How Does Knowledge Evolve in Open Knowledge Graphs?

Polleres, Axel; Pernisch, Romana; Bonifati, Angela; Dell'Aglio, Daniele; Dobriy, Daniil; Dumbrava, Stefania; Etcheverry, Lorena; Ferranti, Nicolas; Hose, Katja; Jiḿenez-Ruiz, Ernesto; Lissandrini, Matteo; Scherp, Ansgar; Tommasini, Riccardo; Wachs, Johannes

Published in:
Transactions on Graph Data and Knowledge (TGDK)

DOI (link to publication from Publisher):
10.4230/TGDK.1.1.11

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

How Does Knowledge Evolve in Open Knowledge Graphs?

Axel Polleres
Vienna University of Economics and Business, Austria
Complexity Science Hub Vienna, Austria

Romana Pernisch[1]
Vrije Universiteit Amsterdam, The Netherlands
Discovery Lab, Elsevier, Amsterdam, The Netherlands

Angela Bonifati
Lyon 1 University, CNRS LIRIS, France
IUF, France

Daniele Dell'Aglio
Aalborg University, Denmark

Daniil Dobriy
Vienna University of Economics and Business, Austria

Stefania Dumbrava
ENSIIE, France
SAMOVAR, IP Paris, France

Lorena Etcheverry
Universidad de la República, Uruguay

Nicolas Ferranti
Vienna University of Economics and Business, Austria

Katja Hose
TU Wien, Austria

Ernesto Jiménez-Ruiz
City, University of London, UK
SIRIUS, University of Oslo, Norway

Matteo Lissandrini
Aalborg University, Denmark

Ansgar Scherp
Ulm University, Germany

Riccardo Tommasini
INSA Lyon, CNRS LIRIS, France

Johannes Wachs
Corvinus University of Budapest, Hungary
Centre for Economic and Regional Studies, Hungary
Complexity Science Hub Vienna, Austria

Abstract

Openly available, collaboratively edited Knowledge Graphs (KGs) are key platforms for the collective management of evolving knowledge. The present work aims t o provide an analysis of the obstacles related to investigating and processing specifically this central aspect of *evolution* in KGs. To this end, we discuss (i) the dimensions of evolution in KGs, (ii) the observability of evolution in existing, open, collaboratively constructed Knowledge Graphs over time, and (iii) possible metrics to analyse this evolution. We provide an overview of relevant state-of-the-art research, ranging from metrics developed for Knowledge Graphs specifically to potential methods from related fields such as network science. Additionally, we discuss technical approaches – and their current limitations – related to storing, analysing and processing large and evolving KGs in terms of handling typical KG downstream tasks.

2012 ACM Subject Classification Information systems → Graph-based database models; Information systems → Data streaming; Information systems → Web data description languages

Keywords and phrases KG evolution, temporal KG, versioned KG, dynamic KG

Digital Object Identifier 10.4230/TGDK.1.1.11

Category Survey

Funding *Axel Polleres*: supported by the European Union's Horizon 2020 research and innovation program under grant agreement No 957402 (Teaming.AI).
Daniil Dobriy: supported by Jubiläumsfonds der Stadt Wien (project: Collaborative Knowledge Management through Knowledge Graphs: The Case of Managing Historical Knowledge about Vienna).

[1] Corresponding author

Ernesto Jiménez-Ruiz: supported by the SIRIUS Centre for Scalable Data Access (Research Council of Norway, project 237889).
Johannes Wachs: supported by the Center for Collective Learning (101086712-LearnData-HORIZON-WIDERA-2022-TALENTS-01 financed by EUROPEAN RESEARCH EXECUTIVE AGENCY (REA))
Acknowledgements We would like to thank the reviewers for their invaluable comments that helped to improve our manuscript.
Received 2023-06-30 **Accepted** 2023-11-17 **Published** 2023-12-19
Editors Aidan Hogan, Ian Horrocks, Andreas Hotho, and Lalana Kagal
Special Issue Trends in Graph Data and Knowledge

1 Introduction

Knowledge Graphs (KGs) [112] are graph-structured representations intended to capture the semantics about how entities relate to each other, used as a general tool for the symbolic representation and integration of knowledge in a structured manner. The actual *semantics* or schema of such graphs can be formally described using expressive logic-based languages such as the Web Ontology Language (OWL) [101], as well as in terms of constraint languages such as the Shapes Constraint Language (SHACL) [135] or Shape Expressions (ShEx) [195]. Thanks to the expressivity provided by such formalisations, KGs have become a de-facto standard data model for integrating information across organisations and public institutions. It also facilitates the collaborative construction of structured knowledge on the Web by dispersed communities. In other words, KGs serve as intermediate layers of abstraction between raw data and decision support systems. Raising the level of abstraction has allowed us to ask more sophisticated questions, integrate data from heterogeneous sources, and spark collaborations between groups with different perspectives and views on business problems.

As a result of their function as a basis for knowledge integration, KGs are rarely produced in a single one-shot process. Instead, KGs are often collaboratively built and accessed over time. As such, KGs have become a significant driver for the collaborative management of evolving knowledge, integrating knowledge provided by different actors and multiple stakeholders: use cases range from the collaborative collection of factual base knowledge in general-purpose Open KGs such as Wikidata [242] to capturing specialised collaborative knowledge about engineering processes in manufacturing [110].

However, the sheer scale of – in particular – openly available, collaborative KGs has exacerbated the challenge of managing their evolution, be it in terms of (i) the size and temporal nature of the data, (ii) heterogeneity and evolution of the communities of their contributors, or (iii) the development of information, knowledge, and semantics captured within these graphs over time.

Even though analysis of the content, nature, and quality of KGs has already attracted a vast amount of research (i. e. [192, 104, 202] and references therein), these works focus less on how their structure and contents change over time, indeed how these systems evolve.

With the present article, we aim to shift the focus on precisely this matter. In particular, we try to answer the following main questions:

RQ1 *Which publicly accessible, open KGs are observable in a manner that would allow a longitudinal analysis of their evolution and how?* That is, how could we obtain historical data about their development, or which infrastructures and techniques would we need to monitor their growth and changes in the future?

RQ2 *Which metrics could be used to compare the evolution and structure over time, and how could existing static metrics be adapted accordingly?* Here, we are particularly interested in approaches from other adjacent fields, such as network science, and how those could be adapted and applied to specifically analyse the evolution of knowledge graphs.

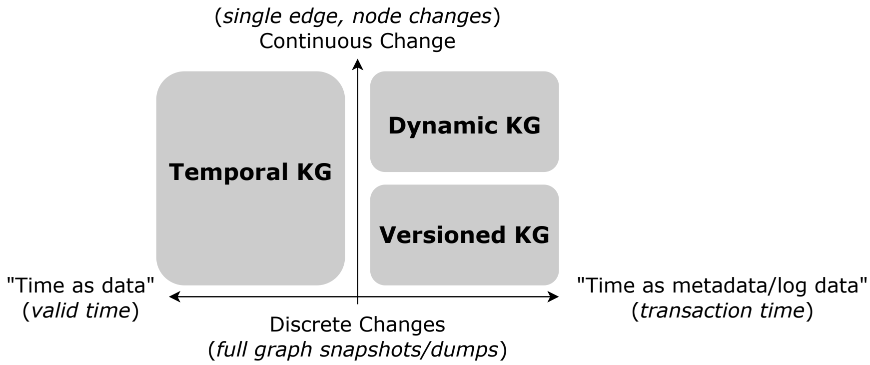

Figure 1 Perspectives on evolving knowledge graphs. We distinguish between considering time as data or metadata (x-axis) vs. at which "granularity" changes range from full static "bulk" snapshots at different points in time to continuous reporting of changes at the level of single edges or nodes (y-axis). We explicitly note that these perspectives are *not* mutually exclusive: the visualisation does not capture the possible overlaps of these perspectives, nor the nature of *how* changes are produced and captured, e. g. in terms of underlying collaborative processes.

RQ3 Finally, *do we have the right techniques to process evolving KGs*, both in terms of scaling monitoring and computing the necessary metrics, but also in terms of enabling longitudinal queries, or other downstream tasks such as reasoning and learning in the context of change – facing the rapid growth and evolution of existing KGs?

To approach these questions, the remainder of this article surveys existing approaches and works and raises open questions in four directions: observing, studying, managing and spreading KG evolution. Before elaborating on these directions, we first discuss the different *dimensions of evolution* in Section 2, introducing relevant terminology. In Section 3, we discuss to what extent data about the evolution of open KGs (like Wikidata or DBpedia) is available and what evolution trends have been observed so far in prior literature. In Section 4, we discuss different types of metrics to study evolving KGs; starting from state-of-the-art graph and ontology metrics, we also discuss metrics related to quality and consistency, as well as potentially valuable works and metrics from the area from network science. In Section 5, we discuss data management problems for evolving knowledge graphs, i. e. data models that capture temporality as well as storage approaches and schema mappings for versioned and dynamic KGs. In Section 6, we focus on downstream tasks on KGs in the specific context of evolution. More precisely, we discuss how querying, reasoning, and learning approaches can be tailored for evolving KGs. We also address the exploration of KGs, an essential aspect of evolving KGs. We conclude with a summary of the main research challenges we currently see unaddressed (or only partially addressed) in Section 7.

2 Dimensions of Evolution

The temporal evolution of graphs, knowledge graphs (KGs), and collaboratively edited KGs has multiple dimensions that we outline in this section, along with relevant terminology. That is to say, there are multiple coherent perspectives we can use to talk about the "*evolution*" of KGs, ranging from considering time and evolution as being part of the data itself to considering evolution and change over time on a meta-level. We illustrate these perspectives in Figure 1.

Temporal KGs: Time as data

The first perspective considers time, or – more concretely, the temporal validity of information in a KG – as part of the KG itself; we call this the "Temporal KG" perspective. In this context, the evolution depicted by the data pertains to the changes in the "world" it represents, not the evolution of the data itself. Following database terminology, this temporal validity of information in a KG is typically referred to as *valid time*; see, for instance, [103]. A very simple example of a temporal KG is illustrated in Figure 2, which contains the year of production of Picasso's "Guernica", as a slightly simplified subgraph DBpedia [146].[1]

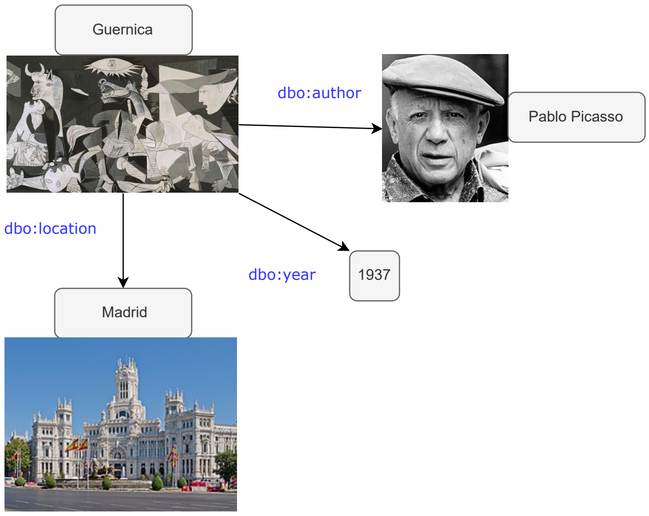

Figure 2 A simple KG containing temporal information as data (literal).

Time and temporality may be represented with a single temporal literal – as illustrated here a year or a timestamp, or likewise an interval: for instance, the production of "Guernica" itself was not a one-shot process, but its painting took place over a longer period. For instance, the production period of "Guernica" was carried out between 1937-05-01 and 1937-06-04, as illustrated in Figure 3, a simplified graph inspired by the Linked Art project.[2]

We note here that capturing intervals typically requires extensions of the "flat" directed labelled graph model used to represent simple knowledge graphs, as shown in Figure 3: contextual information about simple statements (such as in this case, the start and end time of a production interval), can be modelled in various ways, either

1. in terms of adding intermediate nodes to a flat graph model, also often referred to as "reification", or alternatively
2. in terms of bespoke, extended graph models such as so-called *property graphs*

Let us refer to Section 5.1 for a more in-depth discussion of different data models to capture time and temporality in KGs.

Time-varying KGs: Time as meta-data

The second perspective on evolution is scoped by the time granularity of change in the KG itself; in other words, by how the temporal aspect of the data, i. e. nodes, edges, and structure, of the KG is evolving. We call this the "Time-varying KG" perspective. Again, using database terminology, such changes in data are typically referred to as *transaction time* [103].

[1] `https://www.dbpedia.org/`
[2] `https://linked.art/model/`

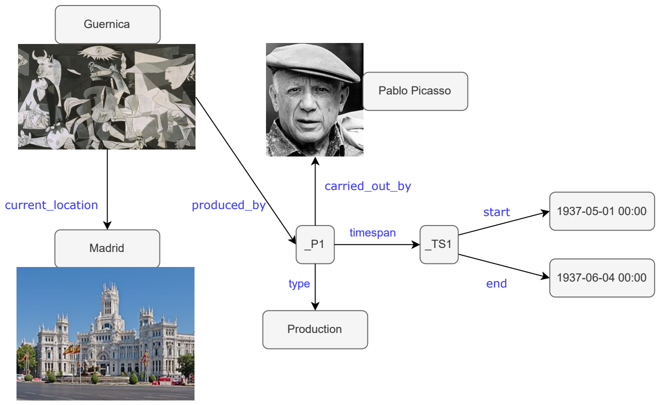

Figure 3 A KG containing temporal information in terms of intervals using a simple form of reification.

We present an example from the arts. Paintings like "Guernica" and information about their artists and other attributes have been added dynamically to Knowledge Graphs like Wikidata over time. The entry for "Guernica" (Q175036) in the Wikidata [242] KG was created on 28 November 2012,[3] while its creator "Pablo Picasso" (Q5593) was added on 1 November 2012[4]. Of course, both of these dates are independent of the birth or production dates of the referred entities themselves. As we will further discuss in Section 3 and also Section 5 below, the granularity and manner of how such changes are stored affect the observability and analysis of a KG's evolution.

In terms of granularity, we can differentiate between two types of knowledge graphs based on how they are stored:

- Dynamic KGs - which allow access to all observable atomic *changes* in the knowledge graph.
- Versioned KGs - which provide *static snapshots of the materialised state of the knowledge graph at specific points in time.*

These represent opposite ends of the granularity spectrum. Figures 4 and 5 show two examples of how the changing information regarding the location of "Guernica" over time[5] could be represented in terms of versions or dynamic changes, respectively.

For instance, as discussed above, Wikidata embodies continuous change, accessible through the entities' edit histories at the level of real-time modifications. At the same time, DBpedia represents both the spectrum's discrete end, releasing snapshot updates,[6] as well as offering small-scale releases with DBpedia Live[7] on minute level. Observe that in both cases, the temporal information about neither the materialisation time of a DBpedia snapshot or the edits of single statement claims on Wikidata are available in terms of the (RDF) graph materialisations of these KGs themselves, but only in terms of the publication metadata or edit histories, which is why we may also speak of "time as meta-data".

We note that this distinction is hardly clear-cut. The difference between dynamic and versioned temporalities is marked by the technical means by which particular KGs evolve. In particular, this boundary is shaped by differences in technical infrastructures supporting these evolutionary processes rather than general characteristics of the KG and the kind of knowledge it captures.

---

3 `https://www.wikidata.org/w/index.php?title=Q175036&action=history&dir=prev`
4 `https://www.wikidata.org/w/index.php?title=Q5593&action=history&dir=prev`
5 The painting was first exhibited in Paris in 1937, and moved to an exhibition in New York in 1939. Since 1992 "Guernica" is displayed in Museo Reina Sofía in Madrid.
6 `https://www.dbpedia.org/resources/snapshot-release/`
7 `https://www.dbpedia.org/resources/live/`

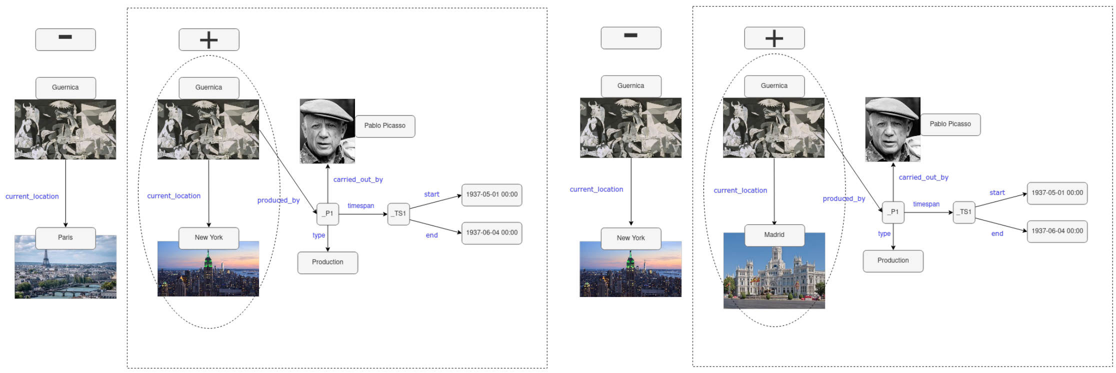

Figure 4 Representation of Dynamic Knowledge Graph, with updates at edge level, i. e. deletions (left) and future additions (right).

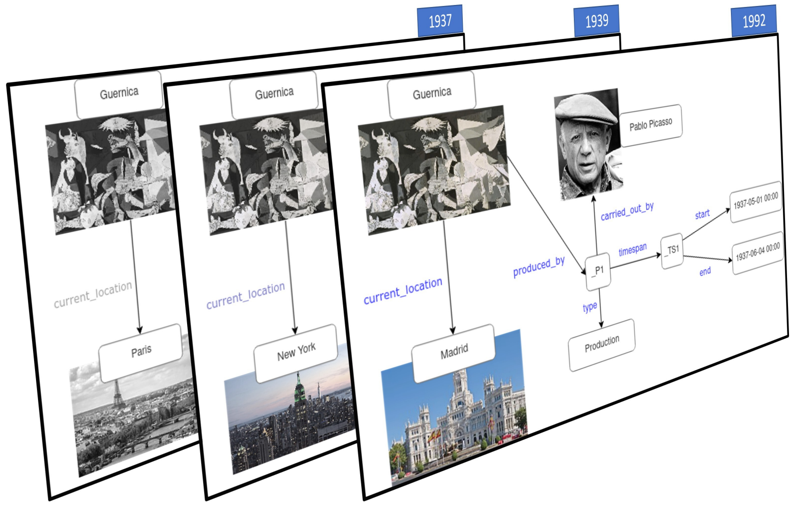

Figure 5 Representation of Versioned Knowledge Graph, with snapshots sorted by time.

For example, on the one hand, while changes in Wikidata may be recorded down to the level of single statements, Wikibase[8] also supports interfaces for bulk updates. Likewise, each single statement change in Wikidata's overall edit history may be theoretically materialised in terms of sequential snapshots. On the other hand, DBpedia's extraction framework constructing a KG from Wikipedia may be analogously applied to any materialised point in time of the fine-granular page edit history of Wikipedia, or even per page [80]. DBpedia's model has also changed over the past years from irregular, approximately annual, snapshots published in its beginnings, to enable more dynamic publishing (monthly) cycles [111] through the DBpedia Databus.[9]

Lastly, we note that analogously to the examples in Figures 2 and 3 both *timestamps* and *time intervals* can be used to represent not only validity but also transaction and versions, i. e. snapshots of the entire graph in the context of KGs. However, depending on which dimension is considered, it will have an impact on how data should be managed, whether evolution is observable, and how the information about evolution is spread into downstream tasks, see the further discussions in Sections 5 and 6 below.

---

[8] `https://wikiba.se`, Wikidata's underlying software framework.
[9] `https://www.dbpedia.org/resources/databus/`

Both of the aforementioned perspectives can serve the purpose of monitoring the evolution of KGs along different yet interrelated (sub-)dimensions. We outline these dimensions in the following subsections. First, according to Section 2.1, the *structural evolution* of KGs can be observed through the temporal information captured in them; here, KGs present a distinction between changes on the data and schema levels. Second, one can analyse the dynamics or velocity of evolution in KG over time, see Section 2.1. Finally, when considering the collaborative processes involved in KG editing and evolution, one can analyse the structure and dynamics of these collaborations, see Section 2.2. After exploring these dimensions in detail, we then discuss concrete metrics in Section 4.

2.1 Structural Evolution, Dynamics, Timeliness, and Monotonicity

In the context of evolving KGs (hereafter EKGs), we may consider different forms of change related to the graph structure, dynamics of change or its nature (monotonic or with deletions), and alternative notions of time. The following will briefly elaborate on our running example in Figure 6.

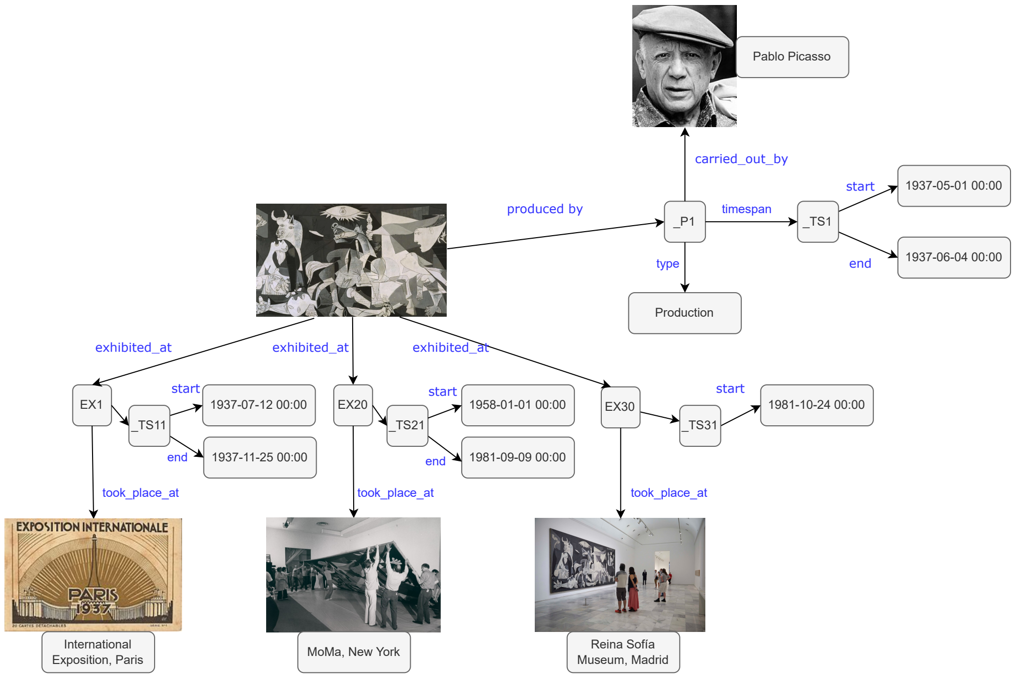

Figure 6 A sample KG containing temporal information about the production (static) and exhibitions (dynamic) of paintings.

