or removed Wikidata statement outgoing from a subject and events that are connected to the same subject in this format (Change_type: ... ; Statement: {...} ; Events: [[event_code, event], ...])
2. Analyze the meaning of the statement and the events. Concentrate on events related to the meaning of the statement.
3. Identify if any of the events directly caused the statement. Be conservative and only choose an event if all the necessary information is provided and no assumptions are needed. Pick only one event.
4. At the end of your process, return -1 if no event caused the statement. If there is such an event, return the event code.
5. Make sure to put your answer inside asterisks, for example: *-1*.
```

Listing 4: LLM Prompt for Selecting Event out of Provided List

As the task involves rational analysis of the presented data, employing a step-by-step approach through chain-of-thought prompting has been found to be highly effective [5]. This technique ensures that no steps are skipped during the reasoning process, while also encouraging the model to evaluate its thinking systematically, leading to improved decision-making [56]. Input and output formats are explicitly provided to enhance the consistency of responses [3]. Additionally, terms such as "conservative" are used to guide the model to function as a strict analytical tool, minimizing the risk of biased assumptions about the provided event data [3]. To prevent the model from altering or fabricating event data, it is instructed to return the index of the selected event rather than providing modified or newly generated event details. The indexed event can then be retrieved from the original list, allowing access to the full event information, such as the time point it happened, or the source. If the model were simply instructed to pick a causal event, this would lead to the model always looking for a suitable event, making up details about the event or statement to justify the connection [5]. To address such scenarios, the model is given the option to return "-1" if none of the provided events can be directly linked to the statement change. Since the generation process involves the model writing out its reasoning steps, extracting the final index answer can be challenging. To tackle this, the model is instructed to enclose its final answer within asterisks, enabling efficient extraction via regular expressions. This requirement is reiterated within the user prompt to ensure compliance and reliability [5]. The prompt for generating an event based on the knowledge base of the LLM utilizes some different techniques. The prompt is detailed in listing 5.

```
 You are an assistant with expert knowledge of world events as well as Wikidata triple statement logic.
 You will be provided with an added or removed Wikidata statement and the time period during which this change happend in this format (Change_type: ... ; Statement: {...} ; Period: year1 - year2).
 Your task is to use your knowledge of the subject and its history to explain why the statement change happend. Think about the statement logically. Consider if the change is something that was caused by an outside event or just a routine fix.
 You will respond with a brief explanation, as well as search terms that can be used to find more information about the event in brackets.
 Do not add any further text into your response. Keep the statement codes out of your explanation.
```

Listing 5: LLM Prompt for Generating Event

This prompt focuses on generating responses, which allows for a more streamlined approach compared to the step-by-step reasoning required in the first prompt. The step-by-step process was not chosen here to eliminate the need for extensive filtering, such as extracting final outputs using asterisks in the earlier approach. Instead techniques like role assignment are employed, which facilitate responses more aligned with the intended purpose [18]. Input formats and rules for the final output are explicitly defined, and these rules are reinforced within the user prompt to ensure compliance [3, 5]. To address the fact that many statement changes, such as typo corrections, do not correlate with significant events, the model is instructed not to force a connection when one does not exist [18]. Therefore the model should expect such cases and react accordingly. Otherwise, the model might try to make up events which did not happen leading to confusion. Nevertheless, avoiding hallucinations entirely remains challenging, and human validation is essential. Attempts to address hallucination by instructing the model to provide a source for the generated events were unsuccessful, as the model often produced non-existent article links. This limitation arises from the fundamental nature of LLMs, which lack true reasoning or factual knowledge and instead rely on token prediction [5]. Instead, the model is instructed to generate a few search terms as part of its output, which can be used to find more information about the event. Although this approach requires additional effort from the user to verify information, it is the most straightforward method for ensuring factual accuracy when needed.

To evaluate the effectiveness of these prompts, a selection of example responses is presented.

```
Statement added between 2019 and 2021:
{'subjectLabel': 'Boston', 'predicateLabel': 'population', 'objectLabel': '+692600'}

Generation duration:
5.630328115s

Response:
Here is my process:
1. I receive the input in the format (Change_type: ... ; Statement: {...} ; Events: [[event_code, event], ...]) which matches the problem description.
2. Analyzing the statement and the events: The statement is related to the population of Boston. Events are connected to various happenings in Boston, such as a train derailment, David Ortiz's return, statues being beheaded, COVID-19 restrictions, a new SARS-CoV-2 variant, and protests.
3. Concentrating on events related to the meaning of the statement (population), I consider if any event directly caused to changes in population. A train derailment might cause temporary disruptions but is unlikely to directly affect population numbers. David Ortiz's return, statues being beheaded, COVID-19 restrictions, a new SARS-CoV-2 variant, and protests do not seem to have a direct causal relationship with the population. However, one event catches my attention: [1004, Massachusetts reports their first case of a more contagious variant of SARS-CoV-2 in a woman in her 20s from Boston who traveled to the United Kingdom and began experiencing symptoms the day after she returned.] While this event might not directly cause an increase or decrease in population, it is related to health concerns that could potentially affect the population indirectly (e.g., through illness or death). However, given the information  or relates provided, I will choose a more conservative approach.
