Don’t Stop Thinking about Tomorrow: Use Cases Demonstrating the Asymmetric Impact of Contextual Temporal Links in Knowledge Graph Evolution & Retrieval[1]

Waterman, K. Krasnow[1][0000-1111-2222-3333]

[1] Decentralized Information Group, Computer Science and Artificial Intelligence Lab,
Massachusetts Institute of Technology 02139, USA
kkw@mit.edu

Abstract. This short paper presents use cases to prompt consideration of the asynchronous and asymmetric nature of context updates when devising schemes and standards for managing and preserving decentralized knowledge graphs. As data are increasingly connected in knowledge graphs that evidence the relationships among them, an open challenge is how to manage and preserve decentralized data so that a graph updates, and a query returns, data that correctly evidences the contextual relationship. Much of the focus on managing and preserving the evolution of data has been about preserving the internal (internal to a dataset or source) history, where preservation and retrieval are synchronous. But, as demonstrated here, in many real-world use cases the correct linkage and, therefore, preservation and retrieval, is neither a temporal match nor related version match.

Keywords: Knowledge graph, knowledge graph evolution, temporal nodes, temporal relationships, web standards, data management, data preservation, data context, context mapping, linked data, semantic web.

1 Introduction

Context tells us about the environment in which information exists. Context educates us as to how information is relevant, by its relation to other things - from the most common comparators of time and location to little known events that impact or are impacted by our data. From the inception of this workshop on Managing the Evolution and Preservation of the Data Web (“MEPDaW”), there has been a recognition of the importance of temporal relationships in the update, recording, storing, and retrieval of linked data [1][2][3]. More recent work has expanded on the abilities to work at scale and with ever increasing numbers of versions [4][5]. Overwhelmingly, work has operated on the presumption that linked data should be preserved or analyzed at a moment in time, past or present. But decentralized data providing context in an evolving graph is not necessarily a temporal match nor an exact versioning match. This paper provides use cases in which searching for historical versions of a knowledge graph will require the ability to identify and retrieve data which does not share the same archival date and/or requires the retrieval of more than one version of some but not all nodes, and possibly edges, of the graph.

2 Temporal Context & Graph Evolution

In the initial design of a knowledge graph, the relationships are often defined based upon knowledge or theory of a use case as a snapshot. As graphs are deployed in production, and the data flows, it becomes apparent that an additional sort of descriptor is required. What should be defined and where – when the impact of updates to temporal context on the evolution of the graph is known?

2.1 Simultaneous Context

Circumstances where a simultaneous set of facts provides context are perhaps the easiest to call to mind. For example, periods of rain readily provide the context for many traffic accidents [6]. In such a case, a decentralized graph might tie the exact time and geo-perimeter of meteorologic data about a phenomenon [7], highway data about the number of vehicles in the vicinity at that time from EZpass or traffic cam counts [8][9], and law enforcement data about accidents from published police reports [10]. In this case, it is straight-forward to retrieve the data by querying *event_date*. Even if, as so often occurs, smaller accidents are reported and entered on later days, retrieval of all the graph's data based on *event_date* will still be effective.

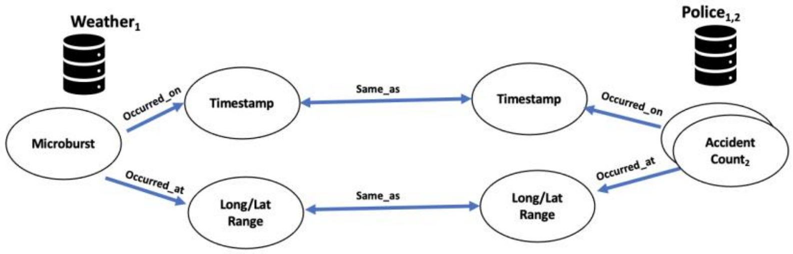

Fig. 1. Simultaneous Temporal Context: Retaining the temporal (& geospatial) edges when one decentralized node is updated.

2.2 Lagging Context

There are many circumstances in which there is a time lag between one set of facts which provides the context for another set of facts. A common example is the relationship between national testing scores of a local school and changes in house prices in the district [11]. In such a case, the testing scores are typically released and ranked once in a year [12][13][14], while house prices, averages, etc. are updated at least monthly by real estate companies and governmental agencies [15]. In these cases, the relevant temporal nodes share neither the same name nor date. For example, the relevant score date is not the *event_date* (regardless of whether that is defined as *test_date* or *scoring_date*), but the *pub_date* – the first date that the scores could have been known by others; the relevant price date is not the *pub_date* but arguably the *offer_date* – the first date there may be evidence of the market response; and the timing of the relationship begins at *pub_date+N* – the number of days after publication that it could have reached a real estate agent or a buyer. Statistics on either side of the graph can be corrected or updated for a particular date. For these cases, it is important to remember that, even though there is not an exact temporal match, the modification of either set should not break the graphed relationship. And, retrieval should be of the final corrected versions only.

