GenACT: An Ontology-based Temporal Web Data Generator

Gunjan Singh[1], Udit Arora[1], Shashikant Kumar[1], Riccardo Tommasini[2,5], Pieter Bonte[3], Sumit Bhatia[4], and Raghava Mutharaju[1]

[1] Knowledgeable Computing and Reasoning Lab, IIIT-Delhi, India
{gunjans, udit18417, shashikant18096, raghava.mutharaju}@iiitd.ac.in
[2] LIRIS Lab, INSA Lyon, France, riccardo.tommasini@insa-lyon.fr
[3] University of Tartu, Estonia, riccardo.tommasini@ut.ee
[4] KU Leuven Campus Kulak, Belgium, pieter.bonte@kuleuven.be
[5] MDSR, Adobe Inc., Delhi, India, sumit.bhatia@adobe.com

Abstract. Time is a fundamental concept in data processing. The growth of social media (SM) data and the rise of the Internet of Things (IoT) underscores the necessity for studying temporal data on the Web. However, accessing realistic temporal data poses significant challenges across data collection, knowledge representation, and real-time provisioning, with no comprehensive solution available yet. To tackle these challenges, we introduce GenACT, a novel data generator rooted in the dynamics of Academic Conference Tweets (ACT), which serves as an ideal domain for eliciting application scenarios spanning temporality, dynamicity, and timeliness. The foundation of GenACT is a domain-specific ontology crafted to conceptualize tweets around an Academic Conference Event (ACE) realistically. The ACE ontology is available in all four OWL 2 profiles. Additionally, RDF instantiation allows for real-time simulation of ongoing academic discussions on Twitter. GenACT stands out for its ability to configure different data segments using SPARQL-based partitioning strategies. This versatility makes it adaptable to various analytical tasks, enabling researchers to focus on specific aspects of the data for their studies. GenACT is designed to seamlessly provide temporal and static data in a streaming format, tailored specifically for applications in studying knowledge graph evolution, temporal reasoning, and stream reasoning. The ontology, code, and documentation are available under the Apache 2.0 License at https://github.com/kracr/temporal-data-generator.

Keywords: Data Generation· Stream Reasoning · Knowledge Representation

1 Introduction

The domain of knowledge representation and reasoning has evolved significantly with the rise of knowledge-intensive temporal data, largely influenced by social media and the Internet of Things, which resemble dynamic knowledge graphs

(KGs) due to their structure and the ever-evolving nature of data. Although KGs are becoming more prevalent, there remains a lack of exploration in analyzing and managing their evolution. In their recent survey, Polleres et al. [26] encouraged a closer look at KG evolution. However, to study KG evolution effectively, we need more control over the construction and maintenance of the KGs. Even when KG changelogs and versions are available, e.g., in the case of Wikidata [6] or DBpedia [21], the existing real-world datasets fall short in three crucial aspects required to capture real-world activities in data [26]: (a) *temporality*, i.e., the ability of the KG to capture the order and duration of the events, is often incomplete or enforced in RDF via reification; (b) *dynamicity*, i.e., the ability of the KG to capture progressive variation, is usually implemented using versioning and is not well documented; (c) *timeliness*, i.e., the ability of the KG to stay up to date, is neglected, especially when the velocity of change meets streaming scenarios [7].

While the previously discussed aspects are crucial, KGs must also exhibit additional characteristics to capture the dynamic and complex nature of real-world activities accurately: (a) *diverse graph patterns*: real-world data rarely follows a single, rigid structure, and KGs that support a variety of graph patterns can better represent the intricate relationships between entities. (b) *temporal granularity*: real-world events often consist of subevents occurring over different timeframes, and KGs should accommodate these varying timescales. This enables the modeling and analysis of relationships with different complexities. (c) *scalability and data fluctuations*: information density can fluctuate with sudden data bursts during significant events, and KGs that incorporate these fluctuations accurately represent diverse data volumes. (d) *segmented by attributes*: different entities possess unique characteristics, and KGs segmented by attributes can capture these variations, allowing for more nuanced analysis and exploration of specific entity types. KGs endowed with these characteristics offer a significant advantage: *the ability to tailor data to simulate several application scenarios.*

Unfortunately, real-world KGs with these comprehensive capabilities are currently rare. Incorporating this flexibility is crucial for KGs to become more suitable for applications like temporal web data generation and analysis, where capturing the dynamic intricacies of real-world activities is paramount. *Synthetic data generation* [3] offers the flexibility to explore all these features, possibly enhancing evaluation accuracy and enabling realistic simulation of dynamic KG evolution. However, existing works in the area are limited in scope (cf. Section 6).

To address these gaps, this work introduces GenACT, a synthetic data generator for the Web. GenACT data are modeled after realistic data from a knowledge-intensive yet interpretable domain, i.e., Academic Conference Tweets (ACT). From a comprehensive study, we found that ACT is the ideal domain to elicit application scenarios that span temporality, dynamicity, and timeliness. We built an OWL 2 [13] ontology based on tweets related to academic conferences. This ontology is available in four flavors corresponding to each of the four OWL 2 profiles (EL, QL, RL, DL) [13]. This extends the utility of GenACT for

[6] `https://www.wikidata.org/wiki/Wikidata:Main_Page`

applications modeled using any of these profiles. Furthermore, GenACT generates data that can be partitioned into segments based on specific graph patterns and attributes (e.g., conferences, users). These partitions exhibit varying data peaks and intricate relationships, allowing researchers and developers to conduct insightful analyses.

Outline. Section 2, first elaborates on the rationale behind selecting this domain, domain analysis, and different application scenarios that illustrate the practical applications of GenACT. Section 3 discusses the knowledge representation efforts, i.e., an overview of the construction of the ontologies involved. The data generation process is discussed in Section 4, while Section 5 engages in a discussion regarding the generated data, evaluating its effectiveness on two reasoners, CSparql2 [7] and RDFox [22], thereby demonstrating the potential of the generated data. Additionally, we address limitations and outline potential avenues for future development. Section 6 provides a comprehensive review of related work, followed by conclusions in Section 7.