4. At the end of my process, no direct causation between events and the change in population can be established with certainty. Considering this analysis, the correct answer is: *-1*
```

Listing 6: LLM Output When No Event Caused Statement Change

In the example presented in listing 6, the model follows the instructed process comprehensively. During the process it evaluates several events to decide whether one of them could be responsible for the statement change. As it found no event could be directly linked as a cause for the statement addition, it ultimately correctly answer and responded with -1.

```
Statement added between 2019 and 2021:
{'subjectLabel': 'Donald Trump', 'predicateLabel': '
   participant in', 'objectLabel': 'Trump-Ukraine scandal'}

Generation duration:
3.412850427s

Response:
Here's my process:
1. I receive the input in the format: Change_type: added or
   removed; Statement: Wikidata statement; Events: list of
   events connected to the subject. In this case, I have:
   Change_type: added Statement: {subject: Q22686,
   subjectLabel: Donald Trump, predicate: P1344,
   predicateLabel: participant in, object: Q68451938,
   objectLabel: TrumpUkraine scandal} Events: [[... list of
   21 events ...]]
2. Analyzing the meaning of the statement and the events. The
    statement is about Donald Trump participating in the
   Trump-Ukraine scandal.
3. Identifying if any event directly caused the statement.
   Looking through the events, I see one that might be
   relevant: [[1157, The inspector general of Intelligence,
   Michael Atkinson, notifies the House Intelligence
   Committee about an urgent and credible whistleblower
   complaint involving an apparent July 25 telephone call in
   which President Donald Trump promised Ukrainian president
   Volodymyr Zelensky 250 million if he would reopen an
   investigation into Hunter Biden son of former Vice
   President Joe Biden. The White House denies doing anything
    wrong and refuses to release the complaint.]] This event
   is related to a phone call between Trump and Zelensky,
   which might be connected to the Trump-Ukraine scandal.
4. Checking if all necessary information is provided and no
   assumptions are needed. In this case, yes, I have enough
   information to make an educated guess that this event
   might cause the statement.
5. Returning the event code as my final answer. *1157*
```

Listing 7: LLM Output When an Event Caused Statement Change

Listing 7 demonstrates that in case one of the provided events is in fact responsible for the statement change, the LLM is able to identify it using its reasoning and correctly choose the index of the event.

```
Statement added between 2019 and 2021:
{'subjectLabel': 'Douglas Adams', 'predicateLabel': 'award
   received', 'objectLabel': 'Inkpot Award'}

Generation duration:
0.915944626s

Response:
Douglas Adams was posthumously awarded the Inkpot Award, a
   recognition of his contributions to comic art and science
   fiction, particularly for "The Hitchhiker's Guide to the
   Galaxy". This award is given to individuals who have
   significantly contributed to the field of science fiction,
    and Adams' work had become widely popular and influential
    since his passing.
[Douglas Adams, Inkpot Award, The Hitchhikers Guide to the
   Galaxy]
```

Listing 8: LLM Output Without Any Events Available

The example in listing 8 illustrates that even in the absence of any provided events, the model is capable of generating a plausible event that can be further researched to assess its relevance to the statement change. In this instance, while the model identifies a related event, it incorrectly attributes the award to Douglas Adams posthumously when, in fact, it was awarded in 1983. This mistake likely occurred because the time period during which the statement change happened was provided to the model, which is important for identifying causally linked events. On the other hand, it can lead the model to mistakenly attribute historical events to the specified period in an attempt to justify the statement change during this period. It is important to recognize that changes to Wikidata may include historical data added at any time, meaning not all changes are directly tied to contemporary events.

```
Statement removed between 2023 and the time of writing:
{'subjectLabel': 'Boston', 'predicateLabel': 'postal code', '
   object': '2127'}

Generation duration:
0.588882325s

Response:
The removal of the statement that Boston's postal code is
   2127 suggests a routine update or normalization of city
   postal codes. This change likely occurred as part of an
   effort to synchronize Wikidata with current postal coding
   systems.
```

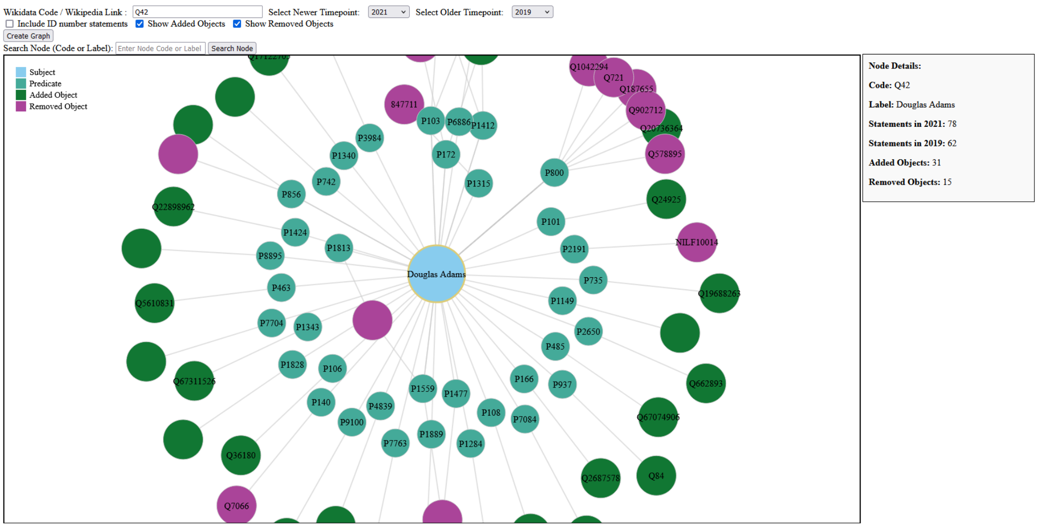

Figure 4: Visualization Tool Showcase

```
[Boston postal code, Postal code standardization]
```

Listing 9: LLM Output When Statement Change Is Typo Fix

Finally, the example in listing 9 showcases that the model can handle cases where the statement change clearly involves a simple correction, such as fixing a typo or removing inaccurate information, rather than being linked to an event.