Structural Evolution

The first dimension to measure on a graph is essentially related to its structure: descriptive statistics about nodes and edge distributions, centrality, connectedness, density, and modularity. In KGs, similar static metrics can also be observed concerning the *schema*, typically the node and edge *types*, and – if additionally axiomatic knowledge on the schema-level is considered – the complexity of this schema.

For all of these structural properties (both on the instance-level and schema-level), we may also be interested in their development over time, i. e. in quantifying their changes. The existing concrete metrics for this dimension will be discussed in more detail in Section 4 below.

Notably, longitudinal investigations of structural properties are not restricted to the time-varying KG perspective: depending on whether temporal information is present in the KG itself, one may also be interested in analysing and comparing structural evolution in terms of "temporal slices".

Dynamics

Dynamics for KGs refers to characteristics such as growth and change frequencies over time and per time interval). These may be observed overall but also in terms of subgraphs or topic-wise components of a KG. For instance, one may consider comparing the change dynamics of entities related to different topic areas, such as "arts" and "sports" within a particular KGs like Wikidata. Again, these dynamics may be observed concerning the KG schema. Referring to a concrete elaboration of our running example in Figure 6, we can derive that properties related to the *production* of paintings evolve more slowly than properties relating to exhibitions. Notably, dynamics and temporal granularity may again be compared and analysed both from secular and time-varying perspectives.

Timeliness

Timeliness, from a data quality perspective, refers to the "freshness" of the data concerning the occurrence of change, the current time, or the time of processing. Timeliness directly links to query answering (or processing in general), as it establishes the value of the retrieved answer considering some requirements. More specifically, the timeliness of data in a KG can be interpreted as

- "out-of-date" or "stale" information: i. e. in terms of recency of temporal information concerning the current time;
- "out-of-sync" or "delayed" information, i. e. in terms of the difference between valid times and transaction times of items in the KG, i. e. the interplay between these temporal and time-varying perspectives.

Regarding the former case, considering Figure 6, the question "Where is Guernica currently?" obtains a different answer at different times. While historical events such as the creation of "Guernica" lie far in the past, even far before Wikidata was founded, the location of paintings is an important dimension to analyse over time as it changes with exhibitions or purchases. If neglecting such variations is an issue for the users, e. g. when an accurate current location is needed to recommend a museum visit, then we witness a data quality problem related to timeliness.

A "drastic" example of the latter, i. e. extended out-of-sync information from the art domain is documented in Rembrandt's "Portrait of a Young Woman" (Q85523581 in Wikidata) from 1632, which was added to Wikidata only in February 2020, after it was recently confirmed to be an authentic Rembrandt.[10] Users who have asked for the number of Rembrandt paintings before 2020 would have received a stale answer.

Monotonicity

Monotonicity refers to the nature of changes, i. e. if they are positive changes only augmenting the content of the graphs, or if they take the form of an update which may include *deletions* of past information.

Continuing our examples in the domain of painting, we consider rectifying a painting's attribution to its artist, which happens repeatedly in arts. A documented case is the painting "Girl with a Flute" (Q3739200) in Wikidata, originally attributed to the Dutch painter Vermeer

[10] `https://news.artnet.com/art-world/pennsylvania-museum-rembrandt-discovery-1773954`

but later confirmed to be the work of another painter.[11] Similar, non-monotonic changes may arise when temporal information itself changes in the KG: imagine, following our running example, that subsequent research may reveal Guernica was actually created in 1936, not 1937.

From this combination of dynamics (i. e. the study of changes), timeliness, and monotonicity (i. e. the frequency of deletions and, therefore, errors and rectifications of incorrect information in a KG), it is also possible to estimate the frequency of future transactions. Together they form an essential dimension of evolving KGs, both in the context of the ability to process evolution technically but in terms of its impact on the validity of updated results of downstream tasks Section 6: as KGs are meant to support sophisticated decision-making tasks, it is often paramount to guarantee up-to-date information and provide answers before they become obsolete.

2.2 Evolution in Collaboration

Knowledge evolution is driven by different types of collaborations [190, 5]. As described by Piscopo et al. [190], collaborative KGs rely on experts for specific types of activities, defining rules and processes for how and by whom some activities should be carried out, or provide tools to facilitate such collaboration.

In the context of KG evolution, we may thus want to analyse the behaviours of single users or user groups over time. To classify the collaboration types, we can distinguish the following roles of users/agents:

- **Anonymous users:** These are *Users* who do not have a registered account or a consistent identity within a project (e. g. anonymous Wikibase users)
- **Registered users:** similarly, these are *Users* who have a registered account or a consistent identity within a project (e. g. registered Wikibase users), ideally also combined with additional information or characteristics which allow to classify such users (e. g. country of origin or other demographic attributes)
- **Authoritative users:** These are *Users* characterised by in-depth domain knowledge or knowledge engineering expertise. This group represents vetted knowledge engineers, domain experts, and moderators.
- **Bots:** These are automated agents performing recurring tasks (e. g. Wikibase bot accounts).

Longitudinal analyses of the contributions of such users may include changes in their behaviours (e. g. in terms of edit frequencies), interests (e. g. in terms of editing particular parts or topics of KGs), or role changes. Additionally, based on the aforementioned roles, various collaboration types can be potentially recognised when analysing the evolution of edits in collaboratively edited KGs [191]:

- **Expert-driven collaboration:** this type of collaboration involves *Authoritative users* developing schemas or editing data on the instance-level (creating mapping rules, as in the case of DBpedia, would be an example of such schema-level expert collaboration, whereas the instance data, origins from Wikipedia, thus following another collaboration model).
- **Crowd-sourced collaboration:** this type of collaboration involves many *Users* not considered *Authoritative users* performing basic editing tasks which neither requires in-depth domain or knowledge engineering expertise nor coordination between the editors (for instance, any users being allowed to edit Wikipedia could be understood as such a crowd-sourced collaboration model, if a more moderated process did not govern it, see below).

[11] `https://www.wikidata.org/w/index.php?title=Q3739200&oldid=803621750`

Table 1 Types of Collaboration in Open-domain Knowledge Graphs.

KG	Expert-driven	Crowd-sourced	Resource-dependent	Community-driven	Bot-assisted
Wikidata [242]		✓		✓	✓
DBpedia [146]	✓		✓		(✓)
YAGO [153]		✓			

- **Resource-dependent collaboration:** This type of collaboration is based on integrating information from external resources, potentially governed by different heterogeneous collaboration models (indeed, DBpedia's extraction of instance data from Wikipedia may be understood as such a resource-dependent "collaboration").
- **Community-driven collaboration:** this type of collaboration relies on self-moderating communities of *Users* characterised by deep involvement in the project, collective discussion, and decision making (e. g. Item/Property discussions characteristic for Wikidata, but also characteristic for the curation process in Wikipedia).
- **Bot-assisted collaboration:** this type of collaboration is characterised by *Bots* performing repetitive tasks alongside *Users* (i. e. curation tasks, e. g. checking property constraints on Wikidata, but also, indirectly in DBpedia, via bot interactions in Wikipedia, cf. [254, 50]).

Table 1 describes the common collaboration models of some existing, collaboratively maintained open general-purpose KGs, according to the literature. We note that the list of KGs shown here is not meant to be exhaustive and that such metrics could be further extended and refined in more fine-grained longitudinal analyses. As described in Section 4.4, for example, topologically identified groups of collaborators could be used to predict outcomes. A concrete methodology to analyse the composition of the collaborators within the KG and assess their effects on quality has been suggested in [189]. Further investigation can also include the different evolution and collaboration approaches and how these influence the possibility of analysing evolution. For example: does the relatively small DBpedia ontology and the limited frequency of updates via mapping changes make the analysis of the evolution of its ontology easier than the direct ontology editing model of Wikidata? Does the extraction and mapping mechanism and changes to the rules that drive them make ontology evolution in turn less flexible for the community in DBpedia? Likewise, does the free-for-all collaboration approach in Wikidata render a structured analysis of ontology evolution impossible, or what are the methods to handle this challenge? For instance (i) can one define "checkpoints" of limited changes that can be used as anchor points to produce useful analyses, or (ii) does it make sense to investigate the evolution of vocabularies specifically scoped to editors' sub-communities? Another avenue for investigation is a more effective utilisation of machine learning in supporting the collaborative evolution of KGs and their schemas. Specifically, it would be interesting to learn how this evolution is affected and affects the interaction of automated extraction (DBpedia), extraction by statistical learning (YAGO), or in leveraging or improving bots (Wikidata): that is, can ontology extraction rules or curation pipelines be improved by observing and learning from the collaboration and evolution processes over time?

2.3 Semantic Drift

Semantic drift is a crucial concept of evolution in language. It refers to the *change in meaning* of a concept over time [246, 218] independently from the downstream tasks like querying or reasoning. Before detecting semantic drift, one needs to identify the two concepts to compare between versions. Although early work on identifying semantic drift focused on the definition of the identity of a concept [246], when a concept changes meaning, it might also change its identifying information. Therefore, it is not always possible to rely only on identity-based approaches to understand semantic drift. In such cases, morphing chain-based strategies are more suitable [90]. The morphing chain approach presents the user with a comparison of a concept to all the concepts between the versions of an ontology and lets the user choose or chooses heuristically which is the most likely concept that a previous one evolved into.

For KGs, Meroño-Peñuela et al. [158] studied semantic drift in DBpedia concepts, while Stavropoulous et al. [219] studied semantic drift in the context of the Dutch Historical Consensus and the BBC Sports Ontology. SemaDrift [218] takes a morphing-chain approach, where three aspects are used to identify concepts that have potentially evolved from another: label, intention, and extension. The advantage of this approach is that every concept in a new version will have evolved from some previous concept. Unfortunately, the identity of concepts, such as URI, is not used in SemaDrift. OntoDrift [44] uses a hybrid approach and can be considered an extension of SemaDrift [218]. Additionally to using the label, intention, and extension aspects of concepts, it also considers the subclass relations. The drawback of this approach is that rules need to be defined for every type of predicate, as demonstrated by OntoDrift.

The notion of logical difference [136] between KGs can also be used to evaluate the semantic drift of the KG concepts. The logical difference focuses on the entailments or facts that follow from one KG but not from the other, and vice versa. Jiménez-Ruiz et al. [126] proposed an approach to evaluate the logical difference among different versions of the same ontology. Considering the new logical entailments/axioms involving a given entity, one could define a metric. The entity's role within the entailment (i. e. the entity is being defined vs. the entity referenced) may also impact the metric.

Potential approaches in the future could make additional use of embeddings, representing concepts in vector space and assessing their neighbourhoods. Pernisch et al. [181] showed that comparing two embeddings to each other is complex, and the similarity between concepts is, e. g. around 0.5 for FB15k-237 with TransE; Verkijk et al. [240] further discuss the difficulties with this approach, especially comparing it to concept shift in natural language. Finally, the lack of domain-specific benchmarks for semantic drift makes comparing methods difficult. For instance, OntoDrift and SemaDrift return very different numbers when detecting drift, but we cannot tell which ones are closer to the truth. Also, the number of studies that look at semantic drift is limited. Not many KGs have been studied, and even though the phenomenon is known, it has not been investigated extensively so far [158, 219].

3 Observe and Analyse the Evolution

This section discusses how far evolution can be observed and analysed along the dimensions defined above in various existing KGs. KGs come in very different flavours and structures, and in particular, we may also assume that their evolution shows very diverse characteristics.

Below, we first characterise different kinds of graphs. In Section 3.1, we discuss tools to observe the historical longitudinal data on the evolution of the most important existing KGs. Section 3.2 provides a respective overview of available studies to analyse and track the dynamics of some of these KGs. We consider both monitoring and analysing the evolution of the instance-level of graph data as well as the schema-level.

Without claiming completeness, we distinguish the following kinds of KGs:

- **General-purpose Open Knowledge Graphs**: publicly available open-domain (or, resp., cross-domain) KGs such as DPpedia [146] and Wikidata [242] as two of the most prominent KGs have been developed since more than a decade by now, covering a wide range of comprehensive knowledge. Yet, they differ fundamentally in the process in which knowledge is maintained and developed within the KG: whereas DBpedia relies on extractors to collect data from Wikipedia's infoboxes regularly, Wikidata comprises a completely collaboratively evolving schema and factbases that, by themselves, feed back into Wikipedia. In particular, we observed significant growth and dynamics in both the instance-level and schema-level of Wikidata over the past years. Collections of structured RDF data and microdata (e. g. schema.org [102] metadata) from Web pages through openly available Web crawls, such as made available regularly by the Webdatacommons[12] project [159], may indeed also be perceived as evolving, general purpose, real-world Knowledge Graphs.

- **Domain-specific Special-purpose Open Knowledge Graphs**: Many open knowledge graphs available to the public are often overlooked. These graphs are collaboratively developed and serve narrow, special-purpose topics or use cases. An example is Semantic MediaWiki (SMW)[138], which has been around for almost 20 years and is still actively developed and used in various community projects. SMW can be considered a predecessor of Wikibase, the underlying platform for Wikidata. Wikibase is increasingly being used in separate, special-purpose community projects. Other examples of domain-specific knowledge graphs include the UMLS Metathesaurus [34], as well as the ontologies in the OBO Foundry [121], and BioPortal [248]. These graphs focus on the schema and are assumed to have significantly different evolution characteristics [182].

- **Task-specific Knowledge Graphs**: One category of Knowledge Graphs that some authors identify is task-specific Knowledge Graphs [122]. These graphs, often used in benchmarks, are typically subsets of larger KGs created to support a specific application or may result from a downstream application (e. g. DBP15K as a subset of DBpedia for cross-lingual entity alignment). However, since these KGs are usually artificially limited and static (i. e. subset of specific snapshots), compared to real-world evolving KGs, we will not discuss them separately in this paper. We note, however, that principled approaches to *create evolving subsets* of KGs for specific benchmarking tasks are sorely needed to better understand these tasks "in evolution".

- **Large (and Small) Enterprise Knowledge Graphs** Lastly, we see many companies reportedly using and adopting Knowledge Graph technologies in their operations and businesses over the past years, including large firms like Google, Amazon, Facebook, and Apple, as well as many other smaller examples. What these KGs typically have in common is that due to their commercial value, they are non-observable to the community and we may only speculate about their sizes and structures using white papers [170, 209, 117], high-level announcements, and to some extent through industry track reports in conference series such as ISWC (e. g. [97]), SEMANTiCS (e. g. [204]), or recently the Knowledge Graph conference series. Given these limitations, we exclude enterprise KGs from the scope of the present paper.

Except for the latter two cases then, it appears that the research community has built up a large number of publicly accessible and observable KGs that vary in characteristics, and purpose, with unique communities of maintainers that seek to capture a rich variety of knowledge artefacts in evolving graph-like structures. In the remainder of this section, we specifically focus on Open General-purpose KGs rather than attempt to cover all types of KGs.

Table 2 Availability of Open KG Versions (V), Schema (S), and Change logs (CL).

	Level	Queryable	Collaborative	Formats	Protocol	Metadata	Temporality	Timeliness
Wikidata	V	Yes	Yes	NT, TTL, HDT, JSON	HTTP, SPARQL	schema.org	No	2-3 Days
	S	Yes	Yes	NT, TTL, JSON	HTTP, SPARQL	schema.org	No	2-3 Days
	CL	Yes	Yes	JSON	SSE	No	Event TS	Seconds
DBpedia	V	Yes	Partial	NT	HTTP, SPARQL	No	No	Quarterly
	S	Yes	Yes	RDF	HTTP, SPARQL	No	No	Daily
	CL	Yes	Yes	RDF	HTTP	No	Graph TS	Daily
YAGO	V	Yes	No	RDF all	HTTP	No	No	NA
	S	Yes	No	RDF	HTTP	No	No	NA
	CL	No	No			No		
LOD	V	Yes	Yes	RDF all	HTTP, SPARQL	Some	No	NA
	S	Yes	Yes	RDF, OWL	HTTP, SPARQL	No	No	NA
	CL	Depends on individual datasets						
GDELT	V	Yes	No	SQL Big Query	HTTP	No	Yes	15 min
	S	No	No	CSV, JSON, XML	HTTP	No	No	No
	CL	No	No	CSV	HTTP	No	Event TS	15 min

3.1 Availability of Graph Data

In the following, we start by assessing how and where historical longitudinal data about existing open KGs and their evolution can be found. We specifically focus on KGs that are still available and, therefore, do not include KGs like Freebase [36] and OpenCyc [156]. These two KGs are no longer maintained but are considered pioneering work and predecessors of the KGs investigated in this subsection. Therefore, it is generally possible for KGs to go dark, e. g. through neglect or malign actions.

Here, we give an overview of the datasets regarding the availability of their versions, their schema, and their changelogs in Table 2. The table captures if the versions, schema, or changelogs are queryable and collaborative. Queryable in this context captures if the KG answers queries in any way or form specifically over (historical) versions, schema as well as change logs, for which we then further specify the protocol (HTTP, SPARQL, etc.); for possible temporal queries over RDF archives that should be enabled over evolving KGs, we refer to, for instance, the categorisation in [84, Section 3.2]. Collaborativeness in Table 2 refers to the possibility of reconstructing user information on the different levels. For example, on the changelog level, a "yes" refers to having user information for individual changes. Wikidata and DBpedia allow anonymous edits, which potentially limits a reconstruction of the editing history, indicated with "Partial" in the table.

[12] `http://webdatacommons.org/`

Further information on formats (RDF, JSON, etc.) is given. Temporality refers to the ability of the KG to capture temporal information for example through reification or other means. With "Event TS", we indicate that the KG allows for events to be timestamped, whereas with "Graph TS", we refer to the whole graph having timestamps. Lastly, timeliness refers to how often the part of the KG is updated.

Wikidata is an open KG read and edited by humans and machines and is hosted by the Wikimedia Foundation. Intuitively, the considerable level of automation and collaboration on Wikidata, and its scale[13] present significant challenges in Wikidata evolution maintenance.

As for direct *queryability*, Wikidata's public SPARQL endpoint[14] provides query access to the current, regularly synced snapshot; it is undisputed that due to its scale, querying Wikidata in the light of its rapid growth – even on static snapshots – is currently reaching its limits in terms of regular SPARQL engines, as well documented for instance in [13]. Yet, there are various ways to access and potentially – given the respective infrastructure – query the historic versions and change data about Wikidata: **Wikidata Entities** dumps are available in JSON in a single JSON array, or RDF (using Turtle and N-triples) with Full RDF dumps are available for download[15] every 2-3 days, and historically for approximately a month. Schema.org metadata is used to describe the dump that contains additional helpful metadata such as the entity revision counter (schema:version), last modification time (schema:dateModified), and the link to the entity node with (schema:about).

As a subset, also *truthy dumps* are provided, which are limited to direct, truthy statements – since Wikidata offers (validTime) temporal annotations for statements, as well as provenance annotated statements, this "truthy" subset contains only currently valid or preferred ranked statements, where however additional metadata such as qualifiers, ranks, and references are consequently left out. The truthy dump could, therefore, be perceived as a "current truth" snapshot of Wikidata. In contrast, the entire dump also contains outdated (valid time) or disputed (in terms of being lower-ranked alternative statements by particular contributors).

RDF HDT[16] hosts roughly annual HDT [83] snapshots of Wikidata's complete dumps. In addition to these hosted RDF dumps, obtaining the statement-level change log from Wikidata's aggregated entity and editing history, which are also available via respective APIs, would be possible.

Finally, Wikimedia offers changes (of both Wikipedia and Wikidata) through the **Wikimedia Event Streams**[17] Web service that exposes continuous streams of JSON event data. It uses chunked transfer encoding following the Server-Sent Events protocol (SSE) and emits changes events, including Wikidata entity creations, updates, page moves, etc. The usage of edit history and event stream data, apart from RDF dumps, also has the advantage of making (where available) user/contributor information visible, which is helpful for collaboration analyses. Pelisser and Suchanek [225] have presented a prototype to provide this additional information in RDF via a SPARQL interface.

Wikidata Schema/Ontology. Wikidata does not follow a pre-defined formal ontology, meaning it does not formally differentiate between classes and instances. Instead, the terminology is derived from the relationships between the items in the graph and is collectively created by the editors. In other words, Wikidata (deliberately) does not make a formal commitment to the logical meaning of its properties and classes, which could be, for instance, roughly defined as the objects of the P31 (instance of) property.

[13] with over 15B triples at the time of writing: `https://w.wiki/7iez`.
[14] `query.wikidata.org`
[15] `https://dumps.wikimedia.org/wikidatawiki/entities/`
[16] `https://www.rdfhdt.org/datasets/`
[17] `https://stream.wikimedia.org/`

As a consequence, Wikidata's schema is evolving entirely in parallel with its data – and analogous considerations for the availability of data about its historic evolution apply as mentioned above. This has been reported to pose significant data quality challenges [190]; moreover, as a primary consequence of such an informal, collaborative process, Wikidata's ontology may change quickly. In practice, this does not impact the evolution of the graph itself, but it poses an obstacle to downstream tasks and analyses. We note that prior attempts to map the user-defined terminological vocabulary of Wikidata to RDFS and OWL, such as [105], could be used to partially map Wikidata to more standard ontology languages and conduct (approximate) analyses on a logical level. In contrast, we should note that theoretically, OWL/RDFS "mappable" properties could evolve independently in Wikidata.

DBpedia is an openly available KG encoded in RDF, which evolves alongside Wikipedia. It has four releases per year (approximately the 15th of January, April, June, and September, with a five-day tolerance), named using the same date convention as the Wikipedia Dumps that served as the basis for the release.[18] **DBpedia Latest Core Releases**[19] are published separately as small subsets of the total DBpedia release. Its extraction is fully automated using MARVIN [111] and then catalogued. The standard release is available on the 15th of each month, five days after Wikimedia releases Wikipedia dumps. **DBpedia Databus**[20] is a platform designed for data developers and consumers to catalogue and version data, not only restricted to DBpedia alone. It enables the smooth release of new data versions and promotes a shift towards more frequent and regular releases. DBpedia takes advantage of this functionality to promptly publish the most up-to-date DBpedia datasets, generating approximately 5,500 triples per second and 21 billion triples per release every month. **DBpedia Live**[21] is a changelog stream accessible in a pull manner. DBpedia Live monitors edits on Wikipedia and extracts the information of an article after it was changed. A synchronisation API is available to transfer updates to a dedicated online SPARQL endpoint, whereas temporal evolution as such is not directly queryable from that endpoint.

DBpedia Ontology (DBO), the core schema of DBpedia, is currently crowd-sourced by its community: *DBpedia mappings* are contributed and made automatically available daily, where DBO is generated every time changes in the mappings Wiki have been made. Notably, DBpedia Latest Core and DBpedia Live are based on the latest DBO snapshot available at the point of generation, i. e. one should consider the evolutions of data (Wikipedia edits), schema (mappings), and also the various releases of the actual DBpedia KG, separately.

Finally, we note that a fine-grained historical development, in terms of reproducing any DBpedia page at any point in time in the past, and thereby reconstructing a fine-grained RDF "history" would be theoretically possible by combining DBpedia's mappings with the Wikipedia edit history API. A prototypical implementation of this approach, the "DBpedia Wayback Machine" – inspired by the Web Archive's Wayback machine – has been presented by Fernández et al. [80].

YAGO is a large multilingual KG with general knowledge about people, cities, countries, movies, and organisations [220]. At the time of writing, there are six versions of YAGO. In its latest version, 4.5, YAGO combines Wikidata and Schema.org. Older versions integrate different sources such as Wikipedia, WordNet, and GeoNames but are independent of the most recent ones. YAGO places a strong emphasis on data extraction quality, achieving a precision rate of 95% through manual evaluation [198]. One of YAGO's unique features is its inclusion of spatial and temporal

[18] `https://www.dbpedia.org/resources/snapshot-release/`
[19] `https://www.dbpedia.org/resources/latest-core/`
[20] `https://databus.dbpedia.org/`
[21] `https://www.dbpedia.org/resources/live/`

information for many facts, enabling users to query the data across different locations and time periods. Since version 4, YAGO combines Schema.org's structured typing and constraints with Wikidata's rich instance data. It contains 2 billion type-consistent triples for 64 million entities, providing a consistent ontology for semantic reasoning with OWL 2 description logics. Temporal information in YAGO 4 is sourced from Wikidata qualifiers, which annotate facts with validity periods and other metadata. YAGO 4 adopts the RDF* model for representing temporal scopes, enabling precise assertions about facts within specific timeframes. This approach ensures accurate temporal modelling without implying current states [180]. YAGO can be accessed in different RDF formats, but little information is provided on its evolution or the changes in its schema.

The **LOD Cloud**,[22] is, although regularly re-published and maintained since 2007, a collection/catalogue of (interlinked) Knowledge Graphs, rather than a KG on its own. Due to its decentralised nature, anyone can submit a dataset, and the evolution of the respective constituent KGs is not observable from this source directly. While many of its catalogues KGs are accessible via dumps or even SPARQL endpoints, at the same time, many of its datasets have disappeared over time and are no longer (or irregularly available).

As for *queryability*, the LOD-a-LOT dataset,[23] which has been created as an attempt to clean and crawl all accessible datasets of the LOD cloud and make it available in HDT [83] compressed form [82] – to the best of our knowledge this remains to date a static, once-off effort. While this dataset has also been re-used in other works, for instance, to analyse cross-linkage and ontology-reuse within the LOD Cloud [104], such investigations are lacking a longitudinal analysis of development over time. Likewise, little is known about the evolution of its schema expressivity: a once-off study from 2012 on the Billion Triple Challenge sample from different LOD Cloud datasets has found for instance that hardly any OWL2 constructs had been used at the time [95], and most of the ontologies in Linked Data had used only a moderately expressive fragment of OWL, which had been called OWL LD in this study. A subsequent or even continuous assessment over time with respect to changes or uptake of OWL constructs in LOD over time is to the best of our knowledge still missing. We note that, while the evolution of the LOD Cloud schema itself was partially studied, e. g. the changes and interlinkage of the RDF vocabularies [1, 2], this study did not include expressivity as such.

Unfortunately, such longitudinal analyses over the LOD cloud's evolution as a whole are hardly reproducible or observable a posteriori, since, by its nature, availability of versions, separate schemata and change logs, as well as information about temporality and timeliness is highly heterogeneous across the LOD Cloud datasets. Only summary statistics about the individual states of available datasets at the time of updates are available; i. e. the LOD Cloud service as such does not capture the LOD's historical development itself and older versions of the data itself are typically not provided. External initiatives have attempted to address this problem:

- the Billion Triples Challenge (BTC)[24] initiative that, starting from a certain set of seeds, collected billions of triples on the LOD using the popular LDspider [118] framework. The first BTC snapshot of the LOD Cloud from 2009 contained about 1B triples. The crawls have been repeated in irregular year-based intervals. The largest version is from 2014, with about 4B triples.
- The Dynamic Linked Data Observatory (DyLDO) [140][25], initiated in 2012, partially overcomes this limitation by providing weekly snapshots of about $90,000$ URIs using the same crawler as the BTC dataset, resembling about 150 to 205 million triples per week. Key characteristics

[22] https://lod-cloud.net/
[23] http://lod-a-lot.lod.labs.vu.nl/
[24] https://www.aifb.kit.edu/web/BTC
[25] http://km.aifb.kit.edu/projects/dyldo/

of the dataset are that the weekly crawls are stored as so-called snapshots using the N-Quad format [45]. This means that the full graph data collected per week is available in a single data dump. The variance of the collected data reflects the changes in the LOD Cloud. The main drawback of this approach in evolution analysis is that the seed URLs have not changed since the start of the data collection; this initiative is apparently the longest-running collection of a subset of the LOD Cloud.

While well-known, publicly available Knowledge Graphs (KGs) such as DBpedia and Wikidata play a significant role in the realm of structured knowledge, there are other, perhaps less widely recognised, but equally substantial KGs that deal with highly dynamic data. Two notable examples are the GDELT Global Knowledge Graph[26] and Diffbot.

The **GDELT** project has been providing an integrated event stream for media news events since 2013, and it has evolved into a comprehensive event KG. It separates events and associated entities such as individuals, organisations, locations, emotions, themes, and event counts into a continuously updating KG. The GDELT 1.0 Global Knowledge Graph, initiated on April 1, 2013, consisted of two data streams – one encoding the complete KG and the other focusing on counts of predefined categories (e. g. protester numbers, casualties). GDELT 2.0's Global Knowledge Graph (GKG)[27] enhances this with additional features, incorporates 65 translated languages, and updates every 15 minutes. Notably, mappings of GDELT into RDF stream were proposed, yet it is limited to only the event graphs and the GKG [235, 236].

As for *queryability*, GDELT can be accessed via Google's BigQuery[28] in its current state [235], updated every 15 minutes in real-time with temporal information available at the event level at different granularities, with a fixed schema.

Being updated in an automated manner from news sources, this stream KG is not in the same sense *collaboratively* evolving as Wikidata or DBpedia, in the sense of individual users contributing changes by their edits, but rather from curated news sources. While, to some extent, these sources could also be interpreted as "collaborative" agents contributing to the KG on the one hand, on the other hand, the act of changes has not collaborative nature in the sense that one of these actors could overwrite or undo others' additions.

Similar to GDELT, Diffbot offers a commercially available Knowledge Graph[29] that combines dynamic event data with information about products, events, and organisations. This Knowledge Graph is only available as a commercial service, wherefore we do not discuss it here in more detail.

3.2 Monitoring Trends

The LOD cloud can be seen as a network of open interconnected KGs, the most prominent of which are Wikidata, DBpedia, DBLP and YAGO. As such, a key part of its evolution has been the open community's continuous maintenance of these KGs. Indeed, their growth has been central to the expansion of the LOD cloud from ≈ 6.7B triples and 90 RDF datasets [20], in 2009, to ≈ 28 B triples and more than 1,200 datasets [177], by 2020.

With the growth of the LOD cloud comes the desire to analyse its temporal changes and track trends and evolution. Below, we first discuss approaches to analyse at the instance-level the changes in the LOD cloud. Subsequently, we take the perspective of the schema-level and consider methods and works analysing the changes of the LOD cloud in terms of the vocabulary.

[26] https://blog.gdeltproject.org/gdelt-global-knowledge-graph/
[27] https://www.gdeltproject.org/data.html
[28] https://console.cloud.google.com/marketplace/product/the-gdelt-project/gdelt-2-events
[29] https://www.diffbot.com/products/knowledge-graph/

3.2.1 Instance-level Monitoring

Several works have sought to capture and understand the nature of KG evolution. One such seminal initiative is DyLDO (see Section 3.1), which has been monitoring Linked Data on the Web since 2012, by collecting continuous LOD snapshots and examining them in terms of their document-level and RDF-level dynamics. The original paper [139] is based on the analysis of 86,696 Linked Data documents for 29 weeks and reveals that $\approx$ 62% of the documents available during that time were, in fact, unchanged. In the remaining, the changes occurred mainly very infrequently, $\approx$ 23%, or very frequently, $\approx$ 8%, with very few documents reporting changes in between. The same polarising trend is recorded for very static domains, $\approx$ 44%, change very infrequently, $\approx$ 28%, or very frequently, $\approx$ 25%. The study also reveals that data changes occurred most frequently at the level of object literals, while schema changes (involving predicates and `rdf:type` values) were very infrequent, often related to time stamps, and very rarely involved the creation of fresh links.

Analyses of the DyLDO dataset include the work of Nishioka and Scherp [166] who applied time-series clustering over the temporal changes of the DyLDO snapshots and determined the most likely periodicities of the changes using an algorithm from Elfeky et al. [75]. This resulted in the finding of patterns in the evolution of the graph data. Although 78% of the first three considered years of DyLDO snapshots do not change at all, the remaining nodes could be organised into seven clusters of various sizes and periodicity. The latter ranges from periodicity prediction every week to once every half a year or year. Information-theoretic analyses have also been applied to analyse pairwise changes in graph snapshots of the DyLDO dataset [167]. Time-series clustering allowed us to organise the evolution into segments of similar behaviour. The study reveals that nodes of the same type show a similar evolution, even if these nodes are defined in different pay-level domains, i. e., different organisations. Finally, Gottron and Gottron analysed the same dataset but applied perplexity to explain the evolution of graph data [98].

At the level of the individual LOD cloud KGs, **Wikidata is an especially interesting example of an evolving KG**, having 90M entities and 1.4B revisions by more than 20K users.[30] The recent Wikidated 1.0 dataset [208] records the fine-grained organic evolution of Wikidata from its inception in 2012 until June 2021. The statistical characteristics of Wikidated 1.0 reveal a linear growth in the number of entities, which has been slightly accentuated after the Freebase integration in 2015. Also, almost all entities have less than 100 revisions, with half having less than 10. In terms of revision speed, the analysis highlights that most entities are edited frequently. Specifically, 60% of the revisions of a given entity occurred less than a month after a previous revision of the same entity. Inspecting the types of revisions, the paper indicates that most revisions consist of atomic changes, with approximately 90% containing less than 10 triple additions; moreover, 80% of revisions do not feature triple deletions. Another interesting trend indicates that half of the triples are added less than a day after the creation of their entity, while deletions take much longer, with over half involving triples that are deleted more than 6 months after they have been added. Although the vast majority of Wikidata triples are never deleted, $\approx$ 10% are deleted only once and less than $\approx$ 1% are deleted repeatedly after being added again. The CorHist dataset [224] is also built from Wikidata's edit histories, although with a focus on constraint violations and their corrections. The study shows that users are more likely to accept corrections for familiar constraints and certain types of constraints favour over-represented entities, highlighting the impact of biases. The evolution of Wikidata has also been studied in terms of editor engagement [207] and impact [191], as well as the quality of provenance information [188].

[30] According to `https://www.wikidata.org/wiki/Wikidata:Statistics`.

The work in [169] analyses the changes in Wikidata KG from a topological perspective. As such, it establishes that the evolution of the number of nodes and edges resembles a power law [147], similar to those commonly observed in social network graphs; based on this, it proposes classifiers that verify whether changes are correct.

Levels of Granularity. Alloatti et al. [10] propose to analyse KG evolution trends by capturing their changes across different snapshots at three levels of granularity: *atomic* focuses on operations at the resource level, *local* targets the evolution of a resource within its community, and *global* detects communities at the level of the entire graph. At the level of atomic evolutions, given a set of atomic updates performed between two snapshots, the authors distinguish between *statistical* changes, quantifiable in terms of the mean and variance with respect to a normal distribution, and so-called *noteworthy* ones, which capture snapshot features that diverge from the expected KG evolution with respect to a given threshold that is dataset-specific. An example of the former type would be quantifying the number of citations of a paper, while an exceptionally high number of new citations would illustrate the latter. Local evolution would also account for community-level features, such as graph density. As such, a publication may be noteworthy only at the level of its community, and communities themselves may be identified as noteworthy based on specific features, such as topological ones. At the global level, community detection methods can provide insights into the general behaviour of the different entities in the KG. When considering KGs as multi-community networks, various detection algorithms can be applied using custom network metrics, as reviewed in [193, 87]. When it comes to investigating KG evolution at a *global* level, studies have applied metrics transferred from different disciplines, such as databases [70], information theory [167, 98], web data crawling [68] and machine learning [168, 169].

Future Directions

Even with the large number of analyses already done in the past, there are many avenues to investigate further when it comes to monitoring, but especially analysing evolving KGs at instance-level. One such direction involves exploring the commonality of data sources across different open KGs. For example, knowledge graphs like YAGO3 and Wikidata draw extensively from various language editions of Wikipedia. Investigating the extent of shared data sources and how this commonality has evolved can provide valuable insights into the collaborative dynamics of KG development. By understanding the overlaps and changes in data sources, researchers can gain a more comprehensive understanding of how this influences evolution; for example, an investigation of link evolution and cross-references between KGs over time could deliver new insights here.

Another compelling area for analysis pertains to the role of programmatic intervention in the development of knowledge bases. Many knowledge graphs, including YAGO and DBpedia, rely on automated processes for data extraction and transformation, including, in the case of YAGO, statistical learning. Likewise, Wikidata's data generation, while predominantly carried out by its users, also relies partially on programs that extract information from external sources through bots. Delving into the balance between manual curation and automated data extraction and its impact on KG growth and quality can offer valuable insights into the mechanisms that drive their evolution.

These future directions in KG analysis provide exciting opportunities to deepen our understanding of how these structures evolve, the factors influencing their development, and their crucial role in the dissemination of structured knowledge. Addressing these challenges will contribute to the ongoing advancement of knowledge representation and dissemination in the digital age.

3.2.2 Schema-level Monitoring

All the aforementioned studies of the evolution of Web graphs focused on the instance-level of the graph *data*, i. e., the nodes modelling the entities in the domain. Only a few works also considered **analysing the evolution of the *schema-level* of the graph**. An early study by Dividino et al. [70] shows that indeed, the schema of a node changes over time when one considers how the available RDF properties and RDF types are combined to a set of edge labels and node types to model a node. We call this set of properties and types the *schematic structure* of a node. Over one year in the DyLDO dataset, the authors analysed the schema structures of the nodes in terms of both the outgoing properties as well as types. They found that in each snapshot between 20% and 90% of the schema structures change from one version to the next. This means that more or fewer nodes have the same schema structures, nodes with new schema structures are observed, and some schema structures are not used anymore. There are also some combinations of properties and types where the schema structure of the nodes is very stable, i. e. the set of nodes with that specific schema structure did not change for one year [166, 70].

Just like new data nodes appear and change in the Web graph, the vocabularies used to model such data also change, but at a much slower speed. New vocabulary terms are coined to cover additional requirements or reflect changes in the domain. Other existing terms are modified or even deprecated. Previous work analysed the amount and frequency of changes in vocabularies based on different snapshots of the Billion Triples Challenge, DyLDO and Wikidata datasets [1]. Although the evolution of vocabularies is slow [1, 140], i. e., they happen on average a few changes every year only, a change may still have a significant impact due to the large amount of distributed graph data on the Web.

Another insight is that, in the course of an evolving vocabulary, the update of new terms from released vocabulary versions varies greatly and ranges from a few days to years. It is not surprising that even deprecated terms are still used by data publishers. Moreover, it is important to analyse both the change in the vocabulary, as well as how the various terms are used in combination. This can be seen at the schema-level: one can observe changes in the node and property shapes (e. g. SHACL shapes), as well as in their prevalence. For example, a recent study [196] compared the property shapes extracted from two Wikidata snapshots (one from 2015 and one from 2021). The analysis reported that the number of RDF classes increased from 13K to 82K and the number of predicates from 4,906 to 9,017, while the number of distinct property shapes increased from 202K to more than 2M. This calls for an in-depth study of how the different elements of the vocabulary evolve, not only in isolation but also together at the schema-level.

Finally, similar to the LOD Cloud showing the dependencies of different Web graph datasets, one may also consider the Network of Linked Vocabularies (NeLO) where the nodes are the vocabularies and the edges model vocabulary reuse [2]. Vocabulary reuse is generally encouraged, as it improves the interoperability of data, but at the same time, it also introduces dependencies between vocabularies that are to be resolved when vocabulary terms in the network change, are deprecated, or deleted. The NeLO network has been analysed over a history of 17 years based on the data from the Linked Open Vocabulary (LOV) service[31] with respect to standard network metrics, such as size, density, degree and importance [2]. LOV collects the temporal information from hundreds of RDF vocabularies added to the service through a review-based process. The evolution of this schema-level graph has been analysed with respect to the impact of vocabulary term changes, term reuse and vocabulary importance [1, 2].

[31] `https://lov.linkeddata.es/dataset/lov/`

Future Directions

Exploring the schema-level dynamics of open KGs reveals several promising avenues for future research and analysis. These areas of inquiry offer valuable insights into the evolving nature of knowledge graphs and their impact on knowledge representation.

One important aspect of KG analysis pertains to understanding how schemas are structured and evolve within graphs, but also how re-use between graphs evolves. Many open KGs, including Wikidata and DBpedia, make use of RDFS and OWL to organise their ontologies. However, the specific integration of schemas into the data varies. For instance, some graphs incorporate their ontologies directly into the data, while others maintain separate ontology files. Investigating the consequences of these schema design choices on knowledge graph evolution is another possible research direction. Additionally, assessing how expressive power and intended meaning in these schemas evolve and potentially influence KG development is of strong interest.

KGs exhibit varying degrees of semantic underpinnings, ranging from basic RDFS to more complex representations like OWL. Some, like Wikidata, may have intricate intended meanings and collaboratively evolving schema constructs that go beyond OWL's expressivity, which may necessitate advanced logics for interpretation (for instance the constantly evolving set of Wikidata's property constraints). Analysing the gap between intended, implied and supported semantics in KGs and its implications for their evolution is a further promising area of investigation. Overall debates within the Semantic Web and Knowledge Graph communities, about additional complex ontology features and the evolution of ontology languages as such, may also raise questions about the role of evolving ontology expressiveness in shaping knowledge graph structures over time.

Comparing the rates of schema/ontology evolution vs instance/data evolution in different knowledge graphs in depth is another potential future direction: preliminary observations may suggest that in some cases, the evolution of ontology structures lags behind changes in the data. Such temporal misalignment raises questions about how it affects the overall coherence and semantics of knowledge graphs over time; as a concrete example, let us again name constraints in Wikidata, which partially become outdated (and even explicitly deprecated) by their actual use – which could indeed be understood as a form of semantic drift.

Comparative analyses between knowledge graphs, especially those with similar characteristics or shared data sources, can provide valuable insights into ontology evolution, schema design and knowledge representation choices. By examining similarities and differences in their evolution processes, researchers can identify best practices and challenges in crowd-sourced ontology development.

These future directions in schema-level analysis offer opportunities to gain a deeper understanding of how knowledge graphs evolve structurally and semantically. By addressing these challenges, researchers can contribute to advancing our knowledge of knowledge representation dynamics and the evolving landscape of open KGs.

4 Study the Evolution

In this section, we discuss methods for studying the evolution of KGs. First, we introduce some relevant static graphs and KG metrics, as they have been defined to inform KG quality and are sometimes used to analyse KG evolution. Second, we address measures that concern consistency and quality specifically using constraints, as opposed to the simple metrics introduced first. In the third part, we discuss measures specifically developed to capture and quantify evolution, and we finish this section with a focus on how network science approaches could be used in the future for the study of KG evolution.

4.1 Basic Graph and Knowledge Graph Metrics

This section introduces metrics designed initially to study the properties of graphs and specifically knowledge graphs, which have been used to assess ontology quality [11, 142, 91, 37, 213, 227, 205] and that has also been used to study KG evolution [250, 252, 73, 71, 172]. Table 3 summarises such metrics, which – however – do not take an evolving KG as input for their calculation as they consider only one graph at a time. We can broadly group these static metrics into two groups: graph metrics and knowledge graph metrics.

Graph metrics are applied to a graph version of the KG or adapted to work on the KG. Examples of these metrics include *average depth* [71, 73, 91, 142], *number of paths* [142], *tangledness* [11, 91, 142] and *absolute leaf cardinality* [11, 91, 142]. In the work of Alm et al. [11], Gangemi et al. [91] and Lantow et al. [142], the metrics are applied only to the isA graph, whereas Djedidi et al. [71] apply the average depth on the OWL graph, the same as Duque-Ramos et al. [73].

Knowledge Graph metrics can be distinguished from graph metrics based on the idea of taking semantics into account. However, each approach, metric or paper specifies what type of semantics (RDF, RDFS, OWL or other) are considered and if the metrics are applied to materialised KGs or not. We do not make this specification here but leave it up to the interested reader to follow the cited sources. While instance-level analyses focus on the data graph, schema-level analyses focus on the semantic information [33]. Therefore, we divide the metrics into three groups:

- **Primitive metrics** focus on a single aspect of the KG; for instance, they are used to characterise the *number of entities* of a KG [37, 142]
- **Schema metrics** focus on the schema or T-Box of the KG. Examples of such metrics include *Property Class Ratio* [250, 252, 172, 73], *Depth of Inheritance Tree* [250, 172, 73] and *Inheritance Richness* [71, 73]. For example, most of these metrics are used in the OQuaRE quality assessment by Duque-Ramos [73] to inform about varying quality (sub-)characteristics.
- **Data metrics** or A-Box metrics mostly combine an aspect of the A-Box with one from the T-Box. Examples of such metrics include *Average Population* [73] and *Instance Comprehension* [71]. Due to their simplicity, data metrics give only a partial view of KG quality and often need to be contextualised for a complete evaluation [73].

In summary, KGs have been analysed by calculating static metrics like the ones in Table 3 on linear/nonlinear series of consecutive snapshots: by combining these measures over some time, as done for instance in [73, 33, 182, 71, 172], one obtains time series data (a versioned or dynamic KG) that allows (and is currently primarily used) for calculating descriptive statistics (e. g. central tendencies, dispersion, distribution) that partially describe the KG evolution over time.

Future Directions

While static metrics can provide valuable insights at little cost, we argue that designing specific metrics and combining those with more sophisticated time-series analyses can lead to more precise monitoring of KG evolution. In particular – for any of the above-mentioned static metrics – investigating time-series trends in *metrics variations* such as seasonality or stationarity or even more complex models [214] can provide further insights about the KG evolution. We illustrate some ideas for such future metrics by the example questions listed below:

- *Trends*: How has the average degree of nodes or centrality developed in KGs such as Wikidata over the past N years? How interconnected is the KG becoming over time?
- *Seasonality*: Are there recurring periods of increased or decreased growth in the size (number of nodes or edges)? Is there any correlation with specific events?

Table 3 Overview of general graph metrics and specific Knowledge Graph metrics from the literature: metrics are only included if there are at least 3 papers (graph metrics) using and defining a measure (excluded 98 metrics). We excluded some of the graph metrics cited by the same three papers (5 metrics); for knowledge graph data metrics we also included those with 2 citations – any of these static metrics and changes would seem worthwhile to be also investigated in a longitudinal manner over time.

		Metric	Description	Used/Defined in
Graph		Absolute depth	d_a = sum over the cardinality of each path in a set of paths in graph	IsA graph [11, 91, 142, 250]
		Average depth	d_a / $\|paths\|$	IsA graph [11, 91, 142, 250], graph [37, 71], OWL schema [73, 71]
		Maximal depth	longest path	IsA graph [11, 91, 142], graph [37]
		Number of paths	$\|paths\|$	DAG [142, 250, 252]
		Tengledness	$\frac{n_G}{t}$, n_G = cardinality of G, t = cardinality of the set of nodes with more than one ingoing IsA arc in G	IsA graph [11, 91, 142]
		Degree Distribution	mean-square deviation of the degree of graph nodes	graph [37, 67, 142]
Knowledge Graph	**Primitives**	Entities	number of entities, classes and instances	graph [37, 142], IsA graph [91], OWL [213], DAG [250]
		Properties	number of unique properties or relations	OWL schema [172], OWL [227, 229, 231], DAG [252]
		Classes	$\|C\|$ = number of classes (concepts)	OWL [227, 231], DAG [250, 252]
		Instances	$\|I\|$ = number of individuals	OWL [227, 229, 231]
		Object properties	P_o = number of object properties (non-inheritancE)	Schema [142], OWL [213, 229]
	T-Box/Schema	Depth of Inheritance Tree		Tree [172], OWL [73, 205, 227], DAG [250]
		Property Class Ratio	$\frac{\|P\|}{\|C\|}$	OWL [229, 172, 73], DAG [250]
		Inheritance Richness	$\frac{\|H\|}{\|C\|}$, H = inheritance relations	OWL [71, 73, 205, 227], Schema [142]
		Attribute Richness	$\frac{\|P_d\|}{\|C\|}$, P_d = datatype properties	OWL [71, 227], Schema [142]
		Class Property Ratio	$\frac{\|C\|}{\|P\|}$	Onto [11, 91, 142]
	A-Box/Data	Average Population	$\frac{\|I\|}{\|C\|}$	OWL [73, 227], Onto [91]
		Cohesion	number of connected components	OWL [71, 227]
		Average Class Connectivity	$mean(\|(c1, p, c2\|)$ where c1 and c2 are instances of classes	OWL [205, 227]

- *Moving Averages*: How does the moving average of additions (new triples) or deletions (removed triples) over 12 months compare to the monthly new triples values? Are there evolutionary anomalies?
- *Autocorrelation*: Is there autocorrelation in the time series data of a given ratio metric (e. g. Property Class ratio, etc.) in the KG?
- *Stationarity*: Do structural changes in the KG (for instance, lengths of certain paths or other structural metrics) follow a stationary process?

So far, time series analyses with static metrics for LOD characterisation have been traditionally restricted to descriptive statistics, e. g. in [129, 182, 73]. We argue that this is an opportunity for the Semantic Web and Knowledge Graph research community to rethink more sophisticated metrics designed to precisely measure KG dynamics and change overall and in a modular fashion (e. g. instance data vs. schema dynamics, etc.). Likewise, we see a lack of tools and calculation frameworks geared specifically towards running such more complex time series analytics on evolving KGs at scale.

4.2 Consistency-Based Quality Metrics

Assessing data quality within a KG presents significant challenges that worsen if the aim extends to monitoring, ensuring, or improving such quality over time. Consistency-based quality metrics play a crucial role in assessing many dimensions of data quality, for example, measuring the integrity, coherence and general consistency of KGs [245]. Paulheim and Gangemi [176] estimated inconsistency in DBpedia by clustering conflicting statements; they limit their evaluation to a given snapshot, neglecting the evolution of these inconsistencies.

Various languages have been developed to express and represent constraints in KGs, yet not all are equally suited to "measure" consistency and quality. That is, while formal ontology languages such as OWL [101] and the respective underlying Description Logics [21] allow one to determine inconsistency of the whole KG, typically, due to their expressivity, they suffer from ambiguity between pinpointing and counting violations. Earlier work has used rule-based fragments of OWL, OWL RL to – again statically – quantify and repair inconsistencies [113].

More recent specific standards for KG constraint languages have revived the research on quantifying constraint violations. Specifically, the relatively new W3C standard SHACL [135], and similarly ShEx [195], allows validation and counting violations in a KG, w.r.t. a set of (integrity) constraints and target node/edge definitions. Yet, we only see both formal ontology languages such as OWL, e. g. [95], and these novel constraint languages being only slowly, if ever, adopted in (openly available) KGs.

In the following, we dive deeper into the measurability of quality metrics, focusing on consistency. Consistency metrics evaluate the coherence and absence of contradictions within a KG. Constraints can be used to specify rules regarding relationships between entities, ensuring that the graph remains internally consistent. Inconsistencies, such as conflicting assertions or logical contradictions, can be identified with these metrics. There is a trade-off between measuring consistency and simply measuring missing information. However, this trade-off will be explored as part of defining assessment frameworks.

As a first approach towards monitoring consistency w.r.t. constraints over time, Wikidata has leveraged constraint modelling to enhance data quality and usability. Within the Wikidata ecosystem, the Schemas project[32] uses ShEx to define schemas for modelling various Wikidata classes. Additionally, Wikidata uses its own representation model to define constraints on its properties, known as Wikidata property constraints.[33] These property constraints serve as valuable guidelines for the community of users, aiding in maintaining data integrity and the development of violations is documented over time in Wikidata's own published database reports.[34] In a recent work, Ferranti et al. [86] have attempted to formalise the respective constraints in SHACL and SPARQL, in order to enable generating such violation reports in a standardised manner, on the fly, which may be viewed as a starting point to enable monitoring constraint violation over time.

[32] https://www.wikidata.org/wiki/Wikidata:WikiProject_Schemas
[33] https://www.wikidata.org/wiki/Help:Property_constraints_portal
[34] https://www.wikidata.org/wiki/Wikidata:Database_reports/Constraint_violations/Summary

An alternative approach to quantify violations is to attach the number of violations (n_{C_i}) for each violated denial constraint (C_i) to nodes and edges in the KG. The counting can be done in a bag or set semantics by considering the duplicates in the constraint violations or not. Provenance polynomials can be built by summing the monomials given by $C_i n_{C_i}$. The obtained polynomials and corresponding degrees of quality can be leveraged during query evaluation to characterise the quality of the query results further. Although this approach has been conceived for static relational data [119, 120], the temporal aspects of inconsistency are still largely unexplored.

Despite these starting points, the question of how to measure and monitor quality in terms of consistency in a systematic manner for particular KGs over time seems to be still an open question that opens up engaging scenarios. For example, the presence of time in evolving KGs adds a dynamic perspective to constraint enforcement, facilitating ongoing improvements in the KG through data repairs, as proposed by [57]. Moreover, the analysis of constraints over time can also provide significant insights into the occurrence of semantic drift (see Section 2.3) within the schema layer of a KG. When historical constraint definitions are compared with the current state, it becomes possible to identify schema modifications, shifts in the focus of the schema layer and potential mismatches between the evolving semantics and the intended scope.

Future Directions

As outlined above, consistency is a big factor when assessing the quality of KGs. Hence, we see several potential directions of analyses in the future using constraints to learn more about knowledge evolution concerning quality. For example, before even analysing evolution, an investigation into which KGs use RDFS, SHACL and ShEX but also how expressive their ontologies are and which are entirely based on external data sources. Such questions directly tie into an investigation of quality based on consistency and constraints and how these evolve. First, measures and frameworks must be developed to support these kinds of investigations as they require handling KGs at scale. At the same time, the tradeoff between measuring quality and consistency vs. measuring missing information must be considered in greater detail before applying such approaches to any open general-purpose KGs, as these KGs operate with an open-world assumption.

The analysis directions align well with the dimensions of evolution (dynamics, timeliness and monotonicity), but each requires different approaches or solutions. Thus, we urge the community to use constraint-based metrics to analyse the consistency of the evolution of KGs, the change (trends, seasonality, etc.) of completeness, data freshness, data recency and temporal completeness. Precisely, the last three need to regard time as data rather than meta-data.

4.3 Methods for Quantifying Evolution

In this section, we want to give space to metrics specifically introduced to capture the evolution of a KG, which require pairs of (consecutive) graphs as input in the form of a versioned or dynamic KG, according to the classification presented in Figure 1. Most of the works introduced below study the changes between two (consecutive) versions of a graph, that is, two snapshots, such as [69, 177, 182, 181], making them specifically applicable to versioned KGs rather than dynamic ones. Pernisch et al. [182] propose several metrics to capture evolution on the materialisation and also provide their implementation in a Protégé plugin [183]. The evolution metrics capture the amount of change between two snapshots using simple counts of deltas between the snapshots. Pelgrin et al. [177] developed a framework to analyse various properties of versioned KGs based on changesets computed over pairwise versions of DBpedia, YAGO and Wikidata. Their framework consists of multiple evolution metrics such as growth rate and dynamicity. The authors also measure high-level changes, such as the number of entities changed between a pair of versions, using the metrics we discussed in Section 4.1, but relating them directly to the evolution. The metrics capture the changes between a pair of snapshots but do not directly reflect KG evolution over multiple snapshots, i. e. a sequence of snapshots. Instead, pairwise comparison sequences can be considered to identify trends in evolution. Lastly, Dividino et al. [69] developed a monotonic measure for KG evolution that aggregates the amount of data changes over a sequence of snapshots. This results in a function measuring the evolution of the graph by approximating the actual evolution with an aggregation of absolute infinitesimal changes. When a KG evolves, such as Wikidata, most of the additions and deletions may be valid changes reflecting the nature of the entities modelled. However, collaborative KGs can also receive erroneous changes, be it due to vandalism or carelessness. Evolution information is exploited to assess which changes in a KG are correct [169]. Based on the features for Web data caching [168], several triple features are employed on the subject, predicate and object URIs, including additional information about the age and last edit. Notably, this improvement is achieved by purely employing information about KG evolution and not requiring historical information about the editors who perform changes on the collaborative graph.

Future Directions

As is evident from the studies mentioned above, there are not many metrics specifically developed for the study of KG evolution. This, we identify as a research gap as it is necessary to introduce measures capturing different dimensions and aspects of KG evolution. Following the examples above, measures need to capture the different aspects of evolution while at the same time being outlier-resistant. Approaches from time series analysis can be fruitful to kick-start this future direction and enable the further development of methods and metrics to study KG evolution. In the future, it is important to move from snapshot analysis to more continuous approaches capturing fine-grained evolution at the time of individual edits. We can also potentially borrow approaches from network science, as they also analyse the evolution of networks, even though the networks have a simpler representation than KGs.

4.4 Metrics and Methods from Network Science

Network science has developed tools to map and analyse complex systems, suggesting the possibility of adopting them to study the structural properties of KGs. Researchers have discovered that regularities in domains such as transportation systems, scientific communities, economic sectors, or communication systems can be fruitfully represented and studied as networks. Indeed, there are remarkable regularities in such domains that play an important role in how these systems function and evolve. For example, networks tend to have very heterogeneous degree distributions, which means there are “hub” nodes with orders of magnitude more connectivity than the typical node [7]. Social networks tend to have many triangles, as suggested by the saying that a friend of a friend is likely to be a friend. Scientific community networks often have modular structures [87], reflecting coherent subcommunities of nodes in a larger system. Empirical networks tend to be sparse (i. e. given a network on n nodes, there are far fewer than the possible $n(n-1)/2$ edges). But they also have short paths connecting all pairs of nodes (i. e., low diameters) [247].

Although recent work on multiplex or multi-layer networks considers data with multiple kinds of objects or links between them, most networks studied are generally simpler than those observed in the Semantic Web community. For instance, ordinary networks usually consist of a homogeneous set of nodes (i. e. airports) and relationships between them (i. e. direct flights between airports). Multilayer networks consist of the same nodes and different kinds of relationships they might have. For example, people who may communicate via email and telephone. Studies using this kind of multi-layer data tend rather to just generalise the methods applied to ordinary networks described in this section than to invent new ones [26, 25]. On the other hand, knowledge graphs are multi-dimensional by design. Although undoubtedly useful, such complexity presents an obstacle to studying their evolution using methods from network science. Therefore, to apply these methods to study the evolution of KGs, we must first simplify the data. However, any simplification must be driven by a substantive question to make it meaningful, and it must be significant in the sense that it discards a significant amount of data, to be tractable.

Once a simple network has been constructed, the temporal dimension of the data can be integrated by slicing data into time periods (for instance, as in [143]). Measures of the network, for instance, its diameter, the mean and variance of its degree distribution, the modularity of a community detection exercise, or the prevalence of clustering can be calculated for each slice and then plotted over time. However, the choice of the width of the time slice can have major implications for subsequent analyses [211].

The stylized facts about networks described above have important implications for things that happen to them or to them. They predict the robustness of a network, i. e. how well it holds when its nodes are removed. They predict how quickly things like information or diseases spread. Network structure plays an important role in its navigability: if you do not have a map of the network, can you still find your way from a node you know to another specific node in a reasonably short amount of steps [134, 215]? Network scientists are naturally interested in how changes in a network are captured by these measures and, in turn, how they influence things that happen within networks [165].

Network scientists have two broad solutions for the comparability issue between networks of different sizes. The first is to propose a generative model that captures many of the key properties of the network in question [38, 39], and to instantiate random graphs from this model. Next, one calculates the same statistics on this randomised version of the graph and uses it as a kind of benchmark or normalisation factor. The most simple generative model is the Erdős-Rényi model, in which edges are randomly added between nodes with a fixed probability p. Given two empirical networks of different sizes, one can create corresponding random networks with the same number of nodes and edges for each. Calculating the clustering on these random networks allows us to scale or normalise the clustering observed in the corresponding empirical networks, which then become more comparable. More sophisticated models like the Barabasi-Albert model [7] (which generates networks with heterogeneous degree distributions, i. e. hubs) and the Watts-Strogatz model [247] (which generates "small world networks" that have both short paths and high clustering) can also be used in this way, depending on the research question.

The second way to make network measures comparable between networks of different sizes and over time is to create randomised versions of empirical networks, sometimes called *null models* [128, 206]. Such randomisation typically takes place among the edges, which are randomly rewired or shuffled subject to constraints depending on context. For example, a randomization of links between Wikipedia editors and the articles they touch creates a "random" version of Wikipedia preserving editor activity counts and article edit counts. Such randomisations are similar to statistical Monte Carlo simulations and can be computationally intensive, but the resulting randomised versions of the empirical graph can provide a useful benchmark to compare against the original graph. Although these methods require both a drastic simplification of the data contained in KGs and the deployment of complicated methods such as generative models or null models, they present a significant opportunity to create more robust estimates of the dynamics of KGs. Given the degree of simplification this process requires, a clear research question about the structure and dynamics of KGs is an essential first step.

Future directions

We see the potential of using network science to investigate the collaborative nature of many open general-purpose knowledge graphs. Not only does knowledge evolve, but the way it evolves is intertwined with the editing network, for which network science and its approaches to analysing its changes over time would be beneficial. For example, if one wanted to study whether Wikidata editors were becoming more or less collaborative over time, how could one define a reasonable notion of collaborative behaviour? Could one define collaboration between two editors as a function of their using the same properties or working on the same entities? Should a pair of editors both using the most widely used property be as thickly connected as two editors using a more rarely used property? Network science offers tools to carry out such an analysis, but the researcher must make choices in pursuit of a question. Question-driven modelling of KGs as simplified "networks" can move us beyond a descriptive analysis of KG evolution.

5 Manage the Evolution

Although dynamic/versioned and temporal KGs can be considered as two alternative approaches, they introduce different challenges in their management. In the case of temporal KGs, the main challenges lie in how the temporal information is captured and represented. We discuss different approaches in Section 5.1. Although, when time is not part of the data, the KGs do not require specific data models. The temporal information lies in the updating process itself; they often publish complementary changelog streams that may or may not be represented in RDF. However, time as metadata raises a different set of challenges for KGs, including the representation of the evolution and storage options, discussed in Sections 5.1, 5.2, respectively.

5.1 Data Models for Temporal Knowledge Graphs

The two main approaches for implementing KGs are RDF and labelled property graphs (LPG). In the rest of this section, we describe how researchers and practitioners modelled temporal KGs in these two approaches. In the last part, we elaborate on open challenges with regard to capturing and then analysing the evolution of knowledge in Temporal KGs.

Temporality in RDF

The problem of how to model time-related information has been intensively studied. Amongst the multitude of proposed solutions, a broad distinction can be made by representing time in the data vs. in the metadata.

In the former case, entities can be part of statements together with their temporal properties. The Time Ontology and the Sensor, Observation, Sample, and Actuator (SOSA) ontology implements this idea, e.g. an observation can have a relation `sosa:phenomenonTime` with a `time:TemporalEntity` individual.

In the latter case, the temporal annotation applies to RDF statements (or graphs). A common method to implement it is reification, which involves annotating triples. In [109], various reification schemes were examined:

- *Standard Reification* uses a resource to represent a statement, such that it can be used in other RDF statements to add annotations (including temporal ones).
- *N-ary Relations* represent relationships using resources, stating subject involvement, value, and qualifiers. Instead of stating that a subject has a given value, it states that the subject is involved in a relationship that has a value and qualifiers.

- The *Singleton Properties* approach involves creating a property that is only used for a single statement. The resource representing the statement is annotated with this property to add more information.
- RDF 1.1 introduced the notion of *Named Graphs*, which can, for example, be serialised in N-Quads. One can annotate the named graphs, e.g. associating the same temporal annotation to all the statements contained in the graph.
- *RDF-star* [107] extends RDF through embedded triples, i.e., an RDF statement can be the subject or object in another RDF statement. Just as standard RDF can be queried via the SPARQL query language, RDF-star can be queried using SPARQL-star (formerly SPARQL*), allowing users to query both standard and nested triples.

There is no single way to represent contextual information in RDF graphs, and the different mechanisms have advantages and disadvantages. Reification and n-ary relationships model complex facts in RDF. However, adding reification triples for each reified triple increases the data volume, making metadata queries cumbersome due to the need for additional subexpressions to match the corresponding reification triples. Other methods, such as singleton properties and named graphs, reduce the number of extra triples. However, these approaches require verbose constructs in queries, introducing artefacts to associate triples with their metadata [171]. RDF-star is more compact and adds facilities to the query language via SPARQL-star but does not achieve the levels of flexibility as some previous approaches. Of the strategies presented, named graphs are the most flexible since they allow assigning one annotation to sets of statements; RDF-star is the least flexible option since it cannot capture different sets of contextual values on an edge [112].

Temporality in Labelled Property Graphs

Labelled Property Graphs (LPGs) are another popular solution to represent KGs. The problem of the representation of evolution, particularly temporality, has also been addressed in their context. However, while in principle LPGs allow direct attachment of temporal information to edges in the graph, there is no consensus on a single approach for temporal LPGs. Similar to core RDF approaches, works in LPG in the literature differ in supported time dimensions (valid time, transaction time or both/bitemporal), types of possible changes to graph structure and properties, and representation as either a series of graph snapshots or a single graph reflecting changes over time.

The *Temporal Property Graph Model* (TPGM) [201] extends the *Extended Property Graph Model* (EPGM) to support analytical operators on directed graphs that evolve in Gradoop. TPGM adds support for two different time dimensions, valid and transaction time, to differentiate between the evolution of the graph data concerning the application and managing the data. This approach offers a flexible representation of temporal graphs with bitemporal time semantics. TPGM expands EPGM with four new time attributes as mandatory for vertices, edges, and logical graphs: two for transaction time intervals and two for valid time intervals.

Debrouvier et al. [60] apply temporal database concepts to graph databases to model, store, and query temporal graphs for historical data tracking. The focus is on the *Interval-labelled Property Graphs* data model, which timestamps nodes, relationships, and node properties with temporal validity intervals, allowing for heterogeneous graphs with different types of relationships. This model enables richer queries and supports two path semantics: *Continuous Path Semantics* and *Consecutive Path Semantics*.

Andriamampianina et al. [12] propose a conceptual model to represent temporal property graphs and define a set of operators to perform queries on these. The model establishes various concepts to represent objects, their relationships, and their evolution over time. It manages time through valid time intervals to track changes and occurrences in the real world. To describe an object, the model introduces the notion of *temporal entity*, comprising a set of states to represent different versions of the entity over time. Each state includes attributes, attribute values, and a valid time interval. A temporal relationship, analogous to a temporal entity, describes the link between two entity states.

Future Directions

Despite RDF and LPGs originating in different contexts, the two approaches are valid for creating and representing KGs. Several graph database vendors support both approaches to offer their customers flexibility and choice. In this context, an ongoing research direction lies in the *interoperability* between the approaches. Despite the active research [4, 15, 144], to the best of our knowledge, there is no study on the RDF-LGP interoperability in the context of temporal KGs. The challenge lies in the way the time can be represented in both RDF and LPGs: the multitude of different approaches leads to many possible conversion procedures. We argue that reference models are needed to unify the existing approaches and to set the basis for standardisation initiatives that will ease the creation, storage and processing of temporal knowledge graphs in different engines.

Another direction relates to query languages for temporal KGs. SPARQL and the LGP query languages consider temporal annotations as any other type of annotations. As such, query writers need to understand how time is represented in the graph and write the query accordingly. However, temporal annotations enable specific time-related operations, such as creating selection criteria based on Allen's relations [9]. Encoding such relations in the queries is not trivial and often error-prone. Treating time as a first-class citizen in the data models can lead to query languages with specific time-related operators, simplifying the query writing process and constructing dedicated query engines that can efficiently evaluate such operators. While this idea has been investigated in the context of continuous query processing over RDF streams (see section 6.2), it has not yet been deeply investigated for temporal knowledge graphs.

Interoperability between the two models would also further enable the possible application of analysis frameworks, existing and future ones. The same can also be said about SPARQL integrations, as in the past analyses have made use of SPARQL. Therefore, a SPARQL extension for temporality (of any dimensions) would further support efforts into KG evolution analysis.

5.2 Storage Methods

Since in temporal KGs the time dimension is managed as part of the data, temporal information integrates naturally in the data model and can therefore be captured using standard methods as outlined in Section 5.1. In the case of dynamic and versioned KGs (Figure 1), alternative approaches have been proposed capturing temporal information outside the data model itself.

An intuitive way of storing versioned KGs is to store each complete version of the KG as a new copy, often referred to as the *Independent Copies* approach [81]. While this can even be implemented using standard triple stores with named graphs, it has scalability issues regarding the number of named graphs (one for each version) and the required storage space for larger KGs. An advantage of this approach is that all queries to be executed on a single full version of a KG can be executed very efficiently since no additional computation (see below) is needed to retrieve the complete version of a graph to execute the query on. IC approaches are generally very useful for small knowledge graphs [177].

To reduce the storage overhead, *Change-Based* approaches store several full versions of the KG as snapshots but only sets of changes (deltas) for the versions in between. This makes them a hybrid solution between versioned and dynamic KGs. In this setup, querying versions that correspond to snapshots is again very efficient since the full KG is readily available. The disadvantage of this approach is that for the versions between snapshots, chains of deltas have to be applied on the preceding snapshot to recreate full intermediate versions [222, 19, 5, 179, 178]. An important aspect is then to identify which versions to materialise as snapshots and which ones to capture as deltas.

Instead of capturing entire versions of complete KGs, dynamic KGs annotate individual triples with timestamps, so-called *Timestamp-Based* approaches. In such a setting, it is then of course expensive to recreate particular versions of a KG since this requires filtering all triples based on their temporal validity. On the other hand, it becomes efficient to look up the temporal validity of each triple.

Future Directions

While most systems implement only one of the above-mentioned storage methods [223], there are hybrid approaches that can be configured to resemble one or the other. In this sense, one direction of future work is to investigate how to exploit the strengths of different storage techniques for certain use cases and develop adaptive approaches that choose and adjust the storage layout based on how the data is used.

Building upon existing approaches for the above-mentioned storage models, one of the main challenges is scalability. On the one hand, we need to develop more efficient storage methods to reduce the storage overhead of capturing information about versions and temporal validity. On the other hand – and this is very much determined and influenced by how the data is stored – future work needs to develop efficient methods for querying that can not only retrieve complete versions of a KG but also allow efficient query processing over certain versions of a graph (see also Section 6.1).

Finally, it is worth noting that the way the data is stored affects the type of possible analyses on KG evolution. For example, if one wants to run time-series analyses (as described in Section 4.1), change-based approaches are ideal due to their focus on changes. Independent copies may not contain enough fine-grained information to perform such analysis. However, metrics based on consistency metrics (as described in Section 4.2) may not work in change-based approaches as some intermediate changes may affect the consistency of the KG. Therefore, we envision storage solutions able to store KGs following different approaches, with the ability to perform a wide range of analytics tasks on KG evolution in efficient ways.

5.3 Mapping Schemas

Supporting KG versions is a key approach to ensure the stability of downstream applications for KGs. Therefore, it is essential to capture the evolution on the schema-level by sets of schema changes that typically occur in collaborative and decentralised processes.

Schema evolution requirements have been discussed in the past, in particular with respect to ontology evolution [28]. The availability of expressive and declarative mappings specifying the evolution between an original version of a schema S and an evolved version S' makes it possible to cater for the automatic propagation of the changes on the corresponding instances.

There exist two inherent problems with mappings between schemas. The first problem corresponds to the (semi-)automatic computation of the schema mappings by leveraging schema matchings and Diff(erence) computation [197]. Schema matchings can be defined as one-to-one correspondences between two different versions of a schema, and they can be coupled with a confidence value. On the other hand, schema mappings are declarative specifications, typically expressed in a subset of First-Order logic, representing the transformation between two different versions of the underlying data. Schema mappings are typically expressed as source-to-target tuple generating dependencies (s-t tgds), whose left-hand side is a conjunctive query and right-hand side is a conjunctive query enhanced with existential variables, which lead to value creation. In the case of schema evolution, schema mappings are adapted after schema changes and meta modelling abstractions serve the need of providing high-level programming interfaces than other techniques [31].

The second problem concerning mappings between schemas is the so-called schema mapping or data exchange problem [78, 28], consisting of computing the transformed target instance (also called target solution) by applying the source-to-target tuple-generating dependencies between source and target schemas. In the case of schema evolution, the target schema might undergo some changes, thus entailing the propagation of these changes to both the mappings (s-t tgds) and the corresponding target solution.

The most expressive schemas for KGs are ontologies, which allow conceptualising a domain. They provide a steerable vocabulary for a given domain of interest, defining the ontology concepts as well as the properties and relationship between these concepts. Several research approaches study collaborative ontology evolution and ontology matching, as surveyed in [108, 77]. Without going into the details of these approaches, we point out that in the last decade after the above approaches, schemas for graphs have profoundly evolved, thus bringing more open challenges for KG mappings and transformations.

Finally, often KGs originate from external databases that can contain graph data in different formats or even other data models, such as relational or document databases. There are approaches, such as R2RML [55], to facilitate the latter, but the mappings from relational data to RDF have to be (manually) adapted whenever the native (or the integrated) schema changes.

Future Directions

Recent schemas for KGs range from RDFS [42], SHACL [135], and ShEX [22] to PG-Schema [14] and their evolution, as well as the mapping problems related to computation of schema mappings and computation of the target solution, are not yet studied. The first three schemas are applicable to mapping RDF data, while the latter is applicable to mapping property graphs [38]. One relevant future direction consists of studying the automatic generation of schema mapping transformations and the data exchange problem for the above models in a time-varying context thus exploring schema evolution and versioning for evolving knowledge graphs under recent schema languages.

Another important direction concerns the mappings from RDF to property graphs or the other way round [132, 15] in order to pay attention to producing incremental or comparable schemas in comparison to previous versions. Especially complex constructs have alternative translations into the other model. Hence, small changes can have big structural impacts on the integrated result. It therefore remains mostly unclear how to appropriately capture and measure schema evolution caused by schema changes in the input data. Although some proposals, such as the OneGraph vision [144], propose to achieve graph interoperability by allowing users to use Cypher or SPARQL independently from whether RDF or property graphs were chosen as the data model, this only means that users are free to choose the query language that they prefer or that is more appropriate for a different use case; the underlying challenges of how to capture evolution in the underlying graph model remain the same.

6 Spread the Evolution

Typical tasks to process KGs include querying, reasoning, and machine learning. When we move from static to evolving KGs, one should consider the temporal dimension. In Sections 6.1 and 6.2, we discuss two classical operations on knowledge graphs: querying and reasoning. Next, we discuss learning techniques in Section 6.3. We conclude by discussing evolving KG exploration in Section 6.4.

6.1 Query Processing

We introduced data models for temporal KGs in Section 5.1. As the temporal information can be modelled in standard RDF (e. g. through named graphs or reification), in RDF-star and LPG, it follows that their relative query languages, such as SPARQL (or SPARQL-star), can be used to retrieve data from them. However, as we explain in Section 6.1, several researchers proposed ad-hoc query languages where time is a first-class citizen. Next, we discuss querying for versioned KGs in Section 6.1, focusing on the solutions to extract and query a specific KG version. Finally, we introduce continuous queries in Section 6.1 to monitor changes and to evaluate a query on evolving data continuously.

Temporal Querying

Temporal queries refer to languages and operators that offer native support for retrieving and manipulating time-referenced data. The semantics of a temporal query language are usually closely coupled to a temporal data model that defines the underlying data abstractions (see Section 5.1).

Despite the growing popularity of temporal data in KGs, this research area is still in its infancy. Exciting proposals (with a few exceptions) represent the graphs using either RDF or LPG and approaching change as a snapshot sequence. Thus, their query-answering capabilities are limited to those possible under the snapshot reproducibility principles, i. e. answering a temporal query over a database is equivalent to taking the union of all the answers obtained by evaluating the non-temporal variants of the query for each database state [35]. For example, τ-SPARQL [226], SPARQL^T[251] propose syntactic extension meant to access RDF triples annotated with a timestamp. Zhang et al. [253] went one step further with their proposal, SPARQL[t], extending the annotation with an interval-based validity time. Raising the expressivity bar, Arenas et al. [18] studied Temporal Regular Path Queries (TRPQ) to interrogate reachability over time over property graphs extended with time intervals of validity. Intervals of validity represent consecutive time points during which no change occurred for a node or an edge in terms of its existence or property values. Their approach, similar to T-GQL [61] and the Temporal Graph Algebra [161], is designed for Labelled Property Graphs. The main drawback of such a query model is the lack of support for operations that explicitly reference temporal information [18]. Therefore, an extension of this query model that propagates temporal information across snapshots has been proposed [66].

Querying Versions

Querying archives is not straightforward; since there is no well-defined or commonly accepted standard, archiving engines typically propose customised solutions for querying their data. AnQL [256] and SPARQL-T [92], for instance, are SPARQL extensions based on quad patterns – where the fourth component indicates the version over which the given query should be executed. T-SPARQL [100] instead is a SPARQL extension where groups of triple patterns are annotated with constraints regarding temporal validity supporting time ranges and timestamps. Other extensions go beyond the temporal dimension and include geospatial constraints [30, 185]. Some archiving engines [178, 179] also use the GRAPH clause of SPARQL to denote specific versions.

Apart from different approaches on how to formulate queries syntactically, one can distinguish different types of queries over archives based on the way they access the available versions of the knowledge graph [81, 177]. Two basic retrieval tasks are to extract a specific full version of a KG from storage (*Version Materialisation*) and to extract deltas (changesets) between pairs of versions (*Delta Materialisation*). In addition, we can distinguish different types of queries; the most commonly supported type of queries on evolving KGs are those where a SPARQL query is to be evaluated over a specified full version of the KG (*Single Version*). Another type of query aims at comparing answers to full SPARQL queries on different versions of a KG (*Cross Version*, e.g. which of the current countries was not in the original list of UN members. Instead of retrieving the answers to a SPARQL query, one can also aim to retrieve the specific versions in which a given SPARQL query yields (specific) results (*Version*), e.g. in which revisions did the USA and Cuba have a diplomatic relationship?

While the literature also introduces queries on deltas (single delta and cross delta), where queries can be evaluated on the changesets only, we argue that these types of queries can be considered subsumed by the above-mentioned types on full versions of a KG and assume that the archiving engine will detect during query optimization whether a complete version of the KG needs to be retrieved of a retrieving a changeset is sufficient.

Continuous Querying

Continuous queries (CQs), also known as standing queries, differ from other query processing tasks due to their never-ending nature. Indeed, they are typically used to analyse evolving data, including evolving KGs, to identify patterns, trends and outliers. With respect to the running example, one may write a query to monitor the movements of artworks between galleries. While the artwork is displayed in New York, the continuous query returns New York when specifically queried for the "current location". When the artwork is moved to Madrid and consequently the KG is updated, the query's result changes to Madrid as soon as the information changes.

The most relevant trait of CQs is the time-varying nature of the answers. Indeed, a query evaluated under continuous semantics produces a series of responses as if it was evaluated for every time instant. In practice, continuous-query evaluation is either periodic or based on custom conditions, e.g. the occurrence of an event or data change. Although several proposals exist for relational data [237], their potential in the Knowledge Graph world remains substantially unexpressed.

The Semantic Web literature has explored continuous queries for Streaming Linked Data [41] proposing several SPARQL extensions, e.g. C-SPARQL, CQELS, SPARQL$_{stream}$, including some able to combine different modalities [184]. Such languages have been reconciled by Dell'Aglio et al. [64], who explained their continuous query semantics using three families of operators adapted to RDF from [17]. RSP-QL describes how, despite syntactical differences, the existing languages all use window operators to cope with the infinite nature of the input data, usually modelled as a partially ordered sequence of timestamped RDF graphs. On a parallel line of research, EP-SPARQL [16], DOTR [155], and OBEP [233] have explored the approach for detecting event patterns in RDF streams. Such languages leverage time-aware operators and can be evaluated using regular expressions. Although the SPARQL query is entirely supported semantically, such proposals have given little attention to subgraph matching and navigational/exploratory continuous queries. Notably, queries involving (regular) path expressions that cover more than 99% of all recursive queries found in massive Wikidata query logs [40].

Regarding navigational continuous queries, Pacaci et al. [174, 175] modelled the graph as an ever-growing sequence of timestamped edges. Moreover, they studied two query models, Regular Path Queries (RQP) and Union of Conjunctive RPQs. Such query models are analysed with and without explicit deletions as a form of the materialised view.

Finally, continuous subgraph-matching (CSM) is a particular case of the foundational subgraph-matching problem, where the target graph is subject to updating (either append-only or with explicit deletions). Sun et al. [221] recently surveyed the existing exact approaches, modelling the CSM problem as incremental view maintenance.

Future Directions

Besides an investigation of which approaches have been applied to which general-purpose open KGs and how they perform, we distinguish two main directions for what concerns querying evolving knowledge graphs, i. e. addressing the open challenges related to each query model and a more general challenge that goes in the direction of a unified query model.

Temporal Querying for EKG has built upon the adoption of a single temporal model and snapshot reducibility. Future work requires relaxing such assumptions. The simultaneous application of multiple temporal models relates to the heterogeneous nature of graph data. Indeed, KGs are often referred to as a way to address data variety and perform data integration. However, such variety is not allowed within the temporal model, given an entailed complexity exposition. Going beyond the snapshot reducibility means allowing explicit temporal reference within the query settings. Such an approach reduces the temporal-navigational mismatch in the query language, allowing for posing complex questions over hybrid graph data models.

As explained above, querying versions of a KG often entails evaluating queries on a specific version of a KG or multiple ones. Naturally, the storage layout and available indexes determine how efficiently a query can be answered. Hence, developing appropriate indexing, storage layout, and efficient query optimisation techniques exploiting them are important aspects of future work.

The challenge related to continuous queries over EKGs relates to the central role of windowing in Streaming Linked Data, which poses serious limitations to the adoption and the optimisation of continuous queries. Users must know the temporal context of the interested phenomenon to choose an appropriate windowing policy. Moreover, aggregation-optimised windowing, which is well-known for relational data, was not studied for graphs. On the other hand, navigational continuous queries, and in general continuous subgraph matching, were little studied. Their relationship with knowledge evolution is noticeable and further investigation is required.

Finally, searching for a unifying query model that could make the best of the existing one is open and motivated by the specific need to migrate from one model to another when necessary. Currently, the users must pick one data and query model, and thus, their query ability is limited by the design choice of such languages. Instead, a formally verified language for EKG data that can express queries about time, through time, and in time is still missing.

6.2 Reasoning

Reasoning over large KGs layered with an OWL ontology to describe their schema may be prohibitive when using the full power of OWL. However, reasoning within the OWL 2 profiles [137] brings very interesting computational properties. Indeed, state-of-the-art reasoners over KGs typically focus on fragments of OWL (e. g. [164, 46, 238, 29]). For example, OWL 2 RL axioms can directly be translated into Datalog rules [162] enabling the use of efficient Datalog engines (e. g. [164]) that will expand the KG with implicit facts following from the OWL ontology and the KG data. Reasoning also enables the use of the notion of logical difference [136], which can be essential to understanding the evolution of a KG in terms of new entailed facts. For example, `diff(KG, KG')` represents the (entailed) facts in `KG'` not present in `KG`.

Reasoning in Evolving Knowledge Graphs

Rule-based systems typically perform materialisation (i. e. precomputation of the consequences after reasoning) before queries over the KG are evaluated. Changes in the KG require recomputing the materialisation so that query results are up-to-date concerning the changes. This process may be expensive for very large KGs and rule sets, especially if they constantly evolve. Most systems adopt Incremental materialisation when changes are to be reflected as soon as they occur (e. g. [241, 163]). These systems focus only on the part of the KG affected by the changes and implement optimised solutions to perform efficient incremental reasoning. In addition, there have been efforts in the literature to enhance incremental reasoning via modular materialisation (e. g. [114]) and enable distributed materialisation via data partitioning (e. g. [6]).

The evolution of a KG may also require the integration with other KGs as described in Section 5.3. The compatibility of integrating multiple KGs has been extensively evaluated from the ontology alignment perspective. In the literature, several approaches aim at identifying logical errors and unintended logical consequences derived from the alignment of the KGs (e. g. [157, 79, 216]). To the best of our knowledge, at the moment, no studies are focusing on how KG evolution affects consistency in alignment tasks. We believe that this is an important future direction because the effect of changes on reasoning can be substantial [182] and can also unexpectedly impact alignment tasks [183].

Reasoning for Studying Evolution

Logic-based reasoning, as discussed in previous sections, can play a key role in conducting constraint validation and can contribute to the definition of robust metrics to measure KG evolution. For example, the *semantic drift* described in Section 2.3 can be tackled via the logical difference [136] between two versions of a KG after materialisation. $\mathtt{diff(KG}_i\mathtt{,\ KG}_{i+1}\mathtt{)}$ represents the new (materialised) facts in $\mathtt{KG}_{i+1}$ not present in $\mathtt{KG}_i$, while $\mathtt{diff(KG}_{i+1}\mathtt{,\ KG}_i\mathtt{)}$ represents the facts that were lost in the new version of the KG $\mathtt{KG}_{i+1}$. An analysis of the impact of changes on the materialisation in the case of $\mathcal{EL}^{++}$ ontologies in the biomedical domain was analysed in previous work [182], where the authors quantified the change in the materialisation to learn how ontologies evolve over time.

Efficient rule-based reasoning can also be leveraged to evaluate the evolution of the knowledge graph in terms of the conformance of the data with respect to the ontology and available constraints. This conformance evaluation can complement the related *quality metrics* (see Section 4.2). For example, Kharlamov et al. [131] interpreted some OWL 2 axioms involving cardinalities and ranges as integrity constraints and represented them as Datalog rules to identify violations of those constraints. For example, the following OWL axiom (α):

$$\text{MasterPiece SUBCLASSOF (carried\_out\_by SOME Artist)} \tag{1}$$

is transformed into the following Datalog rules:

$$\text{Art\_pieces\_carried\_out\_by\_artists}(?\text{x}) \leftarrow \text{carried\_out\_by}(?\text{x}, ?\text{p}) \ \wedge \ \text{Artist}(?\text{p}) \tag{2}$$

$$\text{Violation}(?\text{p}, \alpha) \leftarrow \text{MasterPiece}(?\text{p}) \ \wedge \ \textbf{not} \ \text{Art\_piece\_carried\_out\_by\_artists}(?\text{p}) \tag{3}$$

In the example above, it is expected that *MasterPiece* in the KG have at least an explicitly associated *Artist*.

Stream Reasoning

When the KGs evolve at a high pace, and the information needs to focus on extracting novel and recent information, we enter the realm of stream reasoning [62]. Stream reasoning combines knowledge representation with stream processing techniques [52] to process evolving ontologies and KGs in a continuous and responsive fashion[64]. Stream reasoning cases relate to Timeliness (Section 2.1), i. e. the inference is needed before data are no longer useful.

Firstly, several research groups worked on defining data models and vocabularies to capture data streams through KGs and ontologies. Zhang and Stuckenschmidt [115] introduce the notion of linear version space to define a sequence of ontologies. Such a notion was later adapted by Ren and Pan [199] to define ontology streams as a sequence of timestamped ontologies. An alternative model for data streams is RDF streams, defined as a sequence of timestamped statements (as in [24, 186]) or graphs (as in [63]).

Reasoning task extensions over streams, such as consistency check and closure, were first studied with a focus on adapting reasoning algorithms to the streaming settings. For example, Barbieri et al. [24] extend the incremental reasoning algorithms DReD for stream reasoning with sliding windows. The authors exploit the knowledge derived from the sliding window operator to calculate when assertions must be deleted and use such information to improve the performance of the materialisation algorithm. Ren and Pan [199] propose a truth maintenance system implemented in the TrOWL reasoner that builds a graph to track the derivations. When the assertion changes, the system incrementally maintains the graph and consequently updates the materialisation.

Over time, the focus moved to the application of temporal logic for stream reasoning: here, Beck et al. proposed the Logic-based framework for Analysing Reasoning over Streams (LARS) [27]. LARS combines temporal logic operators with specific operators to reason over streams, such as the window operator. Tiger and Heintz [230] propose P-MTL, an extension of the Metric Temporal Logic with probabilities to model the state uncertainty. P-MTL allows the use of probabilities in the logic formulas and to use them in the inference process. One of the most recent studies is from Walega et al. [243], who researched DatalogMTL in the context of stream reasoning. They study the conditions to guarantee that no infinite materialisation occurs and show that reasoning over the fragment of DatalogMTL that satisfies such conditions is not more complicated than reasoning over Datalog, i. e. ExpTime-complete for combined complexity.

Lastly, several researchers and practitioners studied stream reasoning applications. One area where stream reasoning found considerable interest is smart cities and traffic management. Lecue et al. propose STAR-CITY [152], a system to analyse streaming heterogeneous data by combining ontological reasoning, rule-based reasoning, and machine learning. Eiter et al. [74] designed a stream reasoning solution based on Answer Set Programming (ASP) to optimise traffic control systems. Le Phuoc, Eiter, and Le-Tuan [187] use stream reasoning to integrate streams of images from car cameras and data streams to reason over them.

Stream reasoning has also found application in other domains. For example, Barbieri et al. [23] applied stream reasoning techniques to social media streams for personalised recommendations; Kharlamov et al. [130] propose stream reasoning in the context of monitoring failures in an industrial setting; De Leng and Heintz [59] integrated stream reasoning techniques in the Robot Operating System (ROS) to reason on the input IoT data and determine the most appropriate configuration. A recent survey discusses the maturity level of knowledge representation and reasoning within the lifecycle of existing stream reasoning applications [41]

Future Directions

Much attention is still required concerning logical reasoning to analyse and spread the evolution in state-of-the-art open knowledge graphs. As discussed, performing reasoning may be prohibitive in modern knowledge graphs if the full expressiveness of the underlying ontology is used. State-of-the-art solutions focus on tractable fragments (e. g. OWL 2 profiles) to scale with large knowledge graphs and ontologies; however, coping with these KGs still poses essential challenges in terms of scale completeness and errors in the data. To assess how far the current approaches can take us, a comprehensive analysis of reasoning methods with a combination of general-purpose open KGs is necessary to understand the limitations in real-world settings. The combination of deductive and inductive techniques [65], as discussed in Section 6.3, is key to tackling these challenges as it leads to data and knowledge-driven techniques to, e. g., complement the evolving knowledge graph and to identify and correct potentially wrong new facts [48].

Stream reasoning is a candidate to have a central role in making sense of evolving knowledge graphs. In particular, expressive stream reasoners like Laser and LARS are candidates as formalisms to capture the complex interrelations between dynamic, versioned, and temporal KGs (cf Section 1). Similarly, it needs to be verified if existing languages like RSP-QL [63] are adequate for defining transformation across EKG types. Moreover, as we envision a more prominent role for events [99], agent-based reasoning methods are an important direction towards efficient methods to spread and handle the evolution [234]. Finally, from an application/engineering standpoint, different reasoning tasks may benefit from alternative KG encoding. Therefore, solutions like RSP4J [232], ChImp [183], or the SR PlayGround [210] need to evolve to welcome EKGs as first-class citizens.

6.3 Learning

In machine learning, KGs or ontologies are often transformed into vector space known as embeddings before use. KG embeddings are low-dimensional vector representations of entities and relationships within a KG. Typical tasks over such embeddings are link prediction, KG completion, node classification, query answering and data integration. Overall, we can distinguish two main families of graph embedding approaches: transductive and inductive. In transductive approaches, all nodes and relations are seen during training while new edges among seen nodes can be predicted at inference time. Inductive approaches instead allow to train on one version of the graph and then perform inference even with new nodes and edges introduced at testing time [8]. Therefore, when dealing with evolving KGs, we can distinguish between approaches that try to adapt transductive embedding methods to the case of dynamic or evolving graphs [43, 228, 249] and inductive methods that try to learn from contextual information and metadata, e. g. attributes or recurrent structures, high-level patterns that should allow inference even when the underlying data changes [89, 255, 58].

In the following, we first discuss existing continual learning approaches for embeddings of time-varying KGs, which could potentially be used to analyse the evolution of KGs in the future.

Next, we discuss temporal embeddings, where instead of embedding changes to the KG, the objective is to embed temporal information in vector space as well, therefore having a temporal KG as input. This type of method inherently requires a different KG, one with temporal information. Lastly, we discuss some applications of learning for KGs with the evolving nature in mind.

We aim to provide a high-level overview of learning with regard to evolving KGs but do not claim to provide an in-depth survey of approaches. We specifically want to highlight known open challenges at the end of this section.

Continuous Embedding Learning

PuTransE provides a self-contained model, based on TransE, which builds on a metaphor of "parallel universes" [228]. It trains several parallel embedding spaces using different subgraphs. The retraining is then limited to some of the parallel universes instead of relearning the entire representation. *DKGE* is another self-contained model [249]. In this approach, the embedding of an entity consists of two parts, the embedding of the entity itself and its context embedding. Both puTransE and DKGE deal with the changing graph as a whole, but their scalability to larger graphs is limited. Song et al. [217] was one the first efforts regarding dynamic KG embeddings, focusing on the addition of new triples on translation-based models, which the authors refer to as enrichment. Cui et al. [53] present a transfer-based strategy for embedding generation for newly introduced entities. This self-contained model is based on auto-encoders and scales well with large graphs. Daruna et al. [54] extends and reformulates the principles of five main types of continual learning methods not specific to KGs. These criteria are applied to KG embedding models, each requiring a different kind of adjustment to fit the continual learning problem. All three methods [217, 53, 54] can only deal with additions and not with deletions or modifications. Lastly, the objective of Hamaguchi et al. [106] is slightly different. They rely on GNNs to generate embeddings for unseen entities at testing time but do not update and reuse the embedding for subsequent use.

All the methods above have drawbacks and there does not exist a go-to method so far to embed KGs continuously. The big challenges are (1) deterioration of the task performance as the embedding is updated and (2) dealing with deletions of triples or nodes.

Temporal Knowledge Graph Embeddings

The goal of temporal KG embeddings is to represent a time-annotated KG in a vector space. As such, these methods are completely different from the methods dealing with evolving snapshots of a KG. Many methods have been proposed for embedding temporal KGs and can be roughly separated into four categories: geometric, matrix factorisation, deep learning, and model-agnostic methods. There are some methods that are meant for dynamic temporal knowledge graphs; however, they only consider additions, arguing that deletions are not necessary for temporal knowledge graphs [148].

Geometric methods use geometrical transformations, such as translations and rotations, to represent the KG elements, e. g. HyTE [56] as an extension of TransE for temporal knowledge graphs: it incorporating time in the entity-relation space through a hyper-plane for each timestamp. TeRo [125] and ChronoR [203] use rotation transformations by creating multiple representations over time and creating time-dependent embeddings for relations respectively.

Matrix factorisation methods produce embeddings by decomposition tensors representing the KG. While a KG is usually represented in a 3rd-order tensor, a temporal KG can be represented in a 4th-order tensor, with the additional dimension representing time. For example, TNTComplEx [141] extends ComplEx. One of the main peculiarities of the method is that it distinguishes between non-temporal predicates and temporal facts.

Deep-learning methods exploit neural networks to learn the embeddings. For example, RE-Net [127] learns temporal KG embeddings using a recurrent neural network, while [151] uses convolutional neural networks to capture the time interaction between facts.

Finally, model-agnostic methods can be applied to time-agnostic KG embedding methods to add the temporal dimension. For example, the Diachronic Embeddings [96] represent the entity as a function of time and entity, while [145] provides a framework to extend methods to deal with arbitrary time granularities.

Applications of Learning on Evolving Knowledge Graphs

Learning on evolving KGs has been extensively used for completion and data integration tasks. Here, we aim to present some examples, not a complete overview.

Completion. Completion is the problem of inferring missing links in a knowledge graph. In recent years, many approaches have been proposed to address completion through KG embeddings. There, the completion problem can be targeted through the link prediction task, i. e. finding a missing element of a statement given the other two, or question answering, i. e. discovering unseen links through approximate query answering. However, KG completion also includes other tasks, namely triple completion, node classification, and relation prediction [212]. Many of the methods presented above have been proposed for the purpose of KG completion and also tested with that task specifically. Shen et al. [212] provide an up-to-date overview of approaches in this area without considering KG evolution. They divide the existing approaches into those only relying on structural information (the knowledge graph) and those that also make use of additional resources. Additionally, some more specialised approaches deal with temporal KGs and their embeddings, commonsense KG, and hyper-relational KGs. Since our goal is not to provide such an overview, we refer to the work of Shen et al. [212]. Other surveys, which might not cover all of KG completion like Rossi et al. [200] who only focus on link prediction or Wang et al. [244] who focus more on the embedding methods and their application. Lastly, Gesese et al. [94] gave an overview of approaches which specifically deal with literals.

Question answering. Then there are the approaches that are more specific for approximate query answering, though they can also be seen as KG completion approaches. When not using the graph information directly, it is possible to answer queries approximately by making use of implicit information, the same as with KG completion. These can be presented in a transductive [160, 49] or inductive setting [88]. There are emerging question-answering systems that target time-related questions. For example, Jia et al. [123] propose TEQUILA, a system that enriches question-answering systems with temporal question-answer capabilities. Three years later, Jia et al. [124] created EXAQT, which answers questions using graph convolutional networks enhanced with time-aware entity embeddings. Otte et al. [173] propose a question-answering system that exploits an ensemble of diachronic temporal KG embeddings.

Data integration. An important practical application of graph embeddings lies in their usage for data integration tasks on KGs. This has been particularly impactful in bio-medicine, where data has been accumulating at an unprecedented rate and where efficient solutions for uniformly integrating and processing them are particularly needed. The work in [72] introduces a semantic KG embedding approach for biomedical data. As such, the authors focus on integrating biomedical literature, e. g. MedLine and PubMed, with ontologies used to contextualise KG entities. At a larger scale, a case in point of KG data integration with embeddings is the Bioteque knowledge graph [85]. This integrates data from 150 sources and comprises 450K biological entities and 30M relationships. To reduce dimensionality, while still capturing the various types of relationships between entities, specific node embeddings are defined.

Future Directions

When it comes to continuous learning of KG embeddings, in light of an evolving KG as input, there are three main challenges still open. From previously published approaches, the deterioration of task performance is a known problem when continuously learning as new information arrives. Here we can also draw parallels to catastrophic forgetting in other continual learning tasks without KGs. Additionally, most approaches currently available for the continuous learning of embeddings, do not consider deletions but only additions. Therefore, being able to handle all manners of changes when embedding evolving KGs is an open challenge. Lastly, studies presented often only deal with a small number of updates to a KG, and hence, investigations are limited and need to be investigated at scale.

Embedding temporal knowledge graphs gained attention in recent years, and it is not at the same level of maturity of embedding techniques for knowledge graphs. One challenge lies in the definition of *temporal* knowledge graph, which is not standardised. Existing studies on the topic consider knowledge graphs where the temporal information is represented differently (see Section 5.1) and can have different semantics, e. g. time intervals where the fact is true or time instant where an event starts. Moreover, there is no set of well-defined and shared tasks, e. g. most studies focus on slightly different variations of the completion tasks, where time can or cannot be predicted. As a consequence, the existing methods are hardly comparable. Therefore, we envision the creation of de-facto standard datasets and tasks, which can help consolidate existing techniques and drive this research trend.

In parallel, as also mentioned in Section 6.2, there is an opportunity to enrich temporal knowledge graph embedding methods with deductive techniques. Specifically, in future, we expect novel research that combines embeddings, which are effective in capturing the structural information stored in a knowledge graph, with temporal logics, which have proven a robust solution to manage and reason on the temporal information.

By embedding a KG into a vector representation, we can potentially learn more about the evolution of the KG and conduct longitudinal analyses, e. g. of concept drift. However, due to the stochastic nature of the learning process, this remains a large open challenge, until the stochasticity problem is resolved to some extent [181]. We see a large number of open challenges when it comes to applications relying on embeddings of evolving KGs. Currently, we lack techniques and approaches for embedding-dependent tasks to be able to handle the changing KG without losing in performance or requiring complete recalculations. We can, however, also look at this from a slightly different perspective, that of the impact of evolution on these applications. When these applications first involve the learning of an embedding, it becomes extremely difficult to judge and capture the impact of evolution [181]. However, judging impact should not only be based on benchmark performance but rather the real impact in terms of changes to predictions. Therefore, we see an open challenge in analysing the performance of evolving tasks not in terms of metrics like mean-reciprocal-rank or accuracy, but rather the changes to the individual predictions. Approaches like inter-rater agreements may be useful for analysing localised changes in predictions [93].

6.4 Exploring Evolving Knowledge Graphs

When it comes to managing and analysing KGs, their heterogeneity constitutes both a defining characteristic and a challenge. In particular, both the contents and the schemas of these graphs have become less and less familiar even to domain experts and almost impenetrable to first-time users, leading to a rising need for exploratory methods for knowledge graphs [149, 150]. *Knowledge graph exploration* [149] is the machine-assisted and progressive process of analysis of a KG leading to (1) the understanding of the structure and nature of the graph, (2) the identification of which portion of the KG can satisfy the current information need, and (3) the extraction of insights that enable the formulation of novel research questions and hypotheses. These goals translate to three main tasks: (i) summarization and profiling, (ii) exploratory data analytics, and (iii) exploratory search. Looking at the dimension of evolution (Figure 1), we see that time adds a new dimension to the data to be explored and becomes a subject of exploration by itself when we explore how the structure (and not only the content) of the KG evolves and can provide new information that can then in turn guide the exploration.

Data profiling is the simplest form of exploration providing descriptive statistics and analysis about a given dataset. Typically, profiling tasks include counting the number of classes and their instances, summarising value distributions for specific (numerical) attributes, and they also identify important descriptors of the structure of the graph, e. g. node degree distribution [154]. There are also *structural summarization* [47] and *pattern mining* tasks [257, 194] to facilitate understanding the structure of the graph and to obtain concise representations of the most salient features of their contents. Profiling an evolving KG will provide insights into its structural changes through time, yet, only a few works scratch the surface of profiling KG evolution [76, 32]. They focus on analysing the statistical dataset characteristics at different snapshots [76], while more recent work started proposing algorithms to incrementally compute and update structural graph summaries defined as equivalence relations [32]. Therefore, to date, how to extend existing methods to tackle the challenges of scalable and continuous profiling of evolving KGs is still an open question. Moreover, as described above (Section 4.1), we are missing methods that can concisely summarise the results of a longitudinal analysis of the evolution of the schema and main characteristics of the dataset.

Exploratory analytics, is similar to data profiling since it is an iterative process of extracting aggregate information from portions of the graph, similar to a localised data profiling task [3, 51, 116]. The typical focus is to provide functionalities equivalent to those of *multi-dimensional analysis* that exist for relational data. Here, we see the need for analytical methods that can effectively include the temporal dimension in exploratory analytics, both when time is part of the data, as well as when time is treated as metadata. In this regard, we have recently witnessed a proposal to allow aggregation both at the attribute and at the time dimension [133, 239]. This is especially relevant since it offers the opportunity to employ exploration strategies that can guide the user through the evolution of the graph based on the identification of time intervals of significant growth, shrinkage, or stability of certain attribute values.

Finally, *Exploratory search* supports information needs that can be answered by retrieving specific entities, relationships, or paths. Exploratory queries change the traditional semantics of the search input: *instead of strictly prescribing the conditions that the desired result set must satisfy, they provide a hint of what is relevant* [149]. In these cases, the system should become an active agent able to suggest or infer query reformulations, refinements, and suggestions to help the user in their navigation. On the one hand, we see the need to help users explore the evolution of a given entity, e. g. identifying the most relevant changes w.r.t. a given stable state. On the other hand, the question is whether tapping into the analysis of the evolution of the KG, this information could be used to provide better suggestions or refinements. Overall, the methods designed to allow for query processing over evolving graphs (see Section 6.1) can still be used under the hood to enable exploratory search. Yet, to date, no method actively accounts for the rate and evolution of given entities and substructures when computing query suggestions to help the user in their exploration.

Future Directions

In summary, we identify both the need for new exploratory techniques that take into account the temporal dimension, and at the same time we highlight how existing techniques need to face the computational challenges posed by a KG that is not static anymore but dynamic. In particular, we postulate the need for new KG profiling techniques that apply longitudinal analysis to the data model in the KG through its lifespan. Furthermore, they see the need for methods that can understand trends in graph-centric measures and can efficiently compute and measure their evolution over time while the graph evolves. Finally, we ask which signals can be extracted from the observation of the evolution of the graph that can be exploited as a signal to help users identify interesting information and to identify methods to assist users in navigating more easily through an unfamiliar KG.

7 Summary and Conclusions

While KGs are gaining attention overall, the analysis and management of their evolution is still a "less conquered" territory in research. The present paper encourages us to look closely at KG evolution and make it a more prominent subject in our research. After emphasising that different types of KGs likely have very different change and evolution characteristics, we motivated various dimensions of looking at the evolution of KGs. We started investigating how known static structural analyses of KGs can be considered in a dynamic context, exploring the evolution of quality and consistency over time, to specific aspects related to dynamic collaboration processes of KG contributors, and finally, semantic drift in KGs. We provided an overview of publicly available KGs and, specifically, the availability of historical longitudinal data about their evolution that could serve as a starting point for analyses, as well as an overview of already existing studies.

We identified a research gap in terms of specific metrics for studying KG evolution in different dimensions; here, in the future, we will need to address concerns regarding the application and adaption of static metrics for longitudinal and time-series analyses on KGs. In particular, regarding the analysis of KG consistency over time, we have sketched viable approaches in Section 4.2; however, these have not yet been applied in an analysis of KG evolution, presenting a notable research gap.

Finally, we had a detailed discussion about the metrics and techniques that can be applied to analyse KGs. We suggested exploring more methods not commonly used in our community but well-established in other fields, such as network science. This field has a long-standing tradition of analysing large-scale networks' structural and dynamic aspects. Given the extensive reach and rapid growth of KGs, it is imperative to implement similar methods in our field. However, we should remember that these methods may require adaptations due to the "multi-level" network characteristic of KGs, as they can be viewed as overlaid networks encompassing all their properties.

We further discussed challenges related to different graph representation models and storage strategies for the extraction/construction of dynamic KGs. They focus mainly on the interoperability of the different ways time is captured in evolving KGs, different schemas and their mapping to each other, and how these could be integrated in the future, for instance in standardised ways to query evolving KGs. Regarding storage, currently, different storage solutions facilitate different types of analyses. Still, in the future, we hope to see storage solutions enabling the storage of dynamic and versioned graphs to enable all kinds of analyses.

The popular downstream tasks when using knowledge graphs, such as querying, reasoning, and learning, can benefit from considering the evolution of knowledge more explicitly. Considering the temporal dimension as a first-class citizen at the query level opens the possibility to specific operators for retrieving data about time, through time, and in time. In the future, we can expect further extensions of SPARQL and other LPG-specific query languages to support these operators, ideally combining temporal, versioned, and continuous flavours in more comprehensive query languages. Similarly, reasoning is affected by evolving knowledge. On the one hand, there are new algorithmic challenges, e. g. how to maintain a materialisation incrementally and reactively (on time). On the other hand, considering temporal logics at a fundamental level could enhance reasoning over evolving KGs and their schemas over time. KG evolution can also provide additional signals for training machine learning models, capturing dynamic processes. However, respective approaches that for instance capture updates in learned embeddings, are still lacking in performance and scalability to be helpful in practical analytical use cases. Finally, we envision querying, reasoning, and learning to be fruitfully combined to overcome individual weaknesses for managing, processing and analysing evolving KGs, eventually creating new applications and services. While such combinations have been studied for static KGs, we expect and hope to see more studies in the future that consider the evolving knowledge case.

In the following list, we summarise the most important future directions and open challenges, in particular about learning more about and understanding how knowledge evolves in open, general-purpose KGs:

- *Systematic analysis* of open general-purpose KGs along various dimensions of evolution such as dynamics, timeliness and monotonicity, but also structural, semantic and collaborative aspects making use of approaches such as time-series analysis and network science.
- Principled approaches to *create evolving subsets* of KGs in evolution for specific benchmarking tasks would be dearly needed to better understand these tasks "in evolution".
- Further development of *metrics for measuring and understanding knowledge evolution in KGs*, specifically capable of handling outliers and the complexity and size of the known KGs.
- *Interoperability between different KG models*, mainly RDF and LPG, and query languages that support these to enable better and complementary analyses of temporal KGs.
- Development of adaptive approaches and respective querying capabilities to store dynamic and versioned KGs simultaneously, making it possible to apply any analysis (time-series and constraint-based) on the evolving KGs.
- The *combination of deductive and inductive techniques* [65] is necessary to tackle challenges with reasoning (scale, completeness, errors) as it leads to data and knowledge-driven techniques. For example, one may complement the evolving knowledge graph and identify and correct potentially wrong new facts.
- Development of novel *continuous embedding approaches and methods for embedding temporal KGs*, i. e., the study of concept drift with large evolving KGs from different perspectives becomes a new open challenge.
- Tackling the computational challenges of existing *exploratory techniques* and the development of new ones specifically facilitating longitudinal analysis through, e. g. graph-centric measures to help navigate the evolution of an unfamiliar KG.

In summary, we have performed an extensive survey of evolution in KGs - significantly more extensive than initially expected. From this survey we conclude that KG's evolution is apparently a field that – while having already attracted a lot of attention – remains to have various open questions. The authors hope we motivated the readers to work jointly on more in-depth investigations and more standardised, agreed-upon methods of capturing and dealing with Knowledge (Graph) Evolution as well as newer methods for analysis as identified in this work.

References

1 Mohammad Abdel-Qader, Ansgar Scherp, and Iacopo Vagliano. Analyzing the evolution of vocabulary terms and their impact on the LOD cloud. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science (LNCS)*, pages 1–16. Springer, 2018. doi:10.1007/978-3-319-93417-4_1.

2 Mohammad Abdel-Qader, Iacopo Vagliano, and Ansgar Scherp. Analyzing the Evolution of Linked Vocabularies. In Maxim Bakaev, Flavius Frasincar, and In-Young Ko, editors, *Web Engineering - 19th International Conference, ICWE 2019, Daejeon, South Korea, June 11-14, 2019, Proceedings*, volume 11496 of *Lecture Notes in Computer Science (LNCS)*, pages 409–424. Springer, 2019. doi:10.1007/978-3-030-19274-7_29.

3 Alberto Abelló, Oscar Romero, Torben Bach Pedersen, Rafael Berlanga Llavori, Victoria Nebot, María José Aramburu Cabo, and Alkis Simitsis. Using semantic web technologies for exploratory OLAP: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):571–588, 2015. doi:10.1109/TKDE.2014.2330822.

4 Ghadeer Abuoda, Daniele Dell'Aglio, Arthur Keen, and Katja Hose. Transforming rdf-star to property graphs: A preliminary analysis of transformation approaches. In Muhammad Saleem and Axel-Cyrille Ngonga Ngomo, editors, *Proceedings of the QuWeDa 2022: 6th Workshop on Storing, Querying and Benchmarking Knowledge Graphs co-located with 21st International*

Semantic Web Conference (ISWC 2022), Hangzhou, China, 23-27 October 2022*, volume 3279 of *CEUR Workshop Proceedings*, pages 17–32. CEUR-WS.org, 2022. URL: `https://ceur-ws.org/Vol-3279/paper2.pdf`.

5 Christian Aebeloe, Gabriela Montoya, and Katja Hose. ColChain: Collaborative Linked Data Networks. In *Proceedings of the Web Conference 2021*, pages 1385–1396, Ljubljana Slovenia, apr 2021. ACM. `doi:10.1145/3442381.3450037`.

6 Temitope Ajileye and Boris Motik. Materialisation and data partitioning algorithms for distributed RDF systems. *Journal of Web Semantics*, 73:100711, 2022. `doi:10.1016/J.WEBSEM.2022.100711`.

7 Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, jan 2002. `doi:10.1103/RevModPhys.74.47`.

8 Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. Bringing Light Into the Dark: A Large-Scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8825–8845, dec 2022. `doi:10.1109/TPAMI.2021.3124805`.

9 James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983. `doi:10.1145/182.358434`.

10 Francesca Alloatti, Riley Capshaw, Molka Dhouib, María G. Buey, Ismail Harrando, Jaime Salas, and Claudia d'Amato. Towards an automatic detection of evolution in knowledge graphs. In Valentina Anita Carriero, Luigi Asprino, and Russa Biswas, editors, *Knowledge Graphs Evolution and Preservation - A Technical Report from ISWS 2019*, volume abs/2012.11936 of *Computing Research Repository (CoRR)*. arXiv, 2020. `doi:10.48550/arXiv.2012.11936`.

11 Rebekka Alm, Sven Kiehl, Birger Lantow, and Kurt Sandkuhl. Applicability of quality metrics for ontologies on ontology design patterns. In Joaquim Filipe and Jan L. G. Dietz, editors, *KEOD 2013 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Vilamoura, Algarve, Portugal, 19-22 September, 2013*, pages 48–57. SciTePress, 2013. `doi:10.5220/0004541400480057`.

12 Landy Andriamampianina, Franck Ravat, Jiefu Song, and Nathalie Vallès-Parlangeau. Querying Temporal Property Graphs. In Xavier Franch, Geert Poels, Frederik Gailly, and Monique Snoeck, editors, *Advanced Information Systems Engineering - 34th International Conference, CAiSE 2022, Leuven, Belgium, June 6-10, 2022, Proceedings*, volume 13295 of *Lecture Notes in Computer Science (LNCS)*, pages 355–370. Springer, 2022. `doi:10.1007/978-3-031-07472-1_21`.

13 Renzo Angles, Carlos Buil Aranda, Aidan Hogan, Carlos Rojas, and Domagoj Vrgoč. WDBench: A Wikidata Graph Query Benchmark. In Ulrike Sattler, Aidan Hogan, Maria Keet, Valentina Presutti, João Paulo A. Almeida, Hideaki Takeda, Pierre Monnin, Giuseppe Pirrò, and Claudia d'Amato, editors, *The Semantic Web – ISWC 2022*, pages 714–731, Cham, 2022. Springer International Publishing. `doi:10.1007/978-3-031-19433-7_41`.

14 Renzo Angles, Angela Bonifati, Stefania Dumbrava, George Fletcher, Alastair Green, Jan Hidders, Bei Li, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Stefan Plantikow, Ognjen Savkovic, Michael Schmidt, Juan Sequeda, Slawek Staworko, Dominik Tomaszuk, Hannes Voigt, Domagoj Vrgoc, Mingxi Wu, and Dusan Zivkovic. PG-Schema: Schemas for Property Graphs. *Proceedings of the ACM on Management of Data*, 1(2):198:1–198:25, 2023. `doi:10.1145/3589778`.

15 Renzo Angles, Harsh Thakkar, and Dominik Tomaszuk. Mapping RDF Databases to Property Graph Databases. *IEEE Access*, 8:86091–86110, 2020. `doi:10.1109/ACCESS.2020.2993117`.

16 Darko Anicic, Paul Fodor, Sebastian Rudolph, and Nenad Stojanovic. EP-SPARQL: a unified language for event processing and stream reasoning. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar, editors, *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, pages 635–644. ACM, 2011. `doi:10.1145/1963405.1963495`.

17 Arvind Arasu, Shivnath Babu, and Jennifer Widom. The CQL continuous query language: semantic foundations and query execution. *VLDB Journal*, 15(2):121–142, 2006. `doi:10.1007/S00778-004-0147-Z`.

18 Marcelo Arenas, Pedro Bahamondes, Amir Aghasadeghi, and Julia Stoyanovich. Temporal regular path queries. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*, pages 2412–2425. IEEE, 2022. `doi:10.1109/ICDE53745.2022.00226`.

19 Natanael Arndt, Patrick Naumann, Norman Radtke, Michael Martin, and Edgard Marx. Decentralized Collaborative Knowledge Management Using Git. *Journal of Web Semantics*, 54:29–47, jan 2019. `doi:10.1016/J.WEBSEM.2018.08.002`.

20 Sören Auer. The emerging web of linked data. In Ayman Jameel Alnsour and Shadi A. Aljawarneh, editors, *Proceedings of the 2nd International Conference on Intelligent Semantic Web-Services and Applications, ISWSA 2011, Amman, Jordan, April 18-20, 2011*, page 1. ACM, 2011. `doi:10.1145/1980822.1980823`.

21 Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, Cambridge, UK, 2 edition, 2007. `doi:10.1017/CBO9780511711787`.

22 Thomas Baker and Eric Prud'hommeaux. Shape Expressions (ShEx) 2.1 Primer. Technical report, W3C Shape Expressions Community Group, oct 2019. URL: `http://shex.io/shex-primer/`.

23 Marco Balduini, Irene Celino, Daniele Dell'Aglio, Emanuele Della Valle, Yi Huang, Tony Kyung-il Lee, Seon-Ho Kim, and Volker Tresp. Reality mining on micropost streams. *Semantic Web*, 5(5):341–356, 2014. `doi:10.3233/SW-130107`.

24 Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. Incremental reasoning on streams and rich background knowledge. In Lora Aroyo, Grigoris Antoniou, Eero Hyvönen, Annette ten Teije, Heiner Stuckenschmidt, Liliana Cabral, and Tania Tudorache, editors, *The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010, Proceedings, Part I*, volume 6088 of *Lecture Notes in Computer Science (LNCS)*, pages 1–15. Springer, 2010. doi:10.1007/978-3-642-13486-9_1.

25 Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, 874:1–92, aug 2020. doi:10.1016/j.physrep.2020.05.004.

26 Federico Battiston, Vincenzo Nicosia, and Vito Latora. Structural measures for multiplex networks. *Physical Review E*, 89(3):032804, mar 2014. doi:10.1103/PhysRevE.89.032804.

27 Harald Beck, Minh Dao-Tran, Thomas Eiter, and Michael Fink. LARS: A logic-based framework for analyzing reasoning over streams. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 1431–1438. AAAI Press, 2015. doi:10.1609/AAAI.V29I1.9408.

28 Zohra Bellahsene, Angela Bonifati, and Erhard Rahm, editors. *Schema Matching and Mapping*. Data-Centric Systems and Applications. Springer, 2011. doi:10.1007/978-3-642-16518-4.

29 Luigi Bellomarini, Davide Benedetto, Georg Gottlob, and Emanuel Sallinger. Vadalog: A modern architecture for automated reasoning with large knowledge graphs. *Information Systems*, 105:101528, 2022. doi:10.1016/J.IS.2020.101528.

30 Konstantina Bereta, Panayiotis Smeros, and Manolis Koubarakis. Representation and Querying of Valid Time of Triples in Linked Geospatial Data. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *The Semantic Web: Semantics and Big Data*, volume 7882, pages 259–274, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. doi:10.1007/978-3-642-38288-8_18.

31 Philip A. Bernstein. Applying model management to classical meta data problems. In *First Biennial Conference on Innovative Data Systems Research, CIDR 2003, Asilomar, CA, USA, January 5-8, 2003, Online Proceedings*. www.cidrdb.org, 2003. URL: http://www-db.cs.wisc.edu/cidr/cidr2003/program/p19.pdf.

32 Till Blume, David Richerby, and Ansgar Scherp. Incremental and parallel computation of structural graph summaries for evolving graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pages 75–84, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3340531.3411878.

33 Till Blume and Ansgar Scherp. Indexing data on the web: A comparison of schema-level indices for data search. In Sven Hartmann, Josef Küng, Gabriele Kotsis, A Min Tjoa, and Ismail Khalil, editors, *Database and Expert Systems Applications - 31st International Conference, DEXA 2020, Bratislava, Slovakia, September 14-17, 2020, Proceedings, Part II*, volume 12392 of *Lecture Notes in Computer Science (LNCS)*, pages 277–286. Springer, 2020. doi:10.1007/978-3-030-59051-2_18.

34 Olivier Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(Database-Issue):267–270, 2004. doi:10.1093/NAR/GKH061.

35 Michael H. Böhlen, Anton Dignös, Johann Gamper, and Christian S. Jensen. Temporal data management - an overview. In Esteban Zimányi, editor, *Business Intelligence and Big Data - 7th European Summer School, eBISS 2017, Bruxelles, Belgium, July 2-7, 2017, Tutorial Lectures*, volume 324 of *Lecture Notes in Business Information Processing*, pages 51–83. Springer, 2017. doi:10.1007/978-3-319-96655-7_3.

36 Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1376616.1376746.

37 ES Bolotnikova, TA Gavrilova, and VA Gorovoy. To a method of evaluating ontologies. *Journal of Computer and Systems Sciences International*, 50(3):448–461, 2011. doi:10.1134/S1064230711010072.

38 Angela Bonifati, George H. L. Fletcher, Hannes Voigt, and Nikolay Yakovets. *Querying Graphs*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2018. doi:10.2200/S00873ED1V01Y201808DTM051.

39 Angela Bonifati, Irena Holubová, Arnau Prat-Pérez, and Sherif Sakr. Graph Generators: State of the Art and Open Challenges. *ACM Computing Surveys*, 53(2):1–30, mar 2021. doi:10.1145/3379445.

40 Angela Bonifati, Wim Martens, and Thomas Timm. An analytical study of large SPARQL query logs. *VLDB Journal*, 29(2-3):655–679, 2020. doi:10.1007/S00778-019-00558-9.

41 Pieter Bonte and Riccardo Tommasini. Streaming linked data: A survey on life cycle compliance. *Journal of Web Semantics*, 77:100785, 2023. doi:10.1016/J.WEBSEM.2023.100785.

42 Dan Brickley and Ramanathan Guha. RDF Schema 1.1. Technical report, The World Wide Web Consortium (W3C), feb 2014. URL: http://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

43 Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. Temporal know-

ledge graph completion: A survey. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6545–6553. International Joint Conferences on Artificial Intelligence Organization, aug 2023. Survey Track. doi:10.24963/IJCAI.2023/734.

44 Giuseppe Capobianco, Danilo Cavaliere, and Sabrina Senatore. Ontodrift: a semantic drift gauge for ontology evolution monitoring. In Fabrizio Orlandi, Damien Graux, Maria-Esther Vidal, Javier D. Fernández, and Jeremy Debattista, editors, *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), Virtual event (instead of Athens, Greece), November 1st, 2020*, volume 2821 of *CEUR Workshop Proceedings*, pages 1–10. CEUR-WS.org, 2020. URL: https://ceur-ws.org/Vol-2821/xpreface.pdf.

45 Gavin Carothers. RDF 1.1 N-Quads - A line-based syntax for RDF datasets. Technical report, The World Wide Web Consortium (W3C), 2014. URL: http://www.w3.org/TR/2014/REC-n-quads-20140225/.

46 David Carral, Irina Dragoste, Larry González, Ceriel J. H. Jacobs, Markus Krötzsch, and Jacopo Urbani. VLog: A Rule Engine for Knowledge Graphs. In Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtech Svátek, Isabel F. Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon, editors, *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, volume 11779 of *Lecture Notes in Computer Science (LNCS)*, pages 19–35. Springer, 2019. doi:10.1007/978-3-030-30796-7_2.

47 Sejla Cebiric, François Goasdoué, Haridimos Kondylakis, Dimitris Kotzinos, Ioana Manolescu, Georgia Troullinou, and Mussab Zneika. Summarizing semantic graphs: a survey. *VLDB Journal*, 28(3):295–327, 2019. doi:10.1007/S00778-018-0528-3.

48 Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, Xi Chen, and Erik Bryhn Myklebust. An assertion and alignment correction framework for large scale knowledge bases. *Semantic Web*, 14(1):29–53, 2023. doi:10.3233/SW-210448.

49 Xuelu Chen, Ziniu Hu, and Yizhou Sun. Fuzzy logic based logical query answering on knowledge graphs. In *Thirty-Sixth Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, AAAI 2022, February 22 - March 1, 2022*, pages 3939–3948. AAAI Press, 2022. doi:10.1609/AAAI.V36I4.20310.

50 Maxime Clément and Matthieu J. Guitton. Interacting with bots online: Users' reactions to actions of automated programs in Wikipedia. *Computers in Human Behavior*, 50:66–75, 2015. doi:10.1016/J.CHB.2015.03.078.

51 Dario Colazzo, François Goasdoué, Ioana Manolescu, and Alexandra Roatis. RDF analytics: lenses over semantic graphs. In Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel, editors, *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 467–478. ACM, 2014. doi:10.1145/2566486.2567982.

52 Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys*, 44(3):15:1–15:62, 2012. doi:10.1145/2187671.2187677.

53 Yuanning Cui, Yuxin Wang, Zequn Sun, Wenqiang Liu, Yiqiao Jiang, Kexin Han, and Wei Hu. Lifelong embedding learning and transfer for growing knowledge graphs. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *Thirty-Seventh Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, AAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 4217–4224. AAAI Press, 2023. doi:10.1609/AAAI.V37I4.25539.

54 Angel Andres Daruna, Mehul Gupta, Mohan Sridharan, and Sonia Chernova. Continual learning of knowledge graph embeddings. *IEEE Robotics and Automation Letters*, 6(2):1128–1135, 2021. doi:10.1109/LRA.2021.3056071.

55 Souripriya Das, Seema Sundara, and Richard Cyganiak. R2RML: RDB to RDF mapping language. Technical report, The World Wide Web Consortium (W3C), sep 2012. URL: http://www.w3.org/TR/2012/REC-r2rml-20120927/.

56 Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha P. Talukdar. HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2001–2011. Association for Computational Linguistics, 2018. doi:10.18653/V1/D18-1225.

57 Robert David, Shqiponja Ahmetaj, Mantas Šimkus, and Axel Polleres. Repairing SHACL constraint violations using answer set programming. In *Proceedings of the 21st International Semantic Web Conference (ISWC 2022)*, volume 13489 of *Lecture Notes in Computer Science (LNCS)*, pages 375–391, Virtual Conference (Hangzhou, China), oct 2022. Springer. doi:10.1007/978-3-031-19433-7_22.

58 Daniel Daza, Michael Cochez, and Paul Groth. Inductive Entity Representations from Text via Link Prediction. In *Proceedings of the Web Conference 2021*, WWW '21, pages 798–808, New York, NY, USA, jun 2021. Association for Computing Machinery. doi:10.1145/3442381.3450141.

59 Daniel de Leng and Fredrik Heintz. DyKnow: A dynamically reconfigurable stream reasoning framework as an extension to the robot operating system. In *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots, SIMPAR 2016, San Francisco, CA, USA, December 13-16, 2016*, pages 55–60. IEEE, 2016. doi:10.1109/SIMPAR.2016.7862375.

60 Ariel Debrouvier, Eliseo Parodi, Matías Perazzo, Valeria Soliani, and Alejandro A. Vaisman. A model and query language for temporal graph databases. *VLDB J.*, 30(5):825–858, 2021. doi:10.1007/S00778-021-00675-4.

61 Ariel Debrouvier, Eliseo Parodi, Matías Perazzo, Valeria Soliani, and Alejandro A. Vaisman. A model and query language for temporal graph databases. *VLDB Journal*, 30(5):825–858, 2021. doi:10.1007/S00778-021-00675-4.

62 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. doi:10.1109/MIS.2009.125.

63 Daniele Dell'Aglio, Emanuele Della Valle, Jean-Paul Calbimonte, and Óscar Corcho. RSP-QL semantics: A unifying query model to explain heterogeneity of RDF stream processing systems. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(4):17–44, 2014. doi:10.4018/IJSWIS.2014100102.

64 Daniele Dell'Aglio, Emanuele Della Valle, Frank van Harmelen, and Abraham Bernstein. Stream reasoning: A survey and outlook. *Data Science*, 1(1-2):59–83, 2017. doi:10.3233/DS-170006.

65 Lauren Nicole Delong, Ramon Fernández Mir, Matthew Whyte, Zonglin Ji, and Jacques D. Fleuriot. Neurosymbolic AI for reasoning on graph structures: A survey. *Computing Research Repository (CoRR)*, abs/2302.07200, 2023. doi:10.48550/ARXIV.2302.07200.

66 Anton Dignös, Michael H. Böhlen, and Johann Gamper. Temporal alignment. In K. Selçuk Candan, Yi Chen, Richard T. Snodgrass, Luis Gravano, and Ariel Fuxman, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 433–444. ACM, 2012. doi:10.1145/2213836.2213886.

67 Yuehang Ding, Hongtao Yu, Ruiyang Huang, and Yunjie Gu. Complex network based knowledge graph ontology structure analysis. In *2018 1st IEEE international conference on hot information-centric networking (HotICN)*, pages 193–199, 2018. doi:10.1109/HOTICN.2018.8606002.

68 Renata Queiroz Dividino, Thomas Gottron, and Ansgar Scherp. Strategies for Efficiently Keeping Local Linked Open Data Caches Up-To-Date. In Marcelo Arenas, Óscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d'Aquin, Kavitha Srinivas, Paul Groth, Michel Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, and Steffen Staab, editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*, volume 9367 of *Lecture Notes in Computer Science (LNCS)*, pages 356–373. Springer, 2015. doi:10.1007/978-3-319-25010-6_24.

69 Renata Queiroz Dividino, Thomas Gottron, Ansgar Scherp, and Gerd Gröner. From Changes to Dynamics: Dynamics Analysis of Linked Open Data Sources. In Elena Demidova, Stefan Dietze, Julian Szymanski, and John G. Breslin, editors, *Proceedings of the 1st International Workshop on Dataset PROFIling & fEderated Search for Linked Data co-located with the 11th Extended Semantic Web Conference, PROFILES@ESWC 2014, Anissaras, Crete, Greece, May 26, 2014*, volume 1151 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014. URL: https://ceur-ws.org/Vol-1151/paper4.pdf.

70 Renata Queiroz Dividino, Ansgar Scherp, Gerd Gröner, and Thomas Grotton. Change-a-lod: Does the schema on the linked data cloud change or not? In Olaf Hartig, Juan F. Sequeda, Aidan Hogan, and Takahide Matsutsuka, editors, *Proceedings of the Fourth International Workshop on Consuming Linked Data, COLD 2013, Sydney, Australia, October 22, 2013*, volume 1034 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013. URL: https://ceur-ws.org/Vol-1034/DividinoEtAl_COLD2013.pdf.

71 Rim Djedidi and Marie-Aude Aufaure. ONTO-EVO A L an ontology evolution approach guided by pattern modeling and quality evaluation. In *International symposium on foundations of information and knowledge systems*, pages 286–305, 2010. doi:10.1007/978-3-642-11829-6_19.

72 Jens Dörpinghaus and Marc Jacobs. Semantic knowledge graph embeddings for biomedical research: Data integration using linked open data. In Mehwish Alam, Ricardo Usbeck, Tassilo Pellegrini, Harald Sack, and York Sure-Vetter, editors, *Proceedings of the Posters and Demo Track of the 15th International Conference on Semantic Systems co-located with 15th International Conference on Semantic Systems (SEMANTiCS 2019), Karlsruhe, Germany, September 9th - to - 12th, 2019*, volume 2451 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019. URL: https://ceur-ws.org/Vol-2451/paper-10.pdf.

73 Astrid Duque-Ramos, Manuel Quesada-Martínez, Miguela Iniesta-Moreno, Jesualdo Tomás Fernández-Breis, and Robert Stevens. Supporting the analysis of ontology evolution processes through the combination of static and dynamic scaling functions in OQuaRE. *Journal of Biomedical Semantics*, 7(1):63, oct 2016. doi:10.1186/S13326-016-0091-Z.

74 Thomas Eiter, Andreas A. Falkner, Patrik Schneider, and Peter Schüller. ASP-Based signal plan adjustments for traffic flow optimization. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *ECAI 2020 - 24th european conference on artificial intelligence, 29 august-8 september 2020, santiago de compostela, spain, august 29 - september 8, 2020 - including 10th conference on prestigious applications of artificial intelligence (PAIS 2020)*, volume 325 of *Frontiers in artificial intelligence and applications*, pages 3026–3033. IOS Press, 2020. doi:10.3233/FAIA200478.

75 Mohamed G. Elfeky, Walid G. Aref, and Ahmed K. Elmagarmid. Periodicity Detection in Time Series Databases. *IEEE Transactions on Knowledge and Data Engineering*, 17(7):875–887, 2005. doi:10.1109/TKDE.2005.114.

76 Ivan Ermilov, Michael Martin, Jens Lehmann, and Sören Auer. Linked open data statistics: Collection and exploitation. In Pavel Klinov and Dmitry Mouromtsev, editors, *Knowledge Engineering and the Semantic Web - 4th International Conference, KESW 2013, St. Petersburg, Russia, October 7-9, 2013. Proceedings*, volume 394

of *Communications in Computer and Information Science*, pages 242–249. Springer, 2013. doi:10.1007/978-3-642-41360-5_19.

77 Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching, Second Edition*. Springer, 2013.

78 Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005. doi:10.1016/J.TCS.2004.10.033.

79 Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F. Cruz, and Francisco M. Couto. The AgreementMakerLight Ontology Matching System. In Robert Meersman, Hervé Panetto, Tharam S. Dillon, Johann Eder, Zohra Bellahsene, Norbert Ritter, Pieter De Leenheer, and Dejing Dou, editors, *On the Move to Meaningful Internet Systems: OTM 2013 Conferences - Confederated International Conferences: CoopIS, DOA-Trusted Cloud, and ODBASE 2013, Graz, Austria, September 9-13, 2013. Proceedings*, volume 8185 of *Lecture Notes in Computer Science (LNCS)*, pages 527–541. Springer, 2013. doi:10.1007/978-3-642-41030-7_38.

80 Javier D. Fernández, Patrik Schneider, and Jürgen Umbrich. The DBpedia wayback machine. In Axel Polleres, Tassilo Pellegrini, Sebastian Hellmann, and Josiane Xavier Parreira, editors, *Proceedings of the 11th International Conference on Semantic Systems, SEMANTiCS 2015, Vienna, Austria, September 15-17, 2015*, pages 192–195. ACM, 2015. doi:10.1145/2814864.2814889.

81 Javier D. Fernández, Jürgen Umbrich, Axel Polleres, and Magnus Knuth. Evaluating query and storage strategies for RDF archives. In *Proceedings of the 12th International Conference on Semantic Systems*, SEMANTiCS 2016, pages 41–48, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2993318.2993333.

82 Javier D. Fernández, Wouter Beek, Miguel A. Martínez-Prieto, and Mario Arias. LOD-a-lot. In Claudia d'Amato, Miriam Fernandez, Valentina Tamma, Freddy Lecue, Philippe Cudré-Mauroux, Juan Sequeda, Christoph Lange, and Jeff Heflin, editors, *The Semantic Web – ISWC 2017*, Lecture Notes in Computer Science (LNCS), pages 75–83, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-68204-4_7.

83 Javier D. Fernández, Miguel A. Martínez-Prieto, Claudio Gutierrez, Axel Polleres, and Mario Arias. Binary RDF representation for publication and exchange (HDT). *Web Semantics*, 19:22–41, 2013. doi:10.1016/J.WEBSEM.2013.01.002.

84 Javier D. Fernández, Jürgen Umbrich, Axel Polleres, and Magnus Knuth. Evaluating query and storage strategies for RDF archives. *Semantic Web*, 10(2):247–291, 2019. doi:10.3233/SW-180309.

85 Adrià Fernández-Torras, Miquel Duran-Frigola, Martino Bertoni, Martina Locatelli, and Patrick Aloy. Integrating and formatting biomedical data as pre-calculated knowledge graph embeddings in the Bioteque. *Nature Communications*, 13, sep 2022. doi:10.1038/s41467-022-33026-0.

86 Nicolas Ferranti, Axel Polleres, Jairo Francisco de Souza, and Shqiponja Ahmetaj. Formalizing property constraints in wikidata. In Lucie-Aimée Kaffee, Simon Razniewski, Gabriel Amaral, and Kholoud Saad Alghamdi, editors, *Proceedings of the 3rd Wikidata Workshop 2022 co-located with the 21st International Semantic Web Conference (ISWC2022), Virtual Event, Hanghzou, China, October 2022*, volume 3262 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3262/paper1.pdf.

87 Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, feb 2010. doi:10.1016/j.physrep.2009.11.002.

88 Michael Galkin, Zhaocheng Zhu, Hongyu Ren, and Jian Tang. Inductive logical query answering in knowledge graphs. In *Advances in Neural Information Processing Systems*, 2022. doi:10.48550/ARXIV.2210.08008.

89 Mikhail Galkin, Etienne G. Denis, Jiapeng Wu, and William L. Hamilton. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL: https://openreview.net/forum?id=xMJWUKJnFSw.

90 João Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014. doi:10.1145/2523813.

91 Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, and Jos Lehmann. Modelling ontology evaluation and validation. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings*, volume 4011 of *Lecture Notes in Computer Science (LNCS)*, pages 140–154. Springer, 2006. doi:10.1007/11762256_13.

92 Shi Gao, Jiaqi Gu, and Carlo Zaniolo. RDF-TX: A Fast, User-Friendly System for Querying the History of RDF Knowledge Bases. In *19th International Conference on Extending Database Technology (EDBT)*, 2016. doi:10.5441/002/EDBT.2016.26.

93 Robert Geirhos, Kristof Meding, and Felix A. Wichmann. Beyond accuracy: quantifying trial-by-trial behaviour of CNNs and humans by measuring error consistency. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020. doi:10.48550/arXiv.2006.16736.

94 Genet Asefa Gesese, Russa Biswas, and Harald Sack. A Comprehensive Survey of Knowledge Graph Embeddings with Literals: Techniques and Applications. In Mehwish Alam, Davide Buscaldi, Michael Cochez, Francesco Osborne, Diego Reforgiato Recupero, and Harald Sack, editors, *Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG2019) Co-located with the 16th Extended Semantic Web Conference 2019 (ESWC 2019), Portoroz, Slovenia, June 2,*

2019, volume 2377 of *CEUR Workshop Proceedings*, pages 31–40. CEUR-WS.org, 2019. URL: https://ceur-ws.org/Vol-2377/paper_4.pdf.

95 Birte Glimm, Aidan Hogan, Markus Krötzsch, and Axel Polleres. OWL: yet to arrive on the web of data? In Christian Bizer, Tom Heath, Tim Berners-Lee, and Michael Hausenblas, editors, *WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012*, volume 937 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012. URL: https://ceur-ws.org/Vol-937/ldow2012-paper-16.pdf.

96 Rishab Goel, Seyed Mehran Kazemi, Marcus A. Brubaker, and Pascal Poupart. Diachronic embedding for temporal knowledge graph completion. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, , new york, NY, USA, february 7-12, 2020*, pages 3988–3995. AAAI Press, 2020. doi:10.1609/AAAI.V34I04.5815.

97 Rafael S. Gonçalves, Matthew Horridge, Rui Li, Yu Liu, Mark A. Musen, Csongor I. Nyulas, Evelyn Obamos, Dhananjay Shrouty, and David Temple. Use of owl and semantic web technologies at pinterest. In Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtěch Svátek, Isabel Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon, editors, *The Semantic Web – ISWC 2019*, pages 418–435, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-30796-7_26.

98 Thomas Gottron and Christian Gottron. Perplexity of Index Models over Evolving Linked Data. In *The Semantic Web: Trends and Challenges: 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings 11*, volume 8465, pages 161–175. Springer, 2014. doi:10.1007/978-3-319-07443-6_12.

99 Simon Gottschalk and Elena Demidova. Eventkg - the hub of event knowledge on the web - and biographical timeline generation. *Semantic Web*, 10(6):1039–1070, 2019. doi:10.3233/SW-190355.

100 Fabio Grandi. T-SPARQL: A tsql2-like temporal query language for RDF. In Mirjana Ivanovic, Bernhard Thalheim, Barbara Catania, and Zoran Budimac, editors, *Local Proceedings of the Fourteenth East-European Conference on Advances in Databases and Information Systems, Novi Sad, Serbia, September 20-24, 2010*, volume 639 of *CEUR Workshop Proceedings*, pages 21–30. CEUR-WS.org, 2010. URL: https://ceur-ws.org/Vol-639/021-grandi.pdf.

101 OWL Working group. OWL 2 web ontology language overview (second edition). Technical report, The World Wide Web Consortium (W3C), dec 2012. URL: http://www.w3.org/TR/2012/REC-owl2-overview-20121211/.

102 Ramanathan V Guha, Dan Brickley, and Steve Macbeth. Schema.org: evolution of structured data on the web. *Communications of the ACM*, 59(2):44–51, 2016. doi:10.1145/2844544.

103 Claudio Gutierrez, Carlos A. Hurtado, and Alejandro Vaisman. Introducing Time into RDF. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):207–218, 2007. doi:10.1109/TKDE.2007.34.

104 Armin Haller, Javier D. Fernández, Maulik R. Kamdar, and Axel Polleres. What Are Links in Linked Open Data? A Characterization and Evaluation of Links between Knowledge Graphs on the Web. *Journal of Data and Information Quality*, 12(2):1–34, jun 2020. doi:10.1145/3369875.

105 Armin Haller, Axel Polleres, Daniil Dobriy, Nicolas Ferranti, and Sergio J. Rodríguez Méndez. An analysis of links in Wikidata. In *19th European Semantic Web Conference, ESWC 2022*. Springer, may 2022. doi:10.1007/978-3-031-06981-9_2.

106 Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. Knowledge Transfer for Out-of-Knowledge-Base Entities : A Graph Neural Network Approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 1802–1808, Melbourne, Australia, aug 2017. International Joint Conferences on Artificial Intelligence Organization. doi:10.24963/IJCAI.2017/250.

107 Olaf Hartig, Pierre-Antoine Champin, Gregg Kellogg, and Andy Seaborne. RDF-star and SPARQL-star. Technical report, W3C RDF-DEV Community Group, dec 2021. URL: https://w3c.github.io/rdf-star/cg-spec/2021-12-17.html.

108 Michael Hartung, James F. Terwilliger, and Erhard Rahm. Recent Advances in Schema and Ontology Evolution. In *Schema Matching and Mapping*, pages 149–190. Springer, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-16518-4_6.

109 Daniel Hernández, Aidan Hogan, and Markus Krötzsch. Reifying RDF: What Works Well With Wikidata? In *Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2015)*, volume 1457 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015. URL: https://ceur-ws.org/Vol-1457/SSWS2015_paper3.pdf.

110 Thomas Hoch, Bernhard Heinzl, Gerald Czech, Maqbool Khan, Philipp Waibel, Stefan Bachhofner, Elmar Kiesling, and Bernhard Moser. Teaming.ai: Enabling human-ai teaming intelligence in manufacturing. In Martin Zelm, Andrés Boza, Ramona Diana León, and Raúl Rodríguez-Rodríguez, editors, *Proceedings of Interoperability for Enterprise Systems and Applications Workshops co-located with 11th International Conference on Interoperability for Enterprise Systems and Applications (I-ESA 2022), Valencia, Spain, March 23-25, 2022*, volume 3214 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3214/WS5Paper6.pdf.

111 Marvin Hofer, Sebastian Hellmann, Milan Dojchinovski, and Johannes Frey. The new DBpedia release cycle: Increasing agility and efficiency in knowledge extraction workflows. In Eva Blomqvist, Paul Groth, Victor de Boer, Tassilo Pellegrini, Mehwish Alam, Tobias Käfer, Peter Kieseberg, Sabrina Kirrane, Albert Meroño-Peñuela, and Harshvardhan J. Pandit, editors, *Semantic Systems. In the Era of Knowledge Graphs - 16th International Conference on Semantic Systems, SEMANTiCS 2020, Amsterdam, The Netherlands, September 7-10, 2020, Proceedings*, volume 12378 of *Lecture Notes in Computer Science*

(LNCS), pages 1–18. Springer, 2020. doi:10.1007/978-3-030-59833-4_1.

112 Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge Graphs. *ACM Computing Surveys*, 54(4):71:1–71:37, jul 2021. doi:10.1145/3447772.

113 Aidan Hogan, Antoine Zimmermann, Jürgen Umbrich, Axel Polleres, and Stefan Decker. Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora. *Journal of Web Semantics (JWS)*, 10:76–110, jan 2012. doi:10.1016/J.WEBSEM.2011.11.002.

114 Pan Hu, Boris Motik, and Ian Horrocks. Modular materialisation of datalog programs. *Artificial Intelligence*, 308:103726, 2022. doi:10.1016/J.ARTINT.2022.103726.

115 Zhisheng Huang and Heiner Stuckenschmidt. Reasoning with multi-version ontologies: A temporal logic approach. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings*, volume 3729 of *Lecture Notes in Computer Science (LNCS)*, pages 398–412. Springer, 2005. doi:10.1007/11574620_30.

116 Dilshod Ibragimov, Katja Hose, Torben Bach Pedersen, and Esteban Zimányi. Towards exploratory OLAP over linked open data - A case study. In Malú Castellanos, Umeshwar Dayal, Torben Bach Pedersen, and Nesime Tatbul, editors, *Enabling Real-Time Business Intelligence - International Workshops, BIRTE 2013, Riva del Garda, Italy, August 26, 2013, and BIRTE 2014, Hangzhou, China, September 1, 2014, Revised Selected Papers*, volume 206 of *Lecture Notes in Business Information Processing*, pages 114–132. Springer, 2014. doi:10.1007/978-3-662-46839-5_8.

117 Ihab F. Ilyas, Theodoros Rekatsinas, Vishnu Konda, Jeffrey Pound, Xiaoguang Qi, and Mohamed A. Soliman. Saga: A platform for continuous construction and serving of knowledge at scale. In Zachary G. Ives, Angela Bonifati, and Amr El Abbadi, editors, *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, pages 2259–2272. ACM, 2022. doi:10.1145/3514221.3526049.

118 Robert Isele, Jürgen Umbrich, Christian Bizer, and Andreas Harth. LDspider: An Open-source Crawling Framework for the Web of Linked Data. In Axel Polleres and Huajun Chen, editors, *Proceedings of the ISWC 2010 Posters & Demonstrations Track: Collected Abstracts, Shanghai, China, November 9, 2010*, volume 658 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010. URL: https://ceur-ws.org/Vol-658/paper495.pdf.

119 Ousmane Issa, Angela Bonifati, and Farouk Toumani. Evaluating top-k queries with inconsistency degrees. *Proceedings of the VLDB Endowment*, 13(11):2146–2158, 2020. doi:10.14778/3407790.3407815.

120 Ousmane Issa, Angela Bonifati, and Farouk Toumani. INCA: inconsistency-aware data profiling and querying. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*, pages 2745–2749, 2021. doi:10.1145/3448016.3452760.

121 Rebecca Jackson, Nicolas Matentzoglu, James A Overton, Randi Vita, James P Balhoff, Pier Luigi Buttigieg, Seth Carbon, Melanie Courtot, Alexander D Diehl, Damion M Dooley, William D Duncan, Nomi L Harris, Melissa A Haendel, Suzanna E Lewis, Darren A Natale, David Osumi-Sutherland, Alan Ruttenberg, Lynn M Schriml, Barry Smith, Christian J Stoeckert Jr., Nicole A Vasilevsky, Ramona L Walls, Jie Zheng, Christopher J Mungall, and Bjoern Peters. OBO Foundry in 2021: operationalizing open data principles to evaluate ontologies. *Database Journal on Biological Databases Curation*, 2021, 2021. doi:10.1093/DATABASE/BAAB069.

122 Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, feb 2022. doi:10.1109/TNNLS.2021.3070843.

123 Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. TEQUILA: Temporal question answering over knowledge bases. In Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang, editors, *Proceedings of the 27th ACM international conference on information and knowledge management, CIKM 2018, torino, italy, october 22-26, 2018*, pages 1807–1810. ACM, 2018. doi:10.1145/3269206.3269247.

124 Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. Complex temporal question answering on knowledge graphs. In Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong, editors, *CIKM '21: The 30th ACM international conference on information and knowledge management, virtual event, queensland, australia, november 1 - 5, 2021*, pages 792–802. ACM, 2021. doi:10.1145/3459637.3482416.

125 Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. Encoding temporal information for time-aware link prediction. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2350–2354. The Association for Computational Linguistics, 2016. doi:10.18653/V1/D16-1260.

126 Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga Llavori. Supporting concurrent ontology development: Framework, algorithms and tool. *Data & Knowledge Engineering*, 70(1):146–164, 2011. doi:10.1016/J.DATAK.2010.10.001.

127 Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 conference on empirical methods in natural language processing, EMNLP 2020, online, november 16-20, 2020*, pages 6669–6683. Association for Computational Linguistics, 2020. doi:10.18653/V1/2020.EMNLP-MAIN.541.

128 M. Karsai, M. Kivelä, R. K. Pan, K. Kaski, J. Kertész, A.-L. Barabási, and J. Saramäki. Small but slow world: How network topology and burstiness slow down spreading. *Physical Review E*, 83(2):025102, feb 2011. doi:10.1103/PhysRevE.83.025102.

129 Abbas Keshavarzi and Krys J. Kochut. Kgdiff: Tracking the evolution of knowledge graphs. In *21st International Conference on Information Reuse and Integration for Data Science, IRI 2020, Las Vegas, NV, USA, August 11-13, 2020*, pages 279–286. IEEE, 2020. doi:10.1109/IRI49571.2020.00047.

130 Evgeny Kharlamov, Sebastian Brandt, Martin Giese, Ernesto Jiménez-Ruiz, Steffen Lamparter, Christian Neuenstadt, Özgür Lütfü Özçep, Christoph Pinkel, Ahmet Soylu, Dmitriy Zheleznyakov, Mikhail Roshchin, Stuart Watson, and Ian Horrocks. Semantic access to siemens streaming data: the optique way. In *Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015*, volume 1486 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015. URL: https://ceur-ws.org/Vol-1486/paper_23.pdf.

131 Evgeny Kharlamov, Bernardo Cuenca Grau, Ernesto Jiménez-Ruiz, Steffen Lamparter, Gulnar Mehdi, Martin Ringsquandl, Yavor Nenov, Stephan Grimm, Mikhail Roshchin, and Ian Horrocks. Capturing Industrial Information Models with Ontologies and Constraints. In *15th International Semantic Web Conference*, pages 325–343, 2016. doi:10.1007/978-3-319-46547-0_30.

132 Shahrzad Khayatbashi, Sebastián Ferrada, and Olaf Hartig. Converting property graphs to RDF: a preliminary study of the practical impact of different mappings. In *Proceedings of the 5th ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, pages 1–9, Philadelphia Pennsylvania, jun 2022. ACM. doi:10.1145/3534540.3534695.

133 Udayan Khurana and Amol Deshpande. Storing and Analyzing Historical Graph Data at Scale. In Evaggelia Pitoura, Sofian Maabout, Georgia Koutrika, Amélie Marian, Letizia Tanca, Ioana Manolescu, and Kostas Stefanidis, editors, *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016*, pages 65–76. OpenProceedings.org, 2016. doi:10.5441/002/EDBT.2016.09.

134 Jon M. Kleinberg. Navigation in a small world. *Nature*, 406(6798):845–845, aug 2000. doi:10.1038/35022643.

135 Holger Knublauch and Dimitris Kontokostas. Shapes constraint language (shacl). Technical report, The World Wide Web Consortium (W3C), jul 2017. URL: https://www.w3.org/TR/2017/REC-shacl-20170720/.

136 Boris Konev, Dirk Walther, and Frank Wolter. The Logical Difference Problem for Description Logic Terminologies. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, volume 5195 of *Lecture Notes in Computer Science (LNCS)*, pages 259–274. Springer, 2008. doi:10.1007/978-3-540-71070-7_21.

137 Markus Krötzsch. OWL 2 Profiles: An Introduction to Lightweight Ontology Languages. In Thomas Eiter and Thomas Krennwallner, editors, *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings*, volume 7487 of *Lecture Notes in Computer Science (LNCS)*, pages 112–183. Springer, 2012. doi:10.1007/978-3-642-33158-9_4.

138 Markus Krötzsch, Denny Vrandecic, and Max Völkel. Semantic MediaWiki. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, volume 4273 of *Lecture Notes in Computer Science (LNCS)*, pages 935–942. Springer, 2006. doi:10.1007/11926078_68.

139 Tobias Käfer, Ahmed Abdelrahman, Jürgen Umbrich, Patrick O'Byrne, and Aidan Hogan. Observing Linked Data Dynamics. In Philipp Cimiano, Óscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, volume 7882 of *Lecture Notes in Computer Science (LNCS)*, pages 213–227. Springer, 2013. doi:10.1007/978-3-642-38288-8_15.

140 Tobias Käfer, Jürgen Umbrich, Aidan Hogan, and Axel Polleres. DyLDO: Towards a Dynamic Linked Data Observatory. In Christian Bizer, Tom Heath, Tim Berners-Lee, and Michael Hausenblas, editors, *WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012*, volume 937 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012. URL: https://ceur-ws.org/Vol-937/ldow2012-paper-14.pdf.

141 Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. Tensor decompositions for temporal knowledge base completion. In *8th international conference on learning representations, ICLR 2020, addis ababa, ethiopia, april 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=rke2P1BFwS.

142 Birger Lantow and Kurt Sandkuhl. An analysis of applicability using quality metrics for ontologies on ontology design patterns. *Intelligent Systems in Accounting, Finance and Management*, 22(1):81–99, 2015. doi:10.1002/ISAF.1360.

143 Andri Lareida, Romana Pernischova, Bruno Bastos Rodrigues, and Burkhard Stiller. Abstracting .torrent content consumption into two-mode graphs and their projection to content networks (ConNet). In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 151–159, may 2017. doi:10.23919/INM.2017.7987275.

144 Ora Lassila, Michael Schmidt, Olaf Hartig, Brad Bebee, Dave Bechberger, Willem Broekema, Ankesh Khandelwal, Kelvin Lawrence, Carlos Manuel Lopez Enriquez, Ronak Sharda, and Bryan Thompson. The OneGraph vision: Challenges of breaking the graph model lock-in. *Semantic Web*, 14(1):125–134, nov 2022. doi:10.3233/SW-223273.

145 Julien Leblay, Melisachew Wudage Chekol, and Xin Liu. Towards Temporