Remembering the Important Things: Semantic Importance in Stream Reasoning

Rui Yan[1], Mark T. Greaves[2], William P. Smith[2], and Deborah L. McGuinness[1]

[1] Tetherless World Constellation, Department of Computer Science,
Rensselaer Polytechnic Institute, Troy, NY, USA
yanr2@rpi.edu, dlm@cs.rpi.edu,
[2] Pacific Northwest National Laboratory, Richland, WA, USA
{Mark.Greaves,William.Smith}@pnnl.gov

Abstract. Reasoning and querying over data streams rely on the ability to deliver a sequence of stream snapshots to the processing algorithms. These snapshots are typically provided using windows as views into streams and associated window management strategies. In this work, we explore a general notion of *semantic importance* that can be used for window management of RDF streaming data using semantically-aware processing algorithms. Semantic importance exploits the information in RDF streams and surrounding ontologies for ranking window data in terms of its contribution to solution mappings. We also consider how a stream window management strategy based on semantic importance could improve overall processing performance, especially as available window sizes decrease.

Keywords: semantic importance, stream reasoning, RDF, cache, buffer, window

1 Introduction

Stream reasoning has been proposed with the goal of bridging both semantic reasoning and stream processing [9]. In stream reasoning, streaming data are referred as "RDF streams" [3][4] if RDF is leveraged for data annotation. RDF streams are usually considered to be boundless, thus there is no way to host the overall data view with "enough" computing and storage resources. As a common solution, a sliding window [1] with a limited size is typically employed to isolate relevant portions of the RDF streams, as well as to execute the queries.A window is generated by providing a preset size and step, in which the size refers to the maximum window data capacity and the step indicates how the window advances. The window is usually associated with a window management strategy that controls how the data in the window is consumed, processed and evicted. A simple timestamp-based window management strategy is first-in-first-out (FIFO), where items are replaced in strict order of arrival. Several implemented systems work this way. C-SPARQL [3] leverages either time-based or count-based sliding windows, where data enters and exits in a First-In-First-Out (FIFO) order. IMaRS [2] focuses reasoning tasks by both assigning each new

datum with a time-to-live period[3] and truth maintenance information[4], also in FIFO order. Basically, all timestamp-based strategies implicitly assume that a temporal ordering reliably reflects importance to the processing task, and thus timestamp-based window management will maximize the processing algorithms ability to deliver accurate interpretations of the stream.

Keeping the eviction order linked to the arrival order for data works well in systems like [2] and [3], where the expiration timestamp is either absent or assigned by the system. However, if data needs to exit the window out of order, FIFO-based strategies will not generally be adequate. This can happen if the streaming data itself carries out-of-order expiration timestamps, or if the expiration of the data is determined by other data in the stream, or if the use case supplies data in no particular order.

Our work is motivated by scenarios where stream reasoning needs more general window management strategies that preserve the most important data in the current window and preferentially evict the rest, so that the retained data can continue maximally leveraged.

The technical contributions include (1) the introduction of semantic importance along with an empirical definition; (2) a conceptual model of semantic importance that includes RDF stream management metrics; (3) an exposition of how our semantic importance conceptual model can be used to encode orderings and improve stream reasoning system performance.

2 Semantic Importance

Semantic importance enables order-aware window management that considers not only the temporal attributes of the data, but also the data contributions to solution mappings. We define semantic importance using a set of key metrics, each of which captures different aspects, including but not limited to *query contribution*, *provenance*, *trustworthiness* and *domain awareness*.

Specifically, for each datum [5] , its semantic importance is a priority vector [8] with elements preferentially ordered and with no restriction on the number of vector elements. This allows us to consider multiple semantic importance metrics simultaneously while preserving the ability to prioritize some metrics. For example, if an RDF stream is associated with an explicit expiration timestamp (τ_{exp}), the priority vector $[\tau_{exp}]$ will consider unexpired data to be important. If the query contribution (qc) is included, the priority vector $[\tau_{exp}, qc]$ can emphasize τ_{exp} over qc and thus encode that one unexpired datum is more important than another if it contributes more to the result, and expired data becomes less important however big its contributions are, in order to guarantee the constant output with the latest answers. By explicitly characterizing the importance for

[3] IMaRS attaches each data with an expiration timestamp that is calculated by adding the arrival timestamp to the window size.

[4] IMaRS incrementally materializes the inferences and manages both explicit and implicit statements based on the expiration timestamp of the asserted data.

[5] for RDF this is a graph that can contain one or more triples

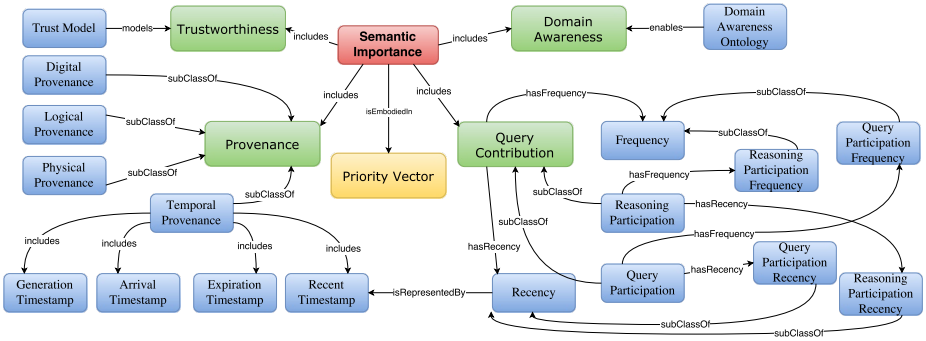

Fig. 1. Semantic importance conceptual model

each datum in the stream, we can enable order-awareness by ranking the semantic importance priority vectors as follows: priority vector $v_1 = [x_1, y_1]$ compares to $v_2 = [x_2, y_2]$ by first comparing the most preferred element x_1 and x_2. If $x_1 > x_2$, then $v_1 > v_2$; if $x_1 < x_2$, then $v_1 < v_2$; if $x_1 = x_2$, then continue to compare the next less preferred element y_1 and y_2, if $y_1 < y_2$, then $v_1 < v_2$; if $y_1 > y_2$, then $v_1 > v_2$; if $y_1 = y_2$, then $v_1 = v_2$. Priority vectors with more than two elements follow the same pattern.

3 Aspects of Semantic Importance

Figure 1 shows the current four aspects (in green color) of semantic importance. For query contribution, we further distinguish two components: frequency and recency. Frequency is an integer value that describes how many times a valid datum participates in the query, and recency describes the most recent timestamp for a valid datum participating in the query. Further, we distinguish two types of query contribution of a stream datum: query participation (direct graph pattern matching), and reasoning participation (intermediate graph pattern matching during reasoning) , where "participation" refers to the process in which the solution mapping[6] is generated by SPARQL query and streaming data. Both query and reasoning participation have corresponding frequency and recency.

Provenance in Figure 1 is also a factor in judging the importance of a streaming datum [5]. Temporal provenance includes (1) generation timestamp, assigned by the streaming source, which describes when the data is generated; (2) arrival timestamp, assigned by the processing system, which describes when the data arrives at the system; (3) expiration timestamp, assigned by either the processing system or the streaming source, which describes when the data expires; (4) recency timestamp, assigned by the system, which is associated with the query contribution recency. We include physical and logical provenance [6] as aspects of semantic importance as well. For example, sensors' physical geo-location and

[6] `https://www.w3.org/TR/rdf-sparql-query/#sparqlSolutions`

maintenance information fall into physical provenance; while sensor's measuring type falls into logical provenance. We also include provenance prominent in social network contexts that we call digital provenance. Digital provenance examples include digital geolocation information, user ID, and device ID. For example, consider a system that processes a tweet stream about UEFA EURO 2016, and executes the query *what is the sentiment distribution of UK residents on England's loss to Iceland?*. Digital provenance can be used to filter the data in the stream window because only those tweets with UK geolocation will be important to the query.

Trustworthiness is another aspect of semantic importance. Consider a smart city application where many traffic sensors are installed, and a shifting number of them are broken and issue erroneous readings. Assigning a specific trustworthiness score for each data element allows the system to manage the streaming window based on whether the individual data elements can be trusted. [7] describes modeling trust in streaming data, and provides a theoretical foundation to calculate a trust score online for each arrival data element. A trust range or threshold can be provided to capture the data with an adequate trust score, and be used to manage the window.

Query contribution, provenance, and trustworthiness are largely independent of the precise subject matter of the RDF stream, and so they provide components of a generic model of semantic importance. However, specific situated topics and reasoning tasks can contribute significant domain-dependent considerations. By examining the actual domain and specific streaming data use cases, we can join generic aspects with domain-specific ones, and model semantic importance in a more precise way. Our domain-aware aspect supports the modeling use case requirements, such as the goal of the streaming analysis, background knowledge and system constraints, etc. We are testing our approach in a soccer offside offence detection use case, comprising data on the real-time position of the ball and everyone on the field. Because only soccer players can commit offside offences, we can conclude that positional data for the referees will never contribute to an offside offence query[7] , and can be flushed from the window. In order to enable domain-awareness in stream reasoning systems, we can provide a domain-aware importance ontology that encodes the domain information necessary for the pre-registered continuous queries. This is different from the background ontology that encodes the essential knowledge of the domain, and is specifically focused on capturing domain knowledge that impacts querying and reasoning required to answer questions.

4 Remembering the Important Things

We discuss some interesting observations from our soccer use case experiments. First, window generation and moving is not constrained by semantic importance,

[7] Referee's position would be relevant for being in a position to "see" the offense but not actually to "affect" if the offense really occurs.

whose focus is on window management. Second, semantic importance can interact with data arrival order. In FIFA Laws of the Game, a player commits an offside offence by being in an offside position before interfering with the play. Thus :player :at :OffsidePosition data must arrive earlier than :player :interfere :play. The query, *who commits offside offence*, will not yield the result if these statements are not simultaneously in the window. Further, depending on the nature of the query, it is possible that neither alone will participate in the query, and so neither will be judged as semantically important without the other. And, if either datum is evicted from the window because it is not important, then the query will give an incorrect result. One possible solution is to decompose the query to alleviate query constraints.

We can decompose our offside query by asking *who is at an offside position* and *who interferes with play*. These two queries do not generate or combine results in stream, instead, they look for offside offence relevant data so that the window can update the semantic importance and rank the data to preserve necessary information. Once all necessities are collected, the target query is executed to provide final results.

Once the domain aware requirement ontology is in the system, domain literate filtering can be performed. Data can be ranked by its domain relevance, then irrelevant data can be filtered out because it is not important. Using the soccer offside example, any player information not used to calculate an offside position should not be considered as important. Domain literate filtering can significantly reduce the data space before the target query is executed, which enhances system response. It is typically suitable when the query relevant data is sparsely distributed in the data stream.

Semantic importance can be used to model several typical order-aware window management strategies: FIFO can be implemented as $[\tau_e, \tau_g]$, where τ_e is the expiration timestamp and τ_g is the generation timestamp. Least Frequently Used (LFU) as $[\tau_e, f_{rp}, \tau_g]$, where f_{rp} denotes the reasoning participation frequency and Least Recently Used (LRU) as $[\tau_e, \tau_{rp}, \tau_g]$, where τ_{rp} denotes the reasoning participation recency timestamp. If domain literate filtering needs to be performed with FIFO, we can implement new strategy FIFO/DL as $[dl, \tau_e, \tau_g]$ where dl denotes the domain literate filtering indicator.

Implementing semantic importance increases system overhead for computing the query contribution metrics because a reasoning explainer is required to trace back to the reasoning process to collect asserted statements. Domain literate filtering needs to perform a set of SPARQL queries that could potentially match a large fraction of the streaming data, which takes some time.

5 Conclusion and Future Work

We presented our preliminary notion of semantic importance together with its key aspects, and described how to use semantic importance to enable order-awareness in stream reasoning. Semantic importance enables a flexible, powerful, efficient and customizable framework to support window management that

can maintain and potentially improve reasoning results in compute- and space-limited settings. There are more details of our specific use of semantic importance in the soccer offside offence use case [8], where we have been able to show that semantic importance can be used to identifiy important data, increase accuracy in reasoning-based query answers and be the basis of streaming window management strategies. In the future, we will further develop the conceptual model by both adding more aspects and enriching additional details of each aspect.

6 Acknowledgments

The research described in the paper is part of the Analysis in Motion research initiative at the Pacific Northwest National Laboratory, a multi-program national laboratory operated by Battelle for the U.S. Department of Energy under contract DE-AC06-76RLO-1830. The work was conducted under Laboratory Directed Research and Development (LDRD) funding from PNNL to RPI.

References

1. Arvind Arasu, Brian Babcock, Shivnath Babu, Mayur Datar, Keith Ito, Itaru Nishizawa, Justin Rosenstein, and Jennifer Widom. Stream: the stanford stream data manager (demonstration description). In *Proc. of the 2003 ACM SIGMOD international conference on Management of data*, pages 665–665. ACM, 2003.
2. Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. Incremental reasoning on streams and rich background knowledge. In *Extended Semantic Web Conference*, pages 1–15. Springer, 2010.
3. Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. C-sparql: a continuous query language for rdf data streams. *International Journal of Semantic Computing*, 4(01):3–25, 2010.
4. Jean-Paul Calbimonte. Rdf stream processing: let's react. In *Proceedings of the 3rd International Conference on Ordering and Reasoning-Volume 1303*, pages 1–10. CEUR-WS. org, 2014.
5. Ming Gao, Che-Qing Jin, Xiao-Ling Wang, Xiu-Xia Tian, and Ao-Ying Zhou. A survey on management of data provenance. *Chinese Journal of Computers*, 33(3):373–389, 2010.
6. Hyo-Sang Lim, Yang-Sae Moon, and Elisa Bertino. Research issues in data provenance for streaming environments. In *Proc. of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS*, pages 58–62. ACM, 2009.
7. Hyo-Sang Lim, Yang-Sae Moon, and Elisa Bertino. Assessing the trustworthiness of streaming data. Technical report, Technical Report TR 2010-09, CERIAS, 2010.
8. Thomas L Saaty. Decision-making with the ahp: Why is the principal eigenvector necessary. *European journal of operational research*, 145(1):85–91, 2003.
9. Emanuele Della Valle, Stefano Ceri, Frank van Harmelen, and Dieter Fensel. It's a streaming world! reasoning upon rapidly changing information. *IEEE Intelligent Systems*, 24(6):83–89, 2009.

[8] `https://tw.rpi.edu/web/Courses/Ontologies/2016/projects/soccer`