3.4 Visualization Implementation

For the visualization implementation, JavaScript was selected as the programming language, utilizing the D3 library. As previously stated, the visualization was designed as an interactive webgraph. A screenshot of the visualization tool is provided in figure 4 to discuss further implementation details.

Users interact with the tool by providing either the Wikidata Entity ID or a Wikipedia resource URI. After specifying the two time points for comparison, users have the option to include IDs or not. As demonstrated in table 3, the inclusion of IDs significantly increases the graph's clutter, and it is generally recommended to exclude these statements as they hold limited analytical value.

Once a graph is generated, it centers on the subject node, which serves as the parent node for all statements. The nodes are connected to the subject via their respective predicate and color coded according to their change type.

Given the potential high number of nodes of the graph, not all nodes can be always displayed simultaneously. However, the interactive nature of the webgraph allows users to reposition and explore the graph as needed. At the top of the graph, additional filtering options are provided, allowing users to hide specific change types to view only additions or removals and to search for specific nodes of interest within the graph.

Clicking on any of the nodes displays the details of that node in the side box. If it is the subject node, it displays some basic statistics regarding the entity in general, such as the total number of statements in each of the time points as well as a total number of added and removed objects. Predicate nodes show all of their respective child objects categorized into added and removed objects as well as the option to hide the predicate node along with its connected objects. This feature was implemented to address the issue of certain predicates containing a substantial number of statement changes, which could obstruct the readability of the graph. By hiding irrelevant predicates, users can create a more comprehensible visualization. Object nodes display information about the event selected by the LLM model as the causal event for the object change along with the time of the event and a source. Alternatively, if no event was found to cause the change from the provided list of events, the LLM generates an explanation along with search terms.

To ensure accessibility for all users, including those with color blindness, a color scheme that accommodates all types of color vision deficiencies was adopted. The scheme ultimately selected was based on the color palettes developed by Paul Tol [53].

4 Discussion and Results

With the methods described, attention is now turned to demonstrating how the visualization tool can be utilized to gain deeper insights into the evolution of any Wikidata entity. To showcase a possible use of our tool, let us start by inspecting figure 5. In the figure, it can be seen that between 2019 and 2021, a significant number of statements about the postal code of Boston were added. Upon inspecting the objects of these statements, some discrepancies can be observed. Mainly, that some of the postal codes seem to be duplicated, with the only difference between them being a leading zero. A quick search reveals that the official postal codes of Boston are 02108-02137, 02163, 02196, 02199, 02201, 02203-02206, 02210-02212, 02215, 02217, 02222, 02126, 02228, 02241, 02266, 02283-02284, 02293, 02295, 02297-02298, 02467; meaning they all include a leading zero. This could have happened since Wikidata can be concurrently edited by multiple people, all with a different format in mind.

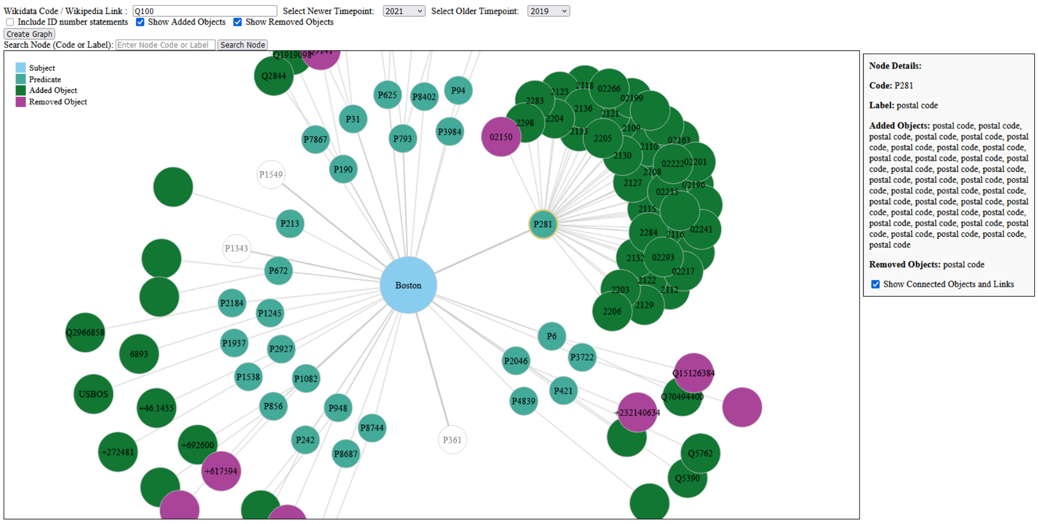

Figure 5: Changes Made to the Entity "Boston" Between 2019 and 2021

For example, a lot of spreadsheet editors such as Excel remove leading zeroes from numbers if their cells are not formatted properly. This discrepancy in the data could cause confusion and problems, since suddenly there are many more postal codes for Boston then there should be.

