Leveraging Large Language Models to Identify Event-Driven Changes in Wikidata Entities

Gregor Vandák, Amin Anjomshoaa

Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria
Institute for Data, Process and Knowledge Management

Abstract

Wikidata undergoes continuous updates from a diverse community of users. This paper explores the work in progress on the application of Large Language Models (LLMs) to establish connections between these entity modifications and real-world events sourced from open databases. By presenting the LLM with a customized prompt alongside relevant events linked to the entity, we instruct the model to identify the event most likely responsible for the observed change. This approach offers causal explanations for entity updates and enriches the contextual understanding of the factors driving changes within a collaboratively edited knowledge base. Ultimately, this research aims to contribute to a deeper understanding of the dynamics that shape the evolution of crowdsourced knowledge bases such as Wikidata.

Keywords

Wikidata, Linked Open Data, Large Language Model, Event Knowledge Graph

1. Introduction

Wikidata is a collaboratively edited knowledge base operated by the Wikimedia Foundation, designed to serve as a structured data repository supporting Wikipedia and other Wikimedia projects [1]. Launched in 2012, it aims to centralize the management of factual data, ensuring consistency and reducing redundancy across various Wikimedia platforms [2]. Users can create and edit data items representing real-world entities such as people and places. These items are described using triple statements, which consist of property-value pairs that articulate the relationships and attributes of the entities [3]. This structure enables the integration of diverse information across different languages and disciplines, promoting a multilingual and universal knowledge ecosystem [4]. All of this information is accessible 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 [5] to improving automatic text generation [6, 7].

Much like other Wikimedia projects, Wikidata aims to provide factual data, and thus needs to be updated constantly. This means constant changes coming from various sources including not just fixes but also new data, with new connections being made and extra details provided for each entity [8]. To this end, earlier literature surveys indicate that the timing of these changes and the speed at which new information is added are among the less-researched aspects concerning the quality of Wikidata [9].

Still there have been efforts to delve deeper into the evolution of Wikidata. For example in *Wikidated 1.0* knowledge graph dataset, the authors develop an approach to create an evolving knowledge graph of Wikidata based on its revision history [10]. C. Sarusa et. al. analysed the editing behaviour of individual types of users, leading to very impressive insights [11]. While these are very compelling and helpful for further contributions, they do not directly tackle the issue of finding the source as well as the timeliness of changes on Wikidata. In this paper we propose a method similar to the one described by Y. Jin and S. Shiramatsu [12] to create causal relations between events and entity changes [13]. In doing so, we aim to get further details into what drives entity changes and explore how current events associated with

International Semantic Web Conference - RAGE-KG Workshop, November 11–12, 2024, Baltimore, Maryland, USA
gregor.vandak@gmail.com (G. Vandák); amin.anjomshoaa@wu.ac.at (A. Anjomshoaa)
https://linkedin.com/in/gregor-vandak-935884321/ (G. Vandák); https://wu.ac.at/dpkm/team/anjomshoaa (A. Anjomshoaa)
0000-0001-6277-742X (A. Anjomshoaa)