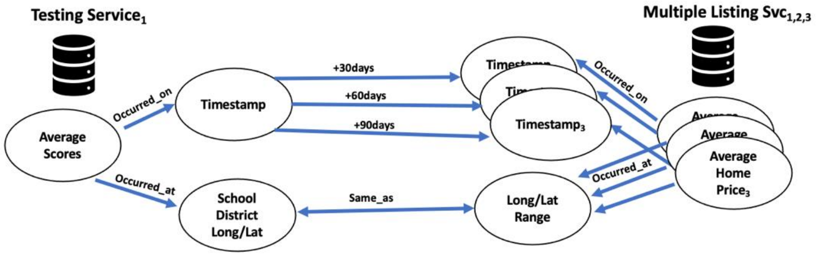

Fig. 2. Lagging Context: Temporal updates treated as new nodes, where one decentralized source provides context for later data from another source.

2.3 Predictive Context

Conversely, there are instances in which retrieval should pull all the iterations, not just the final. Consider this circumstance, where data in one store has a predictive link to data in another store. For example, over the summer of 2021, there was a record number of dogs surrendered to a local animal control agency and this appeared to be a predictor of the number of households to be in distressed circumstances at the end of Covid-related eviction moratoria. Figure 4 shows one possible graph in which the *Surrendered_Dog_Count* node is a separate and distinct daily report, but it causes only intermittent updates to the versioning of the singular node for *Updated_Evictions_Forecast*. It is possible that the iterations of the edges (and resulting node versions) may not be consistently temporally spaced, for reasons ranging from testing and refining the forecasting model to additional forecasting when there is a significant influx of dogs. What then is the appropriate query to restore the history?

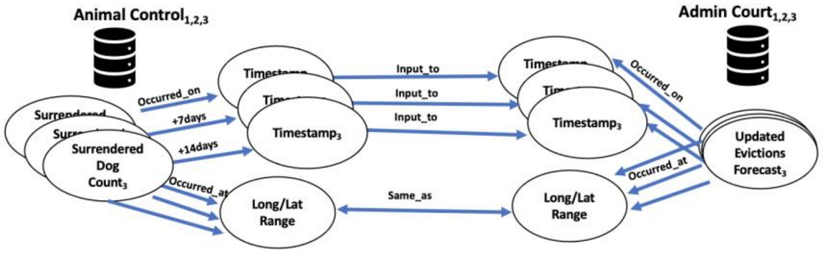

Fig. 3. Predictive Context: One decentralized source creates separate nodes for each update to support a single updated node where another source is calculating predictions.

2.4 Bi-directional Context

Another sort of context which would require the reevaluation of the relationship based upon knowledge at a particular time, is when the change can be prompted by any node. For example, consider advances in knowledge about human reactions to substances and changes in grocery contents. There may be new medical practice or research reporting – for example, the impact that Sucralose has on blood sugar [16] – which changes the graphed labels between diabetes and numerous foods. Sucralose was recently the leading ingredient globally for new foods and beverages with sugar-related claims [17], an example of the constant changes to the contents of groceries [18][19] – foods, toiletries, cleaning supplies – which can also change the nature of the label between an item and a medical condition (e.g., allergy, celiac, diabetes).

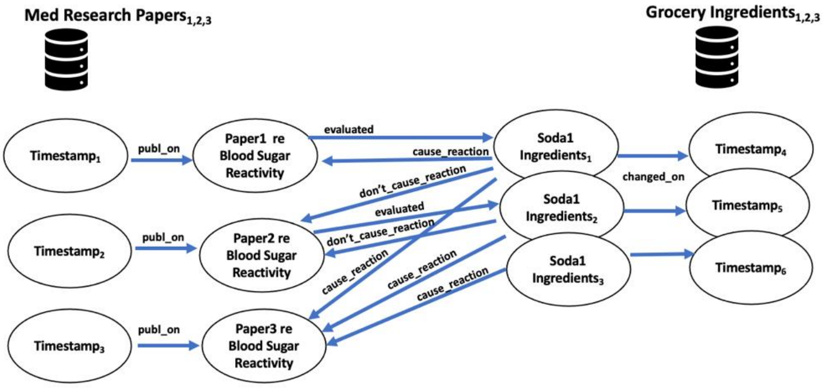

Fig. 4. Bi-directional context: Temporal updates to nodes in either decentralized source can change the edge between the sources.

This particular example is complicated by the fact that, for most purposes, the primary users of each dataset would prefer different outcomes from the updates. The medical researcher more likely would wish to see the medical conclusion mapped to each version of a product's ingredient list, requiring the retention of each as a separate node and a separate edge. While the consumer would likely prefer to see the medical conclusion mapped only to the form of the product currently stocked on shelves, requiring an overwrite that retains only one node and one edge.

3 Discussion

The provided use cases show instances in which searching for historical versions of a knowledge graph will require the ability to identify and retrieve data which does not share the same archival date and/or requires the retrieval of more than one version of some but not all nodes of the graph. These challenges are complicated by the data being owned by different parties, in different subjects, who may not even be aware of the use to which their data is put. Generally, these are not challenges that can be solved by simply using *date+n_days*, as the number of days and versions may be inconsistent. For knowledge graphs to evolve appropriately, there must be a notation to indicate, and a mechanism to produce, the desired impact of a contextual temporal relationship. As shown with the temporal context examples of simultaneity, lag, prediction, and bi-directionality, graph creators need to be able to express whether a temporal update to the data in a node should create a new node or overwrite the existing one, and whether it can result in a change to an edge or create a new one. To facilitate historical retrieval, there should be a standard for data owners to describe not only the data and time produced, but also versioning methodology – for example, metadata indicating whether data is overwritten or new date named versions produced; whether there is a marker for the final version of iterated data; and whether there is a graphed relationship that causes changes to this data.

References[2]

1. Taelman, R., et al, Continuously Updating Query Results over Real-Time Linked Data, In Proceedings of the 2nd Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW 2016) co-located with 13th European Semantic Web Conference (ESWC 2016) CEUR-WS, vol. 1585, pp. 1-10, Heraklion, Crete, Greece (2016) (http://ceur-ws.org/Vol1585/mepdaw2016_paper_01.pdf).
2. Anderson, J & Bendiken, A., Transaction-Time Queries in Dydra (Industry Paper), In Proceedings of the 2nd Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW 2016) co-located with 13th European Semantic Web Conference (ESWC 2016) CEUR-WS, vol. 1585, pp. 11-19, Heraklion, Crete, Greece (2016) (retaining past and current state as separately addressable stores) (http://ceur-ws.org/Vol-1585/mepdaw2016_paper_02.pdf).
3. Fernandez, J.D., Polleres, A., & Umbrich, J., Towards Efficient Archiving of Dynamic Linked Open Data, In Proceedings of the First DIACHRON Workshop on Managing the Evolution and Preservation of the Data Web co-located with 12th European Semantic Web Conference (ESWC 2015), CEUR-WS, vol. 1377, pp. 34-49, Portorož, Slovenia (2015) (http://ceur-ws.org/Vol-1377/paper6.pdf).
4. Quevas, I. & Hogan, A., Versioned Queries over RDF Archives: All You Need is SPARQL? In Proceedings of the 6th Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW) co-located with the 19th International Semantic Web Conference (ISWC 2020), CEUR-WS, vol. 2821, pp. 43-52, Virtual (instead of Athens, Greece) (2020) (exploring querying in and across massive versioned archives) (http://ceur-ws.org/Vol-2821/paper6.pdf).
5. Gleim, L. & Decker, S., Open Challenges for the Management and Preservation of Evolving Data on the Web, In Proceedings of the 6th Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW) co-located with the 19th International Semantic Web Conference (ISWC 2020), CEUR-WS, vol. 2821, pp. 11-16, Virtual (instead of Athens, Greece) (2020) (referring to the resolution of synchronization possibly through TimeMaps) (http://ceur-ws.org/Vol-2821/paper9.pdf).
6. See, e.g., https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm.
7. See, e.g., https://www.weather.gov/media/aly/Past_Events/2015/PNS_Microburst_ Jun_9_2015.pdf (example of open web data re: a microburst, showing time, longitude and latitude).
8. See, e.g., NY Open Data https://data.ny.gov/Transportation/Annual-Average-Daily-Traffic-AADT-Beginning-1977/6amx-2pbv (providing average daily vehicle usage per stretch of roadway).
9. https://catalog.data.gov/dataset/e-zpass-usage-statistics-beginning-2008 (providing EZ Pass usage by year by toll plaza).
10. See, e.g., https://data.ny.gov/Transportation/Motor-Vehicle-Crashes-Case-Information-Three-Year-/e8ky-4vqe (providing timestamp and DOT mileage marker for location of accidents).
11. See, e.g., https://www.opendoor.com/w/blog/how-school-ratings-impact-home-prices and https://www.niche.com/k12/search/best-school-districts/ (offering houses for sale tied to each school district ranking).
12. See, e.g., https://nces.ed.gov/programs/digest/mrt_tables.asp (annual release of education statistics).

[2] All web citations are as of September 5, 2021 and are not listed separately in each reference.

13. See, e.g., http://www.globalreportcard.org/about.html (downloadable global school district data).
14. See, e.g., https://infohub.nyced.org/reports/school-quality/information-and-data-overview (New York City open data on education, including testing scores).
15. See, e.g., https://www1.nyc.gov/site/finance/taxes/property-rolling-sales-data.page (NYC rolling sales data for residential real estate).
16. Pepino, Y.M., Tiemann, C.D., et al, Sucralose Affects Glycemic and Hormonal Responses to an Oral Glucose Load, Diabetes Care, Vol. 36(9), pp. 2530-2535 (American Diabetes Association, Sept. 2013) (https://care.diabetesjournals.org/content/36/9/2530).
17. "Sugar Reduction Innovation," Aug. 2021 (https://www.foodingredientsfirst.com/analysis-popup/sugar_reduction_aug_2021.html).
18. See, e.g., https://www.foodingredientsfirst.com/ (a website for the food industry with focus on rising and declining ingredient trends).
19. See, e.g., USDA Branded Food Products Database (https://data.nal.usda.gov/dataset/usda-branded-food-products-database/resource/cfceb689-7dab-498f-8762-707cd299646b) (providing ingredients for branded foods).