Let us go more into the future to see how these statements evolve. Figure 6 shows the changes that happened between 2023 and the time of writing. It is clear that all of the postal codes without a leading zero have been removed. This means someone probably noticed the incorrect formatting of the postal codes and removed them. Our tool shows that it took at least 2 years to notice and fix this issue. If even more time points were used, the time of change could be pinpointed more accurately.

Inspecting the LLM explanations for these changes in figure 7, shows similar information to what was mentioned before, such as the statements being removed due to bad formatting. This would help a less tech savvy user better understand the rationale behind these changes and give them a deeper understanding of not just the entity of Boston, but the editing process of Wikidata.

In the other example, lets say a person is not aware of any events connected to Elon Musk and wants to gain a more in-depth understanding of his past. They create a webgraph of changes between 2021 and 2023 which can be seen in figure 8. Among these changes, they notice that multiple objects were added to the property "owner of" (P183), one of them being Twitter (Q918).

But since the hypothetical user does not know anything about Elon Musk,

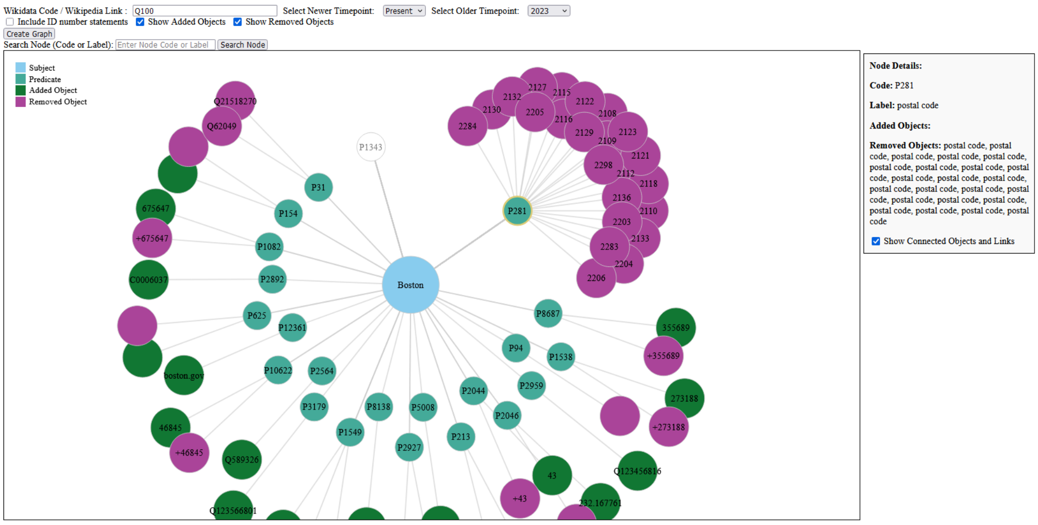

Figure 6: Changes Made to the Entity "Boston" Between 2023 and Time of Writing

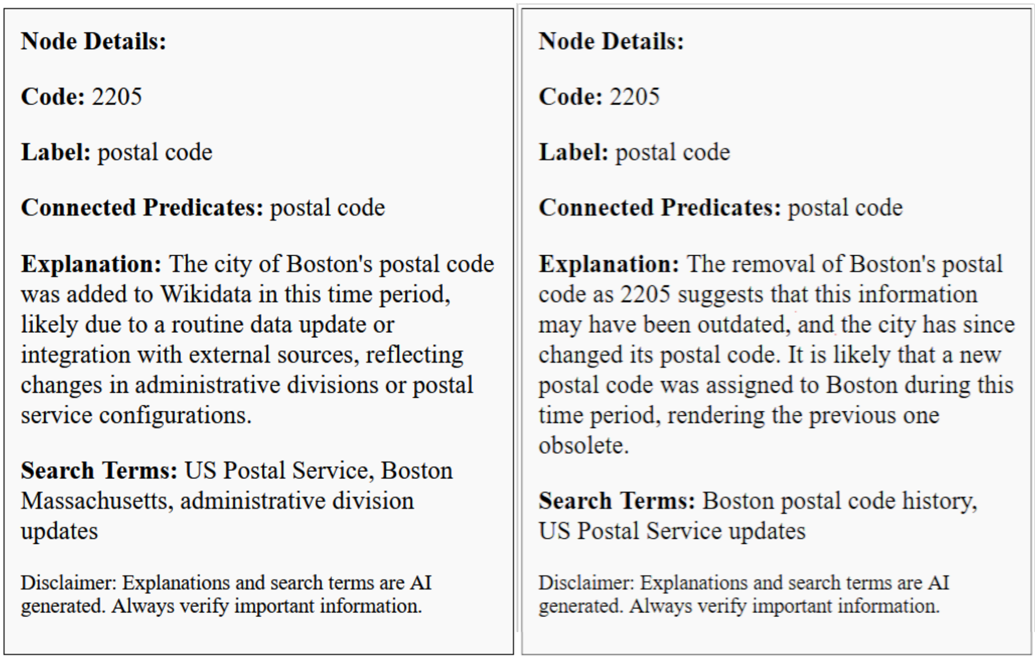

Figure 7: Node Details of Boston's "postal code" Changes

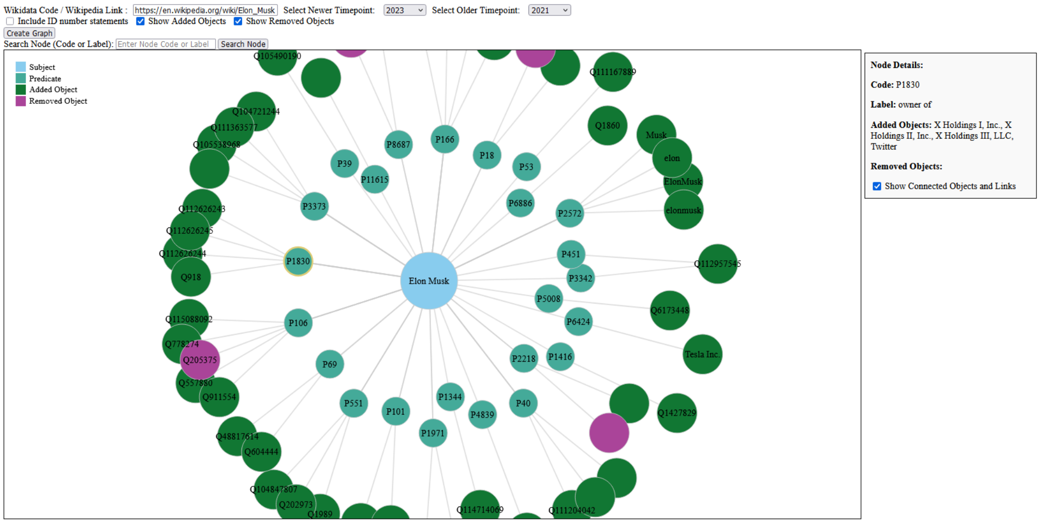

Figure 8: Changes Made to the Entity "Elon Musk" Between 2021 and 2023

they might be confused as to why this change happened. Inspecting the node details in figure 9 gives information about the event that may have caused this change to happen. In the explanation, even more details are provided due to the LLM having access to the necessary context of the event. Since the source