2 Academic Conference Tweets (ACT)

Indeed, drawing upon the principles outlined in [20], a critical step in the development of ontologies is defining a well-scoped domain. The domain needs rich temporality (timestamps, durations, versions) to model evolving knowledge. It should also be dynamic and timely, with continuous information flow and real-time interactions. Ideally, the domain should resonate with the target research community, be representable using ontological modeling frameworks like OWL 2, and allow simulated data generation. Social media platforms like Twitter or Reddit provide an excellent source for such temporally rich data due to their continuous streams of user-generated content, which capture evolving trends, events, and discussions in real-time. However, collecting real data from these

[7] https://github.com/streamreasoning/csparql2

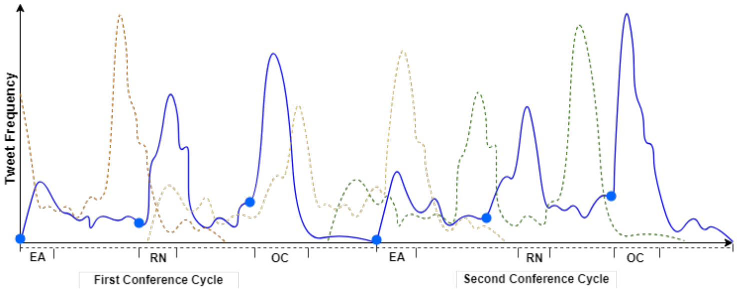


Fig. 1: Conference Lifecycle: dots represent Absolute Events. **Legend:** [E]arly [A] announcements, [R]review [N]otifications, or the currently [O]ongoing [C]conference. Colours and strokes differentiate the conferences, running simultaneously yet in different phases, animating the temporal landscape.

platforms has become increasingly challenging due to their stringent terms of service, limited API access, and the high costs associated with data extraction. Beyond the technical hurdles, there are also significant legal and ethical barriers, such as data privacy regulations which impose strict restrictions on how data can be collected and used. These factors make it difficult to obtain large-scale, real-world datasets from these sources.

Twitter stood out as an ideal data source for our synthetic data generation approach due to its event-driven nature and medium to high rate of data generation [29,24]. Within Twitter, we specifically target tweets surrounding academic conferences. This focus offers two advantages: first, ACT inherently captures the detailed sequence of events (rich temporality) surrounding the conference, which is crucial for knowledge modeling. Second, event-related tweets exhibit extended engagement (Figure 1), aligning with our modeling goals.

	Scenario	Temporality	Dynamicity	Timeliness
1	Trending Topics Across Conferences *Keeps users posted on topics trending across conferences*	✓	✗	✓
2	Active Research Groups *Monitor key players and their collaborators in given fields*	✓	✓	✗
3	Publication Activities of Organizations *Analyse researchers' interdisciplinary work*	✓	✓	✗
4	Conference Match by Research Interests *Identifies conferences that match users' research interests*	✗	✓	✗
5	Interdisciplinary Authors *Identify potential interdisciplinary collaborations*	✓	✓	✗
6	Session Popularity *Identify the most popular topics and speakers*	✓	✓	✗
7	Global Research Focus *Identifies countries most active in a given research field*	✓	✓	✓
8	Funding Organizations *Identifies organizations financing researchers*	✗	✓	✗
9	Networking Opportunities *Identifies researchers interested in the same given areas*	✓	✓	✗
10	Collaboration Networks *Monitors academic collaboration networks to understand research dynamics*	✓	✓	✗
11	Non-academic Collaborators *Monitors industry-academia relations to highlight knowledge transfer*	✗	✓	✗
12	Geographical Distribution *Identifies active countries in given research fields*	✓	✓	✗
13	Platform Impact *Analyse conference trends on various platforms*	✓	✗	✓

Table 1: Scenarios for Competency Questions.

The ACT domain accurately captures the “when” (temporality), “what is new” (dynamicity), and “staying informed” (timeliness) aspects crucial for representing real-world activities. Temporality guarantees the precise sequencing and duration of events, enabling queries like “What happened before the call for papers?” It accurately represents the chronological order of events, such as announcements, submissions, and notifications. Furthermore, ACT incorporates simple ordering, e.g. order (conference announcements precede call for papers), and all forms of temporality [26]: timestamped event data (Tweets), versioned data (conference series and user bios updates), and intervals (conference duration). Dynamicity ensures real-time adaptation to new tweets, updates, and schedule changes, providing users with the most current information. ACT encompasses continuous information flow (real-time schedule updates), interactions (social media engagement), and progressive data changes (user profiles and conference details). Timeliness includes prompt notifications and reminders, enabling effective user participation in the conference process. ACT involves prompt notifications (session starts and submission deadlines), timely data updates (new information reflection), and responsive adjustments (adaptive scheduling). Moreover, ACT offers a rich context to explore and model different language constructs from the OWL 2 profile (discussed in Section 3), facilitating diverse reasoning challenges, which are reflected in the scenarios provided in the next section that can be used as CQs.

Furthermore, the domain’s accessibility and familiarity within the academic community make it highly suitable for research and analytical purposes. Moreover, since Twitter data is not openly accessible, enabling data generation is valuable for stimulating research, underscoring its relevance and impact in knowledge representation and reasoning domains.

Competency Questions (CQs). We explored the domain’s suitability in simulating real-world temporal applications by formulating scenarios that can be used as CQs [30], listed in Table 1. Due to space constraints, these scenarios are presented concisely, yet each entry effectively captures the essence of its corresponding query. The queries corresponding to the competency questions can be found in our GitHub repository[8]. The table also further emphasizes opportunities for modeling temporality, dynamicity, and timeliness, thus addressing the dynamic nature of academic research.

The following conceptualization is required to understand the ontology modeling efforts presented later. In particular, we introduce the notion of Temporal and Continuous Queries for the unfamiliar reader that are needed to address notions such as temporality, dynamicity, and timeliness that push the envelope of the standard Semantic Web stack [13]. We use the CQ 3: “Monitoring Publication Activities of Organizations” as an example.

Definition 1. *Temporal queries offer native support for retrieving and manipulating time-referenced data, such as timestamps, durations, and time intervals..*

[8] `https://github.com/kracr/temporal-data-generator`

Listing 1.1 shows an example query implemented using a temporal extension of SPARQL (T-SPARQL). The query execution leverages the `INTERSECT` clause to exploit data temporality in RDF, in this case, the year 2023.

Definition 2. *A continuous query is a query whose evaluation is repeated at regular time intervals and whose results are the union of results [6].*

Listing 1.2 shows an example query implemented using a streaming extension of SPARQL (RSP-QL). The query execution leverages the `Named Window` class to chunk the input stream into finite portions and continuously updates the results every 10 minutes over a 1-hour window. RSP-QL allows static and streaming data to join for enrichment and reasoning.

3 Ontology Modeling

This section presents our efforts in capturing the knowledge for the ACT domain. The conceptualization is based on a comprehensive study of academic conferences conducted on Twitter. The study aimed to gain insights into the temporal patterns and content categories associated with tweets about conference events.
Domain Understanding. The tweets on an academic conference span several months, commencing with early announcements encompassing conference announcements and the calls for papers. The process then navigates through distinct phases such as submission reminders, review notifications, and post-paper acceptance tweets, culminating in the conference period marked by active sharing of presentations and experiences. Post-conference engagement also occurs as the community reflects on the event's impact, while random tweets inject diversity, including keynote announcements, discussions on trends, and user insights. The tweets around an Academic Conference Event (ACE) span over several months with different data peaks at different time intervals, similar to real-world scenarios. Moreover, the cyclic nature of conferences, occurring regularly every year, further helps simulate real-world scenarios.