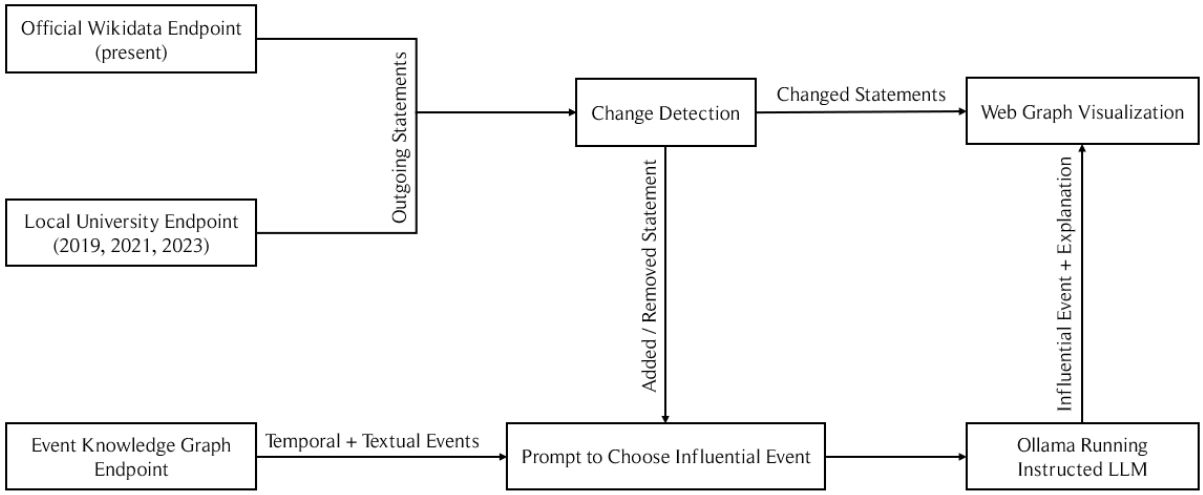

Figure 1: Diagram depicting the workflow

an entity influence updates to the corresponding Wikidata statements. We also explore the insights that can be gained by understanding the reasons behind these changes, focusing on the shifting priorities of community efforts and developing a deeper comprehension of the entity itself. Ultimately, we would like to use our approach in a visualization tool, as to provide people with valuable context and details into the evolution of Wikidata entities.

2. Methodology

To effectively trace the connections between Wikidata entity changes and real-world events, we developed a systematic approach that leverages the capabilities of Large Language Models (LLMs). This methodology involves integrating structured data from Wikidata with event data from open databases, enabling the model to identify potential causal relationships. To achieve this, we utilize the live Wikidata SPARQL endpoint along with multiple older endpoints to track entity changes over time. In order to identify the changes between two time points, we query the corresponding SPARQL endpoints for all entity statements and compare the results. This way we can see which statements were added and which removed in the selected time period. In parallel, we query an open database which focuses on events to get all events connected to the entity in the given time period. Finally, we prompt a selected LLM to identify which of the given events could have caused the identified changes in the entity statements. Figure 1 depicts our proposed workflow and the core components of our approach.

2.1. Wikidata & Event Knowledge Graph

To be able to get the statements of the entity for any given time, we use our historical Wikidata SPARQL endpoint that are publicly available. These endpoints are created based on the historical versions of the Wikidata which are available for download by Wikidata team. For the purposes of this paper, we will be using the 2019, 2021, 2023, and the live versions of Wikidata. With the endpoints set up, we can query the database. The following is the sample query we will use in this paper, with the example entity of *Donald Trump*, represented with the Wikidata QID "Q22686":

```
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:Q22686 AS ?subject).
```

Table 1
Example outputs of outgoing statements of Donald Trump at 2021

Subject	SubjectLabel	Predicate	PredicateLabel	Object	ObjectLabel
Q22686	Donald Trump	P101	field of work	Q7163	politics
Q22686	Donald Trump	P101	field of work	Q7188	government
Q22686	Donald Trump	P102	member of political party	Q29468	Republican Party

```
#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: Example Wikidata query

This query extracts all the outgoing statements, which means that our entity of interest acts as the subject of the statement triple. It also extracts the labels of the triple in English, which is important to make the statement readable for both humans and LLMs. In Table 1 you can see a couple of lines from this query for the year 2021.

There can be various reasons for a change, such as correcting a typo, adding new information, or removing false data, making the classification of these changes quite challenging. In this paper, we follow a similar approach as described in *Wikidated 1.0* [10] and only classify statement additions and removals. To identify the changes that occurred between two time points, we can compare the statements of an entity at each time point to determine what has changed. Since the subject is fixed for all statements, we simply check whether the same predicate and object combination is present in both data sets. If a statement is present only in the newer dataset, it is classified as an addition; if it appears only in the older dataset, it is classified as a removal.

Now that we have our changes we can shift our focus to retrieving the corresponding events. To this end, we use the EventKG [1] for finding the relevant events. EventKG can be queried via its SPARQL endpoint. There are two types of events in EventKG: text events and temporal events. Temporal events are represented by a triple statement of subject, predicate and object while text events are more complex and thus have to be represented with a short paragraph of text. Along with the event description, we can also get the source of the event, which could be helpful for further research into a chosen topic. Finally, we also query for the begin and end time points of the event. However, these time points are often not precise (being just the start/end of a year) or are missing entirely. This makes our job of filtering for events that happened only in specific time period more difficult and we might miss some events due to the absence of time-related information. Despite that, in many cases we might still have enough events for explaining the statement changes, as the most influential and important events tend to have accurate time points.

2.2. Large Language Models (LLMs)

Regarding the LLM application in our use case, we have decided to stick with the open access theme, and use open LLM models through Ollama [2] framework which allows users to locally run a wide variety of state-of-the-art models. Additionally, it enables users to create their own model using the "ModelFile" command and providing an instruction prompt to existing models, leading to better results in specific tasks. In the context of our proposed approach, we have developed a specialized prompt to guide the LLM in the task of event identification. The designed prompt is as follows:

[1] https://eventkg.l3s.uni-hannover.de/
[2] https://ollama.com/