of this event is reputable, concerns about its factuality are minimal. Even still, if additional information were needed, the specific event or relevant search terms to investigate further are readily available. Information about when the event took place is also provided, allowing for better pinpointing of how the change followed the event itself. This ensures that even an uninformed user is equipped with the additional necessary context for a statement change on Wikidata.

These are just some cases where our tool was useful for an in-depth inspection of the evolution of Wikidata entities. They showcase how our approach is able to present interesting insights into the editing process of Wikidata to any type of user, thanks to its clear and accessible webgraph representation. However, due to a lack of multiple time points as well as lacking event data when it comes to entities, such as individuals from non-english speaking countries, our research has not been able to fully link real-world events to statement changes. As it stand right now, if more context or additional information is needed, the LLM explanations can serve to provide a causal event for the statement change, or at least search terms to aid in finding more information about what could have caused the change to happen.

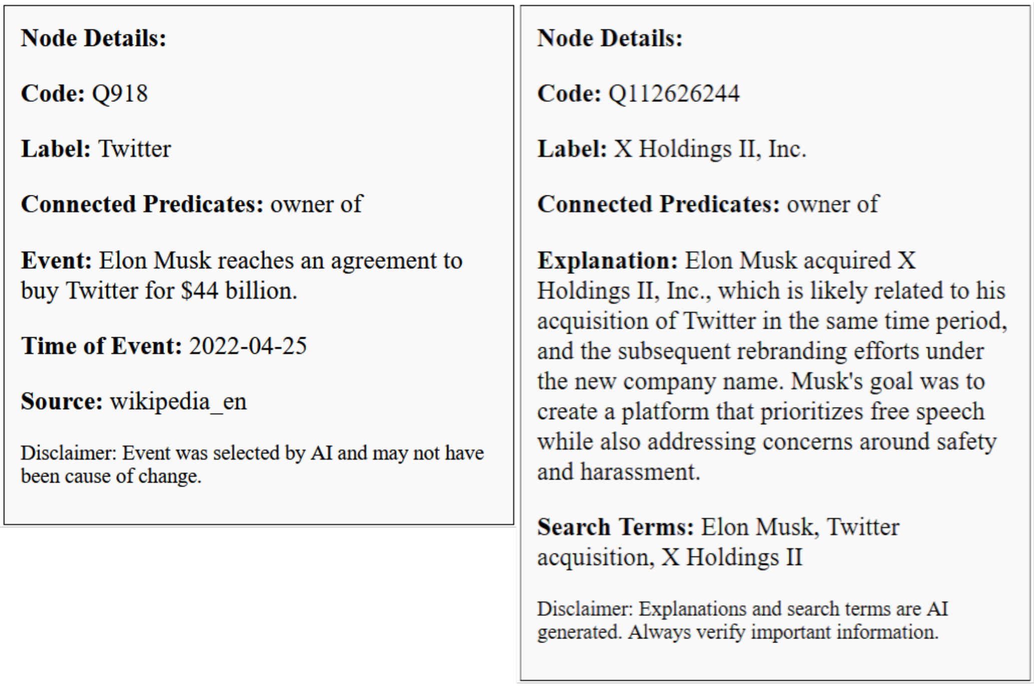

Figure 9: Node Details of Elon Musk's "owner of" Changes

4.1 Revisiting Research Questions

This thesis, while primarily a proof-of-concept, has demonstrated significant potential in addressing the stated goals. The analysis of temporal changes in Wikidata through the lens of a webgraph has introduced a method for enabling researchers from various disciplines to explore the historical evolution of entities. By employing this approach, researchers can gain deeper insights into how collaborative knowledge graphs reflect shifts in collective perception and the dynamic processes underlying the evolution of entities.