We categorized tweets into five main types: Announcements, Reminders, Notifications, Insights, and Others. Table 2 indicates different categorizations and the types of tweets that fall into each category. Announcements, reminders, notifications, and insights follow a structured sequence, while the Others category encompasses tweets that may occur randomly. Such a categorization clarifies

```
SELECT ?organization ?domain (COUNT(?paper) AS ?publications)
INTERSECT(?t,"[2023-01-01,2023-12-31]")
WHERE {
  ?paper :hasAuthor ?author; ?author :hasAffiliation ?organization. | t.
  ?paper :hasDomain ?domain.
}
GROUP BY ?organization ?domain
```

Listing 1.1: Counting papers published in 2023 with T-SPARQL.

the data generation algorithm's flow. Moreover, it sets a basis for extensibility, allowing easy integration of new categories or templates in the future.

Note that this list is not a comprehensive list of tweet types, and there may be additional categories, such as tweets expressing disappointment after rejections or complaints, among others. However, we have included a significant number of tweet templates that replicate the real-world scenarios described earlier. Our data generation algorithm is adaptable and can incorporate new templates or categories (see Section 4) in the future.

We can correlate the tweeting pattern with the concept of 'Absolute Events', as depicted in Figure 1. Three distinct data peaks, each signifying an absolute event, mark critical junctures like 1) the Early Announcement (EA) encompassing the main Conference Announcement (CA) and the first Call for Papers (CfP), 2) Review Notifications (RN), and 3) the ongoing conference (OC). Notably, peaks rise significantly during notification periods, show a small increase just before the conference, reach their highest point during the event, and gradually decline in the post-conference phase.

Domain Conceptualisation. Our ontologies address temporality by representing order, eventful timestamps, and duration intervals [26].

<u>The Ordered Time Model</u> captures the relation between absolute events. Indeed, we can ask CQs such as "*Was the call for paper announced before the conference?*" to have a negative (or positive) answer and can be leveraged to test the temporal consistency of the data.

<u>The Absolute Time Model</u> adds to the previous model the notion of timestamps and allows a finer grain comparison. In particular, it is possible to ask questions about trends (cf Section 1), quantify time windows, or compute historical queries like "*How many times, on average, authors write during a conference*?"

<u>The Interval Time Model</u>. We adopted this model for the conference (and its sessions). They are described by start and end dates and duration (e.g., five days). Such modeling enables asking questions such as "*What happened during ESWC 2024?*", and "*What is the session that overlaps with session 3, and is having the most participation*?".

```
REGISTER STREAM <http://example.org/stream/publications>
FROM NAMED WINDOW :W1 ON <stream1> [RANGE PT1H STEP PT10M]
SELECT ?organization ?domain (COUNT(?paper) AS ?publications)
WHERE {
    WINDOW :W1 {
  ?paper :hasAuthor ?author; ?author :hasAffiliation ?organization. | t.
  ?paper :hasDomain ?domain.
    }
}
GROUP BY ?organization ?domain
```

Listing 1.2: RSP-QL Query to continuously monitor publication activities of organizations over a window of 1 hour.

ACE Ontology. We developed four moderate-sized ontologies describing an ACE, one for each OWL 2 profile (EL, QL, RL, and DL). The ontologies can be accessed from the GitHub repository[8]. Each ontology consists of concepts and properties describing various aspects of academic conferences. The hierarchy among some classes, including the relations between them, is shown in Figure 2.

Every ontology includes the classes `Conference`, `Organizer`, `Talks`, `Manuscript`, `Author`, `Role`, `EventMode` and `Attendee`, object properties such as `hasAuthor`, `attends`, and `hasCollaborator` and data properties such as `hasTitle`, `hasLocation`, and `givesTalkOn`. We ensure that the axioms cover all the constructs corresponding to a particular OWL 2 profile. In Table 3, we include a few examples of the axioms in description logics [14] from ACE ontology.

To expedite development, the ACE ontology reuses concepts such as `foaf:Person` and `foaf:Organization`. We also incorporate concepts such as `Student`, and `Organization`, from OWL2Bench [31], a benchmark based on a university domain designed for evaluating OWL 2 reasoners.

To demonstrate the syntactic and semantic limitations imposed on different constructs in each OWL 2 profile, we use the example, *Every manuscript has at least one author*. The class `Manuscript` is added to all the profiles, and the definition varies slightly (provided below) depending on the OWL 2 profile.

- In OWL 2 EL, $\texttt{Manuscript} \equiv \exists\texttt{hasAuthor.Author}$, indicating that every manuscript must have at least one author.
- In OWL 2 QL, where existential restrictions are not allowed in subclass expressions, $\texttt{Manuscript} \sqsubseteq \exists\texttt{hasAuthor.Author}$. This definition implies that when there is at least one author, it is classified as a manuscript.
- In OWL 2 RL, due to the prohibition of existential restrictions in superclass expressions, the definition is $\exists\texttt{hasAuthor.Author} \sqsubseteq \texttt{Manuscript}$, which signifies that when an author exists, it is associated with a manuscript.
- In OWL 2 DL, the definition uses qualified minimum cardinalities, expressed as $\texttt{Manuscript} \equiv\ \geq 1\ \texttt{hasAuthor.Author}$, ensuring that every manuscript must have at least one author and making the axiom more expressive.

Tweet Ontology. Our ontology development prioritizes modularity for future scalability and integration. Hence, the ontology that captures the tweet meta-

Tweet Category	Before Conference	During Conference	After Conference
Announcements	Main Conference ▲ Call for Papers ▲ Keynotes, Panelists and Sponsors ▲	-	Best Paper Awards ▲ Next Conference ▲
Reminders	Submission ● Registration ●	Upcoming Sessions ●	-
Notifications	Accepted Papers ■ Schedule Changes ■	Ongoing Sessions ■	-
Insights	Based on Accepted Papers or Schedule Changes ☆	Based on Presentations ☆	Recap and Reflections ☆
Others	Excitement for the Conference ◆ Gratitude for the Grants and Volunteer Opportunities ◆	Job Postings ◆	Acknowledgement and Gratitude ◆

Table 2: Overview of Tweet Categories. The shape indicates the category.

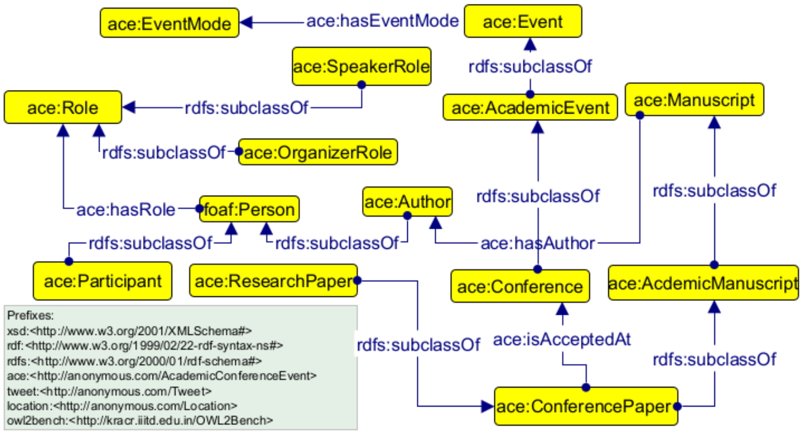


Fig. 2: Partial class/relationship hierarchy in ACE Ontology.

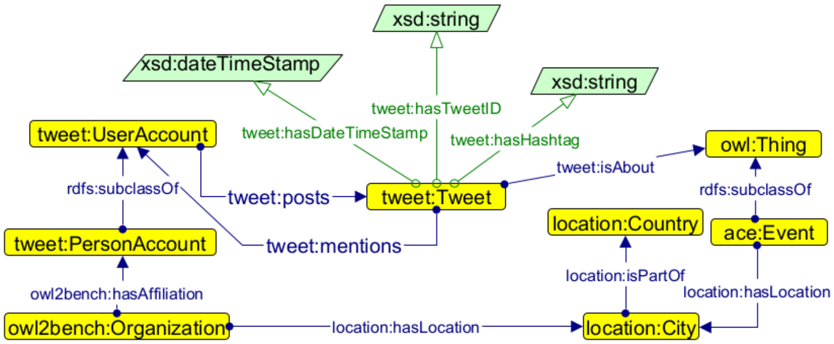


Fig. 3: Other Schemas

data is kept separate from the ACE ontology. This allows for future expansions, potentially incorporating other social media platforms. The core classes in the Tweet ontology include `UserAccount` and `Tweet`, which are connected via the object property `posts`. Each `Tweet` includes attributes such as `dateTimeStamp`, `hasHashtags`, and `mentions` of other Twitter users. To capture the relatively stable but occasionally changing nature of the user's bio information, we added the object property `hasAffiliation`. Additionally, entities from the Tweet and ACE ontology are linked with OWL2Bench and Location ontologies, such as `Organizations` and `City`, which are considered static background knowledge for the generated data. Figure 3 illustrates this integration. Note that the location of `Organizations` is defined as `City` for simplicity, but this will be refined in the future for more detailed location information.

4 Data Generation

The data generation is based on several key observations. Firstly, given the inherent diversity in natural language expressions within tweets, multiple variations of a particular tweet type can exist. For example, when a paper is accepted, the co-authors might post about it. Despite language differences, the meaning of the tweet from each of the co-authors would, most likely, remain consistent. Secondly, despite the brevity of the tweets, each tweet encapsulates significant implicit information related to the conference, covering multiple entities from the ontologies. Figure 4 illustrates how multiple tweets of the same type are mapped to the same RDF Graph, involving several entities, regardless of how short the tweet is. This mapping approach was leveraged to generate the data.

As shown in Figure 5, the data generation pipeline consists of two steps: Event Data Generation and Sequence Data Generation. The first step generates data for the specified number of conference instances. The second step allows users to create segments of the generated data to simulate different scenarios.

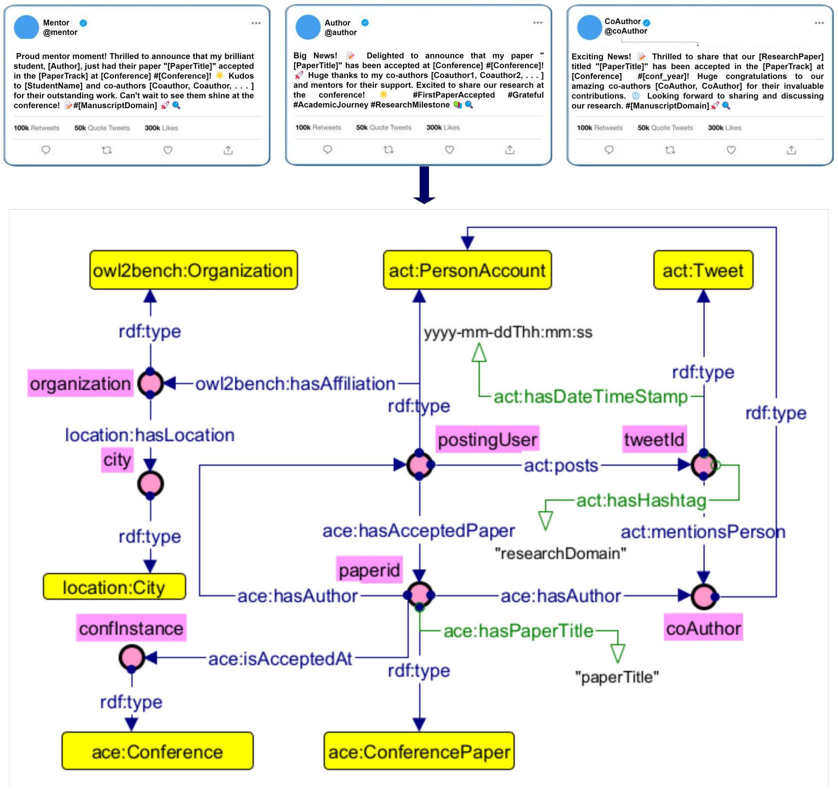
4.1 Event Data Generation

In the first step, we defined several realistic templates for different categories of tweets (Table 2). These templates act as structured formats for building RDF graphs. After generating these tweet templates, they are mapped with RDF triples based on the underlying knowledge. The Twitter templates and their mappings are on our GitHub repository[8].

After establishing the RDF graph structure for each template, we fill in specific details through instantiation. The data generation process begins with user-specified conference counts and cycles. For example, if a user requests data for two conferences and ten cycles, GenACT generates two distinct data directories, each representing a conference like *ER2024* and *ISWC2024*. Data is organized within each directory to represent ten cycles, for example, spanning from *ER2014* to *ER2024*. This structured approach ensures clarity and organization in the generated dataset.

OWL Construct Involved	Axiom
ObjectSomeValuesFrom	$\texttt{KeynoteSpeaker} \equiv \exists\texttt{hasRole.KeynoteSpeakerRole}$
ObjectAllValuesFrom	$\texttt{OrganizingCommittee} \equiv \forall\texttt{hasOrganizingCommitteeMember.Organizer}$
ObjectExactCardinality	$\texttt{SingleAuthorPaper} \equiv =1\texttt{hasAuthor.Author}$
ObjectMaxCardinality	$\texttt{ConferencePaper} \equiv \leq 1\ \texttt{isAcceptedAt.Conference}$
ObjectMinCardinality	$\texttt{Manuscript} \equiv \geq 1\ \texttt{hasAuthor.Author}$
ObjectIntersectionOf	$\texttt{StudentAuthor} \equiv \texttt{Student} \sqcap \texttt{Author}$
ObjectComplementOf	$\texttt{NonAuthor} \equiv \neg\texttt{Author}$
SubClassOf	$\texttt{Volunteer} \sqsubseteq \texttt{Participant}$
EquivalentClasses	$\texttt{Attendee} \equiv \texttt{Participant}$
ObjectPropertyChain	$\texttt{hasAcceptedPaper} \circ \texttt{hasAuthor} \sqsubseteq \texttt{hasCoAuthor}$

Table 3: Examples of TBox axioms from GenACT. OWL Constructs are in Manchester syntax, and the axioms are written in Description Logics Syntax.



Mentor
@mentor

Proud mentor moment! Thrilled to announce that my brilliant student, [Author], just had their paper "[PaperTitle]" accepted in the [PaperTrack] at [Conference] #[Conference]! 🌟 Kudos to [StudentName] and co-authors [Coauthor, Coauthor, . . .] for their outstanding work. Can't wait to see them shine at the conference! 📝#[ManuscriptDomain] 🚀 🔍

100k Retweets 50k Quote Tweets 300k Likes

Author
@author

Big News! 📝 Delighted to announce that my paper "[PaperTitle]" has been accepted at [Conference] #[Conference]! 🚀 Huge thanks to my co-authors [Coauthor1, Coauthor2, . . .] and mentors for their support. Excited to share our research at the conference! 🌟 #FirstPaperAccepted #Grateful #AcademicJourney #ResearchMilestone 📚 🔍

100k Retweets 50k Quote Tweets 300k Likes

CoAuthor
@coAuthor

Exciting News! 📝 Thrilled to share that our [ResearchPaper] titled "[PaperTitle]" has been accepted in the [PaperTrack] at [Conference] #[conf_year]! Huge congratulations to our amazing co-authors [CoAuthor, CoAuthor] for their invaluable contributions. 🌐 Looking forward to sharing and discussing our research. #[ManuscriptDomain]🚀 🔍

100k Retweets 50k Quote Tweets 300k Likes

Fig. 4: Mapping of tweets generated after paper acceptance to RDF Graph

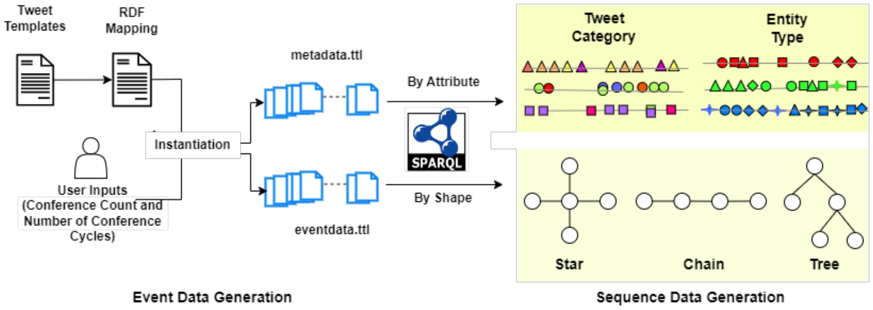


Fig. 5: Data Generation Pipeline. Identical shapes in a sequence denote partitioning by tweet categories (Table 2), while identical colors denote partitioning by the same type of entity (e.g., Conference, User).

For realistic data generation, the temporal modeling of tweets is essential. Tweets, generated in chronological order, naturally exhibit temporal relations.

For example, a tweet announcing a keynote speech or a paper to be presented typically precedes tweets discussing its content. When instantiating RDF graphs, we ensure the temporal sequence in our data generation. While we maintain a sequence in tweet generation, we introduce an element of randomness by generating tweets in a non-sequential manner. This adds richness to the dataset by simulating the spontaneity and diversity in real-world social media interactions. To allow flexibility in generating varying peaks, we use random number generators to vary the start times of conferences, durations between events, and tweets within these conferences. It is noteworthy that, akin to users modifying their bios on Twitter, we also ensure that certain users can update their bios.

We have adopted an approach to enhance efficiency to meet the requirement of tunable streaming rates for temporal web data. As illustrated in Figure 5, instead of having all tweets represented as RDF triples in a single file, we have organized them into separate files for each tweet instance. Specifically, we store RDF triples corresponding to each tweet in two files, one for the metadata corresponding to the tweet *(metadata.ttl)* and another for the conference-related information *(eventdata.ttl)*. Storing each tweet's data in separate files facilitates quick access to timestamps for specific observations and helps avoid the performance issues related to querying a single large file. This method is especially advantageous for generating diverse sequential data that can be streamed at the desired rate (see Section 4.2).

4.2 Sequence Data Generation

The generated data can be divided into multiple segments through strategic partitioning using SPARQL[9] queries. This customization enhances the dataset's utility for diverse research and analysis purposes.

Partitioning by Attribute. The generated event data offers flexibility by allowing partitioning based on various attributes. Each tweet metadata file created in the previous step contains relevant information that can be used to generate attribute-specific datasets. These attributes include different entities involved in the ACT domain, such as conferences, persons, organizations, research domains, and conference phases. For instance, querying `?tweetid :isAbout ?conferenceInstance. ?conferenceInstance rdf:type :Conference` on the metadata files, the triples in the corresponding event data file are used to create dedicated `Conference` segments for each `?conferenceInstance`. Similarly, querying the metadata files for `?tweetid :mentions ?userAccount` will create different `User` segments. Figure 6 shows these two partitions: `Conference` and `User`. The former creates segments for each conference. This facilitates analysis of a conference's temporal evolution, from the initial announcement to its conclusion. While these segments are fewer, they encompass a variety of users, organizations, and related data. Figure 6 shows the mix of tweets within each conference stream. Partitioning by users results in segments representing the

[9] `https://www.w3.org/TR/sparql11-overview/`

activities of individual users. Although individual user tweet frequency might be low, user-based segments offer a multitude of interconnections, leading to potentially complex reasoning tasks. Here, the number of partitions equals the number of users involved in the conference tweets. It is important to note that a single user might be involved in multiple conferences and have different roles within those conferences. Similarly, a single tweet might involve multiple users. Figure 6 also depicts this user-based partitioning with its inherent challenges due to the high number of streams and interconnected entities.

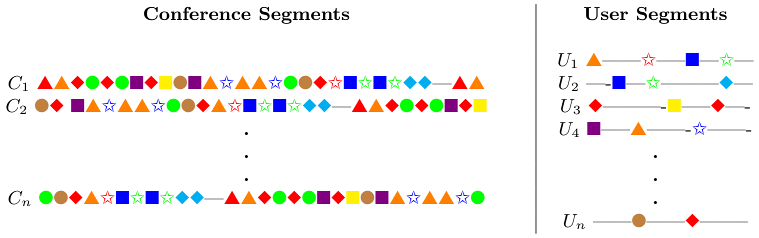


Fig. 6: Partitioning by Attributes. Different shapes and colors represent different tweet categories as outlined in Table 2.

Partitioning by Shape. Our data partitioning strategies extend beyond specific attributes such as conferences and users. The data can also be partitioned based on its inherent structure, revealing valuable relationships within the event data. Here, we define and enforce specific structural patterns within data segments, ensuring that all events within a segment adhere to a consistent form. This fosters a structured approach to data modeling and analysis [8].

Figure 7 illustrates three common shapes (out of a variety of possible graph patterns) in our generated data: tree, star, and chain. Tree structures represent hierarchical relationships, useful for analyzing nested dependencies. Chain structures model linear events or interactions, which is ideal for chronological analysis. Star structures centralize around a single node with multiple connected entities. This is useful for hub-and-spoke models like social media interactions where a central user connects with many others.

5 Discussion

Patterns and Peaks Analysis. Figure 8 illustrates an example of the tweet volume patterns and peaks generated per week for five conferences. It demonstrates that even if each conference displays three distinct peaks individually, merging the data into a single stream leads to diverse peaks and volumes. Thus providing insights into the dynamic nature of the events and their usefulness in mimicking different real-world activities for various analytical purposes.

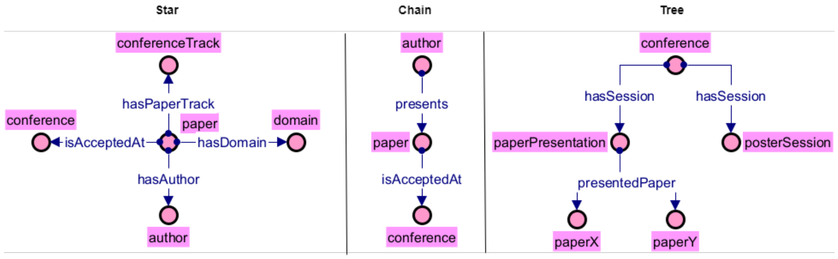


Fig. 7: Partitioning by Shapes

Functional Utility. To demonstrate GenACT's utility in evaluating reasoning systems, we executed two queries corresponding to CQ 3: *Monitoring Publication Activities* and CQ 7: *Global Research Focus* (cf. Table 1), on two reasoners: CSparql2[7] and RDFox[22]. CSparql2 is a stream reasoner based on RSP4J [33], while RDFox is a highly scalable, parallelized, in-memory RDF store supporting incremental reasoning. We employed pre-existing code available at [10]. Our evaluations were conducted on a system equipped with an Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz and 8 GB RAM. CQ 7 is a variant of CQ 3 (refer to Listings 1.1 and 1.2) with an additional basic graph pattern in the query: `?organization :hasLocation ?location`. This query utilizes static background data about locations to respond. Due to reasoners' limitations in performing OWL-based reasoning, we excluded triples involving OWL constructs from the queries, such as `?paper rdf:type :ConferencePaper`, which involves existential restrictions and cardinalities depending on the selected ACE ontology profile. The evaluation outcomes, as illustrated in Figure 9, reveal that the generated data is effective in identifying variations in performance. It is important to note that our focus here is not on performance evaluation but on demonstrating the

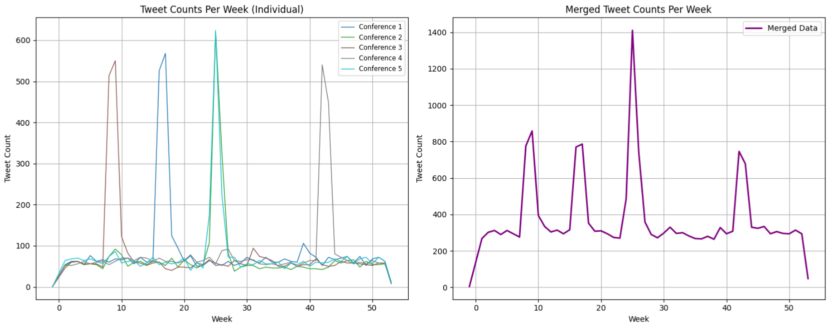
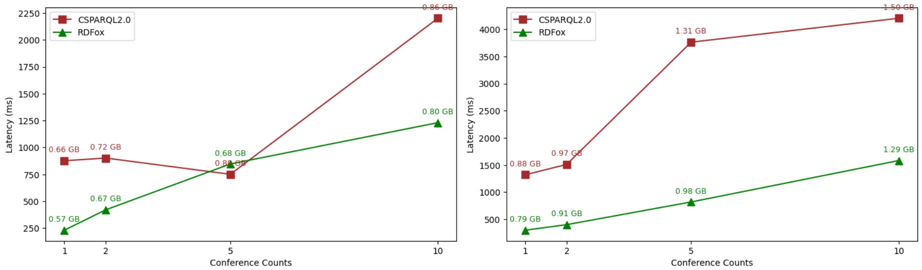


Fig. 8: Tweet volume patterns and peaks for five conferences: individual conference streams (left) and merged stream (right).

[10] `https://github.com/SRrepo`



(a) Varying number of parallel streams without background data

(b) Single stream with static (increasing) background data

Fig. 9: Key performance indicators: Latency (ms) and Memory Consumed (GB), for the queries corresponding to CQs 3 and 7, for different numbers of conferences.

utility of the generated data in evaluating reasoning systems. Detailed results will be added to the GitHub repository[8].

6 Related Work

Bonifati et al. [5] discuss the efforts related to graph data generation in detail. They identified 35 data generators classified based on the domain (Social Media, Academic, e-commerce, or User Defined), their generation capabilities (fixed, schema-driven, extracted), and data distribution (and other dimensions out of the scope of this work). Being GenAct, a fixed, academic, and social media-based RDF data generator that follows realistic Twitter data, we discuss its relation with the closest ones selected by such survey and additional works published afterward. To this extent, we divide the selected generators into the following sub-areas: (Knowledge) Graph Generation and time-related data generation.

Linked Data. In the Semantic Web context, data generation is commonly associated with benchmarking. **WatDiv** [2] focuses on stress testing, i.e., it is workload-oriented. It provides a data generator that generates scalable datasets according to the provided schema. Temporality, dynamicity, and timeliness are not considered in the dataset or the workload. The **SP^{2}Bench** [28] is based on the DBLP dataset. The generated RDF graph follows the same data distribution as the original DBLP dataset, mimicking entity relations and correlations. A fixed seed ensures that the data generation process is deterministic. The final data contains a temporal dimension related to the publication date. However, this is not prioritized in the generation or related queries. **LinkGen** [16] can generate data in RDF for a given vocabulary. It supports Zipf's power-law and Gaussian distributions for entity generation. Being agnostic to the vocabulary, LinkGen could be tuned to generate temporal data. Moreover, the generation can be performed in streaming. Nevertheless, it does not support stream partitioning, nor can it describe absolute events in the vocabulary. In future work, we plan to compare the efficiency of GenAct generation with LinkGen.

Streaming Linked Data. Among the numerous benchmarks [25,35,10,23,17,32,1,9], only LSBench, SLUBM, and LASS provide a way to stress stream reasoning systems and data generators. The former two approach the generation process naively, considering a single data stream. While LSBench shares GenAct's domain of social media and the static-streaming data duality, it lacks accurate Twitter templates and realistic data distribution. On the other hand, SLUBM streams LUBM [12] individuals without considering data distribution or partitioning. Additionally, RSPLab [34], a framework for testing RDF stream processing systems, incorporates a solution for generating data based on TripleWave [19]. Although RSPLab claims the ability to study the RSP response to change in the input frequency, such a feature does not appear in the code.
Reasoning. The benchmarks designed for highly expressive ontology reasoners [12,18,31] are limited to static data, making them unsuitable for mimicking domains that require reasoning over temporality, dynamicity, and timeliness. This latter gap was attempted to be bridged by RSP benchmarks, like LASS 1.0 [32], but its adoption seems minimal. **PyGraft** [15] is a tool for KG generation written in Python. It is domain-agnostic but supports a subset of OWL 2 and RDF language features. Data generation is pipelined with a DL reasoner to ensure consistency. Although PyGraft allows users to implement their own conceptualization, it does not include any explicit temporal feature nor consider dynamicity and timeliness, making it unsuitable for studying evolving Knowledge Graphs. **GDD**$_x$ [11] is a schema-driven graph generator based on Extended Graph Differential Dependencies, i.e., an extension of graph entity/differential dependencies that represent formal constraints for graph data. However, it cannot generate domain-agnostic schemas, so an existing schema is required as input.

7 Conclusion

This paper presents GenACT, a temporal web data generator based on ACE that can generate data that has temporal, dynamic, and timeliness characteristics. These are critical for evaluating systems on KG evolution and stream reasoning, two areas that do not have realistic and large, publicly available data. GenACT comes with a novel ontology for each of the OWL 2 profiles, allowing it to stress different reasoning systems. The generation process allows for the creation of different data segments using SPARQL-based partitioning strategies, allowing users to define custom partitioning strategies.

Our synthetic data generation approach emphasizes adaptability across multiple dimensions. Future plans include expanding to platforms like Reddit and LinkedIn, incorporating the Shapes Constraint Language (SHACL)[11] for enforcing temporal constraints and data validation and extending data generation beyond RDF, e.g., Property Graphs or Datalog, to encourage the benchmarking of expressive reasoners [4] and graph stream processing languages [27].

[11] https://www.w3.org/TR/shacl/

References

1. Ali, M.I., Gao, F., Mileo, A.: Citybench: A configurable benchmark to evaluate RSP engines using smart city datasets. In: ISWC 2015, Bethlehem, PA, USA, October 11-15, 2015. vol. 9367, pp. 374–389. Springer (2015)
2. Aluç, G., Hartig, O., Özsu, M.T., Daudje, K.: Diversified Stress Testing of RDF Data Management Systems. In: The Semantic Web – ISWC 2014. pp. 197–212. Springer International Publishing, Cham (2014)
3. Bauer, A., Trapp, S., Stenger, M., Leppich, R., Kounev, S., Leznik, M., Chard, K., Foster, I.T.: Comprehensive exploration of synthetic data generation: A survey. CoRR **abs/2401.02524** (2024), `https://doi.org/10.48550/arXiv.2401.02524`
4. Beck, H., Dao-Tran, M., Eiter, T.: LARS: A logic-based framework for analytic reasoning over streams. Artif. Intell. **261**, 16–70 (2018). https://doi.org/10.1016/J.ARTINT.2018.04.003, `https://doi.org/10.1016/j.artint.2018.04.003`
5. Bonifati, A., Holubová, I., Prat-Pérez, A., Sakr, S.: Graph generators: State of the art and open challenges. ACM Comput. Surv. **53**(2), 36:1–36:30 (2021), `https://doi.org/10.1145/3379445`
6. Bonifati, A., Tommasini, R.: An overview of continuous querying in (modern) data systems. In: Barceló, P., Sánchez-Pi, N., Meliou, A., Sudarshan, S. (eds.) Companion of the 2024 International Conference on Management of Data, SIGMOD/PODS 2024, Santiago AA, Chile, June 9-15, 2024. pp. 605–612. ACM (2024). https://doi.org/10.1145/3626246.3654679, `https://doi.org/10.1145/3626246.3654679`
7. Bonte, P., Calbimonte, J., de Leng, D., Dell'Aglio, D., Valle, E.D., Eiter, T., Giannini, F., Heintz, F., Schekotihin, K., Phuoc, D.L., Mileo, A., Schneider, P., Tommasini, R., Urbani, J., Ziffer, G.: Grounding stream reasoning research. TGDK **2**(1), 2:1–2:47 (2024). https://doi.org/10.4230/TGDK.2.1.2, `https://doi.org/10.4230/TGDK.2.1.2`
8. Bonte, P., Ongenae, F., Tommasini, R.: A holistic view over ontologies for streaming linked data. Semantic Web (To Appear) (2024)
9. Bonte, P., Turck, F.D., Ongenae, F.: Towards an evaluation framework for expressive stream reasoning. In: Verborgh, R., Dimou, A., Hogan, A., d'Amato, C., Tiddi, I., Bröring, A., Maier, S., Ongenae, F., Tommasini, R., Alam, M. (eds.) The Semantic Web: ESWC 2021 Satellite Events - Virtual Event, June 6-10, 2021, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12739, pp. 76–81. Springer (2021), `https://doi.org/10.1007/978-3-030-80418-3_14`
10. Dell'Aglio, D., Calbimonte, J., Balduini, M., Corcho, Ó., Della Valle, E.: On correctness in RDF stream processor benchmarking. In: ISWC 2013, Sydney, NSW, Australia, October 21-25, 2013. Lecture Notes in Computer Science, vol. 8219, pp. 326–342. Springer (2013)
11. Feng, Z., Mayer, W., He, K., Kwashie, S., Stumptner, M., Grossmann, G., Peng, R., Huang, W.: A schema-driven synthetic knowledge graph generation approach with extended graph differential dependencies (gdd$^{\mathrm{X}}$s). IEEE Access **9**, 5609–5639 (2021)
12. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. J. Web Semant. **3**(2-3), 158–182 (2005)
13. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)

14. Horrocks, I.: OWL: A Description Logic Based Ontology Language. In: van Beek, P. (ed.) Principles and Practice of Constraint Programming, Sitges, Spain, October 1-5, 2005. Lecture Notes in Computer Science, vol. 3709, pp. 5–8. Springer (2005)
15. Hubert, N., Monnin, P., d'Aquin, M., Brun, A., Monticolo, D.: Pygraft: Configurable generation of schemas and knowledge graphs at your fingertips. CoRR **abs/2309.03685** (2023), https://doi.org/10.48550/arXiv.2309.03685
16. Joshi, A.K., Hitzler, P., Dong, G.: Linkgen: Multipurpose linked data generator. In: Groth, P., Simperl, E., Gray, A.J.G., Sabou, M., Krötzsch, M., Lécué, F., Flöck, F., Gil, Y. (eds.) The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9982, pp. 113–121 (2016), https://doi.org/10.1007/978-3-319-46547-0_12
17. Kolchin, M., Wetz, P., Kiesling, E., Tjoa, A.M.: Yabench: A comprehensive framework for RDF stream processor correctness and performance assessment. In: Bozzon, A., Cudré-Mauroux, P., Pautasso, C. (eds.) Web Engineering - 16th International Conference, ICWE 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9671, pp. 280–298. Springer (2016), https://doi.org/10.1007/978-3-319-38791-8_16
18. Ma, L., Yang, Y., Qiu, Z., Xie, G.T., Pan, Y., Liu, S.: Towards a complete OWL ontology benchmark. In: Sure, Y., Domingue, J. (eds.) The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4011, pp. 125–139. Springer (2006), https://doi.org/10.1007/11762256_12
19. Mauri, A., Calbimonte, J., Dell'Aglio, D., Balduini, M., Brambilla, M., Della Valle, E., Aberer, K.: Triplewave: Spreading RDF streams on the web. In: Groth, P., Simperl, E., Gray, A.J.G., Sabou, M., Krötzsch, M., Lécué, F., Flöck, F., Gil, Y. (eds.) The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9982, pp. 140–149 (2016), https://doi.org/10.1007/978-3-319-46547-0_15
20. Monfardini, G.K.Q., Salamon, J.S., Barcellos, M.P.: Use of competency questions in ontology engineering: A survey. In: Almeida, J.P.A., Borbinha, J., Guizzardi, G., Link, S., Zdravkovic, J. (eds.) Conceptual Modeling. pp. 45–64. Springer Nature Switzerland, Cham (2023)
21. Morsey, M., Lehmann, J., Auer, S., Ngomo, A.C.: DBpedia SPARQL Benchmark–Performance Assessment with Real Queries on Real Data. In: The Semantic Web – ISWC 2011. pp. 454–469 (2011)
22. Nenov, Y., Piro, R., Motik, B., Horrocks, I., Wu, Z., Banerjee, J.: Rdfox: A highly-scalable RDF store. In: Arenas, M., Corcho, Ó., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Staab, S. (eds.) The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9367, pp. 3–20. Springer (2015), https://doi.org/10.1007/978-3-319-25010-6_1
23. Nguyen, T.N., Siberski, W.: SLUBM: an extended LUBM benchmark for stream reasoning. In: Celino, I., Della Valle, E., Krötzsch, M., Schlobach, S. (eds.) Proceedings of the 2nd International Workshop on Ordering and Reasoning, OrdRing 2013, Co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 22nd, 2013. CEUR Workshop Proceedings,

vol. 1059, pp. 43–54. CEUR-WS.org (2013), https://ceur-ws.org/Vol-1059/ordring2013-paper6.pdf

24. Parra, D., Trattner, C., Gómez, D., Hurtado, M., Wen, X., Lin, Y.: Twitter in academic events: A study of temporal usage, communication, sentimental and topical patterns in 16 computer science conferences. Comput. Commun. **73**, 301–314 (2016), https://doi.org/10.1016/j.comcom.2015.07.001
25. Phuoc, D.L., Dao-Tran, M., Pham, M., Boncz, P.A., Eiter, T., Fink, M.: Linked stream data processing engines: Facts and figures. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part II. Lecture Notes in Computer Science, vol. 7650, pp. 300–312. Springer (2012), https://doi.org/10.1007/978-3-642-35173-0_20
26. Polleres, A., Pernisch, R., Bonifati, A., Dell'Aglio, D., Dobriy, D., Dumbrava, S., Etcheverry, L., Ferranti, N., Hose, K., Jiménez-Ruiz, E., Lissandrini, M., Scherp, A., Tommasini, R., Wachs, J.: How does knowledge evolve in open knowledge graphs? TGDK **1**(1), 11:1–11:59 (2023), https://doi.org/10.4230/TGDK.1.1.11
27. Rost, C., Tommasini, R., Bonifati, A., Valle, E.D., Rahm, E., Hare, K.W., Plantikow, S., Selmer, P., Voigt, H.: Seraph: Continuous queries on property graph streams. In: Tanca, L., Luo, Q., Polese, G., Caruccio, L., Oriol, X., Firmani, D. (eds.) Proceedings 27th International Conference on Extending Database Technology, EDBT 2024, Paestum, Italy, March 25 - March 28. pp. 234–247. OpenProceedings.org (2024). https://doi.org/10.48786/EDBT.2024.21, https://doi.org/10.48786/edbt.2024.21
28. Schmidt, M., Hornung, T., Lausen, G., Pinkel, C.: SP2Bench: A SPARQL Performance Benchmark, pp. 371–393. Springer Berlin Heidelberg (2010)
29. Shin, J., Jian, L., Driscoll, K., Bar, F.: The diffusion of misinformation on social media: Temporal pattern, message, and source. Comput. Hum. Behav. **83**, 278–287 (2018), https://doi.org/10.1016/j.chb.2018.02.008
30. da Silva Quirino, G.K., Salamon, J.S., Barcellos, M.P.: Use of competency questions in ontology engineering: A survey. In: Almeida, J.P.A., Borbinha, J., Guizzardi, G., Link, S., Zdravkovic, J. (eds.) Conceptual Modeling - 42nd International Conference, ER 2023, Lisbon, Portugal, November 6-9, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14320, pp. 45–64. Springer (2023), https://doi.org/10.1007/978-3-031-47262-6_3
31. Singh, G., Bhatia, S., Mutharaju, R.: Owl2bench: A benchmark for OWL 2 reasoners. In: Pan, J.Z., Tamma, V.A.M., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (eds.) The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12507, pp. 81–96. Springer (2020), https://doi.org/10.1007/978-3-030-62466-8_6
32. Tommasini, R., Balduini, M., Della Valle, E.: Towards a benchmark for expressive stream reasoning. In: Calbimonte, J., Dao-Tran, M., Dell'Aglio, D., Phuoc, D.L., Saleem, M., Usbeck, R., Verborgh, R., Ngomo, A.N. (eds.) Joint Proceedings of the 2nd RDF Stream Processing (RSP 2017) and the Querying the Web of Data (QuWeDa 2017) Workshops co-located with 14th ESWC 2017 (ESWC 2017), Portoroz, Slovenia, May 28th - to - 29th, 2017. CEUR Workshop Proceedings, vol. 1870, pp. 26–36. CEUR-WS.org (2017), https://ceur-ws.org/Vol-1870/paper-03.pdf

33. Tommasini, R., Bonte, P., Ongenae, F., Valle, E.D.: RSP4J: an API for RDF stream processing. In: Verborgh, R., Hose, K., Paulheim, H., Champin, P., Maleshkova, M., Corcho, Ó., Ristoski, P., Alam, M. (eds.) The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12731, pp. 565–581. Springer (2021). https://doi.org/10.1007/978-3-030-77385-4_34, https://doi.org/10.1007/978-3-030-77385-4_34
34. Tommasini, R., Della Valle, E., Mauri, A., Brambilla, M.: Rsplab: RDF stream processing benchmarking made easy. In: d'Amato, C., Fernández, M., Tamma, V.A.M., Lécué, F., Cudré-Mauroux, P., Sequeda, J.F., Lange, C., Heflin, J. (eds.) The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10588, pp. 202–209. Springer (2017), https://doi.org/10.1007/978-3-319-68204-4_21
35. Zhang, Y., Pham, M., Corcho, Ó., Calbimonte, J.: Srbench: A streaming RDF/SPARQL benchmark. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I. Lecture Notes in Computer Science, vol. 7649, pp. 641–657. Springer (2012), https://doi.org/10.1007/978-3-642-35176-1_40