```
You are an expert assistant whose role it is to decide whether a change in a Wikidata statement has been caused by one of the given events. You will be provided with a statement with the structure of subject, predicate, object and whether it was added or removed. You will be also provided wtih two JSON files containing events related to the subject. If one of the given events caused the change, you must return that event and a brief explanation why it caused the change in a JSON format. Otherwise, you must return "No responsible event." in the same JSON format.
#INPUT_TEMPLATE:
[entity, predicate, object]; type_of_change; JSON_of_events1; JSON_of_events2
#OUTPUT_IF_TRUE_TEMPLATE#:
{event: event_name, explanation: write an explanation why the event caused the change}
#REMEMBER:
You must return only one event and only if it was a direct cause of of the change. You must not deviate from the output template. Do not return multiple events.
```

Listing 2: Instruction prompt for LLM

We decided for a semi-structured prompt style where we first describe the role of the model and then give further details, as we found in our experimentation as well as literature that this leads to more favorable results [14]. Additionally, we specify the output format to JSON which facilitates data integration and data processing in our workflow. We also reinforce the most important parts of the instruction prompt once more in the ending section to further reduce disobeying [15]. Finally, in the parameters of our model, we set a seed so repeating the same prompt results in the same results as well as set the output format to JSON.

We can move on to an example to see how effective our approach is. Let us say that we want to find an explanation for the following statement addition between the Wikidata 2019 and Wikidata 2021.

Donald Trump | participant in | Trump-Ukraine scandal

Next, we gather the textual and temporal events connected to *Donald Trump* between 2019 and 2021 using the EventKG endpoint. Lastly, we feed all our data to our model keeping in mind the aforementioned input template. Using llama3.1 along with our instructions we get the following output:

```
{
"event": "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",
"explanation": "This event likely caused the addition of Donald Trump as a participant in the Trump-Ukraine scandal, as it relates to the same phone call and controversy."
}
```

Listing 3: Example output

As we can see the LLM correctly chose the corresponding event as an explanation for the change. If desired, we can further enrich this event using EventKG to obtain additional details, such as the time of occurrence and the information source, which can be valuable for deeper analysis.

3. Conclusion & Future Work

In this paper we showcased the current state of our work on connecting Wikidata updates and changes to events using LLMs. We are currently checking the feasibility of this approach through multiple different LLM models and prompting methods. While our approach works pretty well for well known entities such as our selected example of *Donald Trump*, it can still struggle when faced with more obscure Wikidata entities. Moreover, the number of time points to compare together should be increased in the future so that we may better investigate the effect of events on Wikidata updates and changes. More years or even monthly time points would enable to further isolate the effect that singular events have on selected changes.

When it comes to gathering of event data, there are other sources available besides EventKG that would perhaps be more suitable, since EventKG has trouble with providing a specific time point for an event. Additionally, some highly cited entities can have too many relevant events, leading to problems when processing them through the LLM component. An alternative event database is the GDELT project [3] 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 back checked. However, GDELT results are also too large for processing through LLM. In the future, we would consider using the GDELT database together with a filter, which would extract only the most influential events.

We are currently refining the LLM prompt to enhance the results and experimenting with other state-of-the-art open models, such as Gemma2[4], Mistral [5], and Falcon2 [6], to identify the most effective one for our needs. The final choice of the model will also affect our approach to prompt creation and parameter selection. It should also be noted that LLMs tend to have a bias to them, which could affect our results. Given the rapid advancements in the LLM field, we are confident that the number of available open models will continue to grow, so we should remain vigilant in exploring new options.

Finally, we are planning to develop an interactive visualization tool designed to effectively illustrate the evolution of Wikidata entities. This tool will provide users with a user-friendly and accessible platform to track and analyze how specific events have influenced changes in Wikidata entities over a defined time period. It will also enable users to easily explore any Wikidata entity and gain valuable insights into its evolution over time.

References

[1] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Communications of the ACM 57 (2014) 78–85.

[2] D. Vrandečić, Wikidata: A new platform for collaborative data collection, in: Proceedings of the 21st international conference on world wide web, 2012, pp. 1063–1064.

[3] S. Guan, X. Cheng, L. Bai, F. Zhang, Z. Li, Y. Zeng, X. Jin, J. Guo, What is event knowledge graph: A survey, IEEE Transactions on Knowledge and Data Engineering (2022).

[4] L. Zhu, A. Xu, S. Deng, G. Heng, X. Li, Entity management using wikidata for cultural heritage information, Cataloging & Classification Quarterly 61 (2023) 20–46.

[5] C. Rudnik, T. Ehrhart, O. Ferret, D. Teyssou, R. Troncy, X. Tannier, Searching news articles using an event knowledge graph leveraged by wikidata, in: Companion proceedings of the 2019 world wide web conference, 2019, pp. 1232–1239.

[6] T. Sáez, A. Hogan, Automatically generating wikipedia info-boxes from wikidata, in: Companion Proceedings of the The Web Conference 2018, 2018, pp. 1823–1830.

[7] A. Chisholm, W. Radford, B. Hachey, Learning to generate one-sentence biographies from wikidata, arXiv preprint arXiv:1702.06235 (2017).

[8] K. Panciera, A. Halfaker, L. 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, 2009, pp. 51–60.

[9] A. Piscopo, E. Simperl, What we talk about when we talk about wikidata quality: a literature survey, in: Proceedings of the 15th International Symposium on Open Collaboration, 2019, pp. 1–11.

[10] L. Schmelzeisen, C. Dima, S. Staab, Wikidated 1.0: An evolving knowledge graph dataset of wikidata's revision history, arXiv preprint arXiv:2112.05003 (2021).

[11] C. Sarasua, A. Checco, G. Demartini, D. Difallah, M. Feldman, L. Pintscher, The evolution of

[3] https://www.gdeltproject.org/

[4] https://ai.google.dev/gemma

[5] https://mistral.ai/

[6] https://huggingface.co/tiiuae/falcon-11B

power and standard wikidata editors: comparing editing behavior over time to predict lifespan and volume of edits, Computer Supported Cooperative Work (CSCW) 28 (2019) 843–882.

[12] Y. Jin, S. 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, Springer, 2022, pp. 573–580.

[13] B. Drury, H. G. Oliveira, A. de Andrade Lopes, A survey of the extraction and applications of causal relations, Natural Language Engineering 28 (2022) 361–400.

[14] B. Chen, Z. Zhang, N. Langrené, S. Zhu, Unleashing the potential of prompt engineering in large language models: a comprehensive review, arXiv preprint arXiv:2310.14735 (2023).

[15] J. D. Velásquez-Henao, C. J. Franco-Cardona, L. Cadavid-Higuita, Prompt engineering: a methodology for optimizing interactions with ai-language models in the field of engineering, Dyna 90 (2023) 9–17.