In terms of linking entity changes to real-world events, our findings suggest promising potential, though further work is required. To fully isolate and understand the effects of events on entity changes, it is essential to incorporate a greater variety of events and additional temporal data points. Despite these limitations, our research has successfully identified connections between specific events and corresponding changes in Wikidata entities. While some of these connections may appear intuitive, examining whether these changes occur immediately following an event or independently, provides valuable insights. Furthermore, the nature of these links may vary across domains. For instance, political entities are often updated promptly in response to real-world events, whereas entities related to scientific definitions may follow a less predictable and more gradual pattern. These distinctions highlight the need for domain-specific methodologies and deeper investigation to extract concrete conclusions, yet our approach provides a foundational framework for such explorations.

Our integration of LLMs to associate entity changes with real-world events has also proven effective. While there remain areas for improvement, particularly in ensuring consistency and scalability, the model performs well within the current scope. This capability positions our tool as a valuable resource for researchers by providing an informed starting point for investigating potential causal relationships. Although occasional inaccuracies may occur, the necessity of user validation ensures that the model serves as an efficient aid rather than a definitive solution. By streamlining the process of contextual exploration and identifying plausible connections, the model adds significant value to the overall framework. Moreover, as advancements in LLM technology continue, their capabilities are expected to expand, further enhancing their role in facilitating this type of research.

4.2 Limitations and Future Work

In this thesis, only 4 time points of Wikidata were used. By adding more time points, it would be possible to better isolate the effect of events on Wikidata changes even further, by comparing monthly time points within a given year. This would lead to interesting insights about the influence of events on Wikidata changes. Additionally, it could be of interest to dive deeper into the data stored in Wikidata, and analyze not just the main statement triples but also the qualifiers associated with these statements. These qualifiers include interesting information, such as the sources of the information. Analyzing the additions and removals of qualifiers could deepen our knowledge of entity changes even further.

When it comes to gathering of event data, there are other sources available besides EventKG that would perhaps be more suitable, since EventKG may occasionally provide an imprecise time point for an event. An alternative event database is the GDELT project [25] which is very detailed and robust, even though it only deals with political and national events. It gathers its event information from news articles, so every event has a clear source and time point associated with it and can be easily researched. However, GDELT results are also too large for prompting LLMs. In the future, using the GDELT database together with a filter or method, which would extract only the most influential events, could be a solution.

A method to deal with too many events would also be useful to add to the current approach with EventKG, as some entities have far too many events associated with them, mainly politicians or big countries such as the USA. This large amount of event data leads to the LLM model ignoring the system prompt or instructions in general as it gets overwhelmed with the large amount of text. To solve this problem, a new prompt was written which would reduce the number of events based on importance. However, this prompt still struggled when the number of events was in the hundreds, and instead of analyzing the events would select just the last events in the list. A possible fix for this issue would be executing the prompt in batches rather than all at once and then connecting all the batches together. This would lead to a new problem of a long processing time, as each response generation takes a couple of seconds, so doing multiple batches of generation on top of the statement change generation could lead to being just way too slow. These issues could be addressed with the improvements of open LLM models in general. At the time of writing Llama 3.2 has already been released boasting a larger 11B parameters as well as outperforming Llama 3.1 in all benchmarks. There are also larger parameter model versions available, which show better benchmark performance, so if the necessary space to run these models is available, they would also provide better results. Additionally, our prompts could be even further refined, or more fail-safes introduced, such as recursive generation, where a failed or undesirable generation is repeated until the desired outcome is reached. It is safe to say that as time goes on, the technology of LLMs will get better and better and with this so will the accuracy and capability of their use for our purposes.

Finally, a lot of interesting insight could be gained by using the methods and tools presented in this thesis to study the evolution of different entities. Perhaps one could focus on a certain class of entities or subject and do an in depth analysis of the changes and the rationale behind them. Through this, maybe some interesting patterns could be found which would be very helpful for understanding the chosen subject, as well as the process of Wikidata editing as a whole.

5 Conclusion

This thesis presented a comprehensive approach to developing a tool for visualizing changes in Wikidata statements. The proposed tool employs an interactive web-based graph design, providing an intuitive and accessible visualization of statement additions and removals, even for users with minimal prior experience. Furthermore, it introduced a novel method for linking real-world events to individual statement changes using an LLM. An earlier version of this work has already been presented at the International Semantic Web Conference 2024 (ISWC) during the Retrieval-Augmented Generation Enabled by Knowledge Graphs (RAGE-KG) workshop [54].This proof-of-concept demonstrates significant potential for analyzing the evolution of Wikidata entities while uncovering their causal drivers.

This thesis contributes a robust foundation for future investigations into the dynamics of collaborative knowledge graphs. By combining webgraph visualizations with real-world event analysis, our approach enables researchers to examine the drivers of entity changes over time and explore how interest in entities shifts in relation to external events. With continued refinement and broader application, this framework has the potential to unlock new avenues for understanding the interplay between collaborative knowledge systems and real-world developments.

Overall, our approach highlights the possibilities of integrating advanced visualization techniques with event-driven explanations to enhance understanding of collaborative knowledge graphs. By scaling this methodology to include additional temporal data points and expanding the range of event data collected, it could offer valuable insights into the dynamics of knowledge graph development and the interplay between real-world events and digital knowledge ecosystems. This work serves as a foundation for further exploration into these interconnections and the broader implications for managing and understanding dynamic, collaborative knowledge bases.

References

[1] Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan Reutter, and Domagoj Vrgoč. Foundations of modern query languages for graph databases. *ACM Computing Surveys (CSUR)*, 50(5):1–40, 2017.

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

[3] Aras Bozkurt and Ramesh C Sharma. Generative ai and prompt engineering: The art of whispering to let the genie out of the algorithmic world. *Asian Journal of Distance Education*, 18(2):i–vii, 2023.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*, 2023.

[6] Andrew Chisholm, Will Radford, and Ben Hachey. Learning to generate one-sentence biographies from wikidata. *arXiv preprint arXiv:1702.06235*, 2017.

[7] Kornpol Chung, Romana Pernisch, and Stefan Schlobach. Descriptive comparison of visual ontology change summarisation methods. In *European Semantic Web Conference*, pages 54–58. Springer, 2023.

[8] Anthony Colas, Ali Sadeghian, Yue Wang, and Daisy Zhe Wang. Eventnarrative: A large-scale event-centric dataset for knowledge graph-to-text generation. *arXiv preprint arXiv:2111.00276*, 2021.

[9] Richard Cyganiak, David Wood, Markus Lanthaler, Graham Klyne, Jeremy J Carroll, and Brian McBride. Rdf 1.1 concepts and abstract syntax. *W3C recommendation*, 25(02):1–22, 2014.

[10] Xiao Ding, Zhongyang Li, Ting Liu, and Kuo Liao. Elg: an event logic graph. *arXiv preprint arXiv:1907.08015*, 2019.

[11] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[12] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. Introducing wikidata to the linked data web. In *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13*, pages 50–65. Springer, 2014.

[13] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web*, 9(1):77–129, 2018.

[14] Simon Gottschalk and Elena Demidova. Eventkg: A multilingual event-centric temporal knowledge graph. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 272–287. Springer, 2018.

[15] Simon Gottschalk and Elena Demidova. Eventkg–the hub of event knowledge on the web–and biographical timeline generation. *Semantic Web*, 10(6):1039–1070, 2019.

[16] Simon Gottschalk and Elena Demidova. Eventkg statistics. https://eventkg.l3s.uni-hannover.de/, 2019. Accessed: 2025-01-13.

[17] 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.

[18] Thomas F Heston and Charya Khun. Prompt engineering in medical education. *International Medical Education*, 2(3):198–205, 2023.

[19] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard De Melo, and Gerhard Weikum. Yago2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, pages 229–232, 2011.

[20] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

[21] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[22] Yuxi Jin and Shun Shiramatsu. Multilingual complementation of causality property on wikidata based on gpt-3. In *Proceedings of Seventh International Congress on Information and Communication Technology: ICICT 2022, London, Volume 3*, pages 573–580. Springer, 2022.

[23] Nicola Pascal Krenn. Towards analysing the evolution of community-driven (sub-) schemas within wikidata. 2023.

[24] André Filipe Amorim Lara. Visualization of ontology evolution using ontodiffgraph. Master's thesis, Universidade do Minho (Portugal), 2018.

[25] Kalev Leetaru and Philip A Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer, 2013.

[26] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

[27] Zhongyang Li, Sendong Zhao, Xiao Ding, and Ting Liu. Eeg: knowledge base for event evolutionary principles and patterns. In *Social Media Processing: 6th National Conference, SMP 2017, Beijing, China, September 14-17, 2017, Proceedings*, pages 40–52. Springer, 2017.

[28] Quentin Malartic, Nilabhra Roy Chowdhury, Ruxandra Cojocaru, Mugariya Farooq, Giulia Campesan, Yasser Abdelaziz Dahou Djilali, Sanath Narayan, Ankit Singh, Maksim Velikanov, Basma El Amel Boussaha, et al. Falcon2-11b technical report. *arXiv preprint arXiv:2407.14885*, 2024.

[29] Sabrina J Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y Lee, Benoît Sagot, et al. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. *arXiv preprint arXiv:2112.10508*, 2021.

[30] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40, 2023.

[31] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

[32] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. Industry-scale knowledge graphs: Lessons and challenges: Five diverse technology companies show how it's done. *Queue*, 17(2):48–75, 2019.

[33] Jere Odell, Mairelys Lemus-Rojas, and Lucille Brys. Wikidata data model. *Wikidata for Scholarly Communication Librarianship*, 2022.

[34] Katherine Panciera, Aaron Halfaker, and Loren Terveen. Wikipedians are born, not made: a study of power editors on wikipedia. In *Proceedings of the 2009 ACM International Conference on Supporting Group Work*, pages 51–60, 2009.

[35] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great migration. In *Proceedings of the 25th international conference on world wide web*, pages 1419–1428, 2016.

[36] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. *ACM Transactions on Database Systems (TODS)*, 34(3):1–45, 2009.

[37] Romana Pernisch, Daniele Dell'Aglio, Mirko Serbak, Rafael S Gonçalves, and Abraham Bernstein. Visualising the effects of ontology changes and studying their understanding with chimp. *Journal of Web Semantics*, 74:100715, 2022.

[38] Romana Pernisch, Daniëlle Dijkstra, and Stefan Schlobach. Visualisation of ontology changes and evolution: A systematic literature review. *Semantic Web Journal*, 2024.

[39] Alessandro Piscopo and Elena Simperl. What we talk about when we talk about wikidata quality: a literature survey. In *Proceedings of the 15th International Symposium on Open Collaboration*, pages 1–11, 2019.

[40] Axel Polleres, Romana Pernisch, Angela Bonifati, Daniele Dell'Aglio, Daniil Dobriy, Stefania Dumbrava, Lorena Etcheverry, Nicolas Ferranti, Katja Hose, Ernesto Jiménez-Ruiz, et al. How does knowledge evolve in open knowledge graphs? *Transactions on Graph Data and Knowledge*, 1(1):11–1, 2023.

[41] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI*, 2018.

[42] Miguel Ángel Rodrıguez-Garcıa, Luke Slater, Keiron O'Shea, Paul N Schofield, Georgios V Gkoutos, and Robert Hoehndorf. Visualizing ontologies with aberowl. *Semantic Web Applications and Tools for the Life Sciences*, pages 183–192, 2015.

[43] Marco Rospocher, Marieke Van Erp, Piek Vossen, Antske Fokkens, Itziar Aldabe, German Rigau, Aitor Soroa, Thomas Ploeger, and Tessel Bogaard. Building event-centric knowledge graphs from news. *Journal of Web Semantics*, 37:132–151, 2016.

[44] Charlotte Rudnik, Thibault Ehrhart, Olivier Ferret, Denis Teyssou, Raphaël Troncy, and Xavier Tannier. Searching news articles using an event knowledge graph leveraged by wikidata. In *Companion proceedings of the 2019 world wide web conference*, pages 1232–1239, 2019.

[45] Tomás Sáez and Aidan Hogan. Automatically generating wikipedia infoboxes from wikidata. In *Companion Proceedings of the The Web Conference 2018*, pages 1823–1830, 2018.

[46] Cristina Sarasua, Alessandro Checco, Gianluca Demartini, Djellel Difallah, Michael Feldman, and Lydia Pintscher. The evolution of power and standard wikidata editors: comparing editing behavior over time to predict lifespan and volume of edits. *Computer Supported Cooperative Work (CSCW)*, 28:843–882, 2019.

[47] Lukas Schmelzeisen, Corina Dima, and Steffen Staab. Wikidated 1.0: An evolving knowledge graph dataset of wikidata's revision history. *arXiv preprint arXiv:2112.05003*, 2021.

[48] Chris Stokel-Walker and Richard Van Noorden. What chatgpt and generative ai mean for science. *Nature*, 614(7947):214–216, 2023.

[49] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[50] Wikimedia Team. Wikipedia current events portal. `https://en.wikipedia.org/wiki/Portal:Current_events`, 2006. Accessed: 2024-11-30.

[51] Wikimedia Team. Wikipedia lists of events. `https://en.wikipedia.org/wiki/Category:Lists_of_events`, 2009. Accessed: 2024-11-30.

[52] Wikimedia Statistics Team. Wikidata statistics. `https://stats.wikimedia.org/#/wikidata.org`, 2017. Accessed: 2024-11-30.

[53] Paul Tol. Colour schemes. *SRON Technical Note*, 2, 2012.

[54] Gregor Vandák and Amin Anjomshoaa. Leveraging large language models to identify event-driven changes in wikidata entities. 2024.

[55] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[56] Juan David Velásquez-Henao, Carlos Jaime Franco-Cardona, and Lorena Cadavid-Higuita. Prompt engineering: a methodology for optimizing interactions with ai-language models in the field of engineering. *Dyna*, 90(230):9–17, 2023.

[57] Denny Vrandečić. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st international conference on world wide web*, pages 1063–1064, 2012.

[58] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[59] Denny Vrandečić, Lydia Pintscher, and Markus Krötzsch. Wikidata: The making of. In *Companion Proceedings of the ACM Web Conference 2023*, pages 615–624, 2023.

[60] Jie Wu, Xinning Zhu, Chunhong Zhang, and Zheng Hu. Event-centric tourism knowledge graph-a case study of hainan. In *Knowledge Science, Engineering and Management: 13th International Conference, KSEM 2020, Hangzhou, China, August 28–30, 2020, Proceedings, Part I 13*, pages 3–15. Springer, 2020.

[61] Chengbiao Yang, Weizhuo Li, Xiaoping Zhang, Runshun Zhang, and Guilin Qi. A temporal semantic search system for traditional chinese medicine based on temporal knowledge graphs. In *Semantic Technology: 9th Joint International Conference, JIST 2019, Hangzhou, China, November 25–27, 2019, Revised Selected Papers 9*, pages 13–20. Springer, 2020.

[62] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Soeren Auer. Quality assessment for linked data: A survey. *Semantic Web*, 7(1):63–93, 2016.

[63] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[64] Lihong Zhu, Amanda Xu, Sai Deng, Greta Heng, and Xiaoli Li. Entity management using wikidata for cultural heritage information. *Cataloging & Classification Quarterly*, 61(1):20–46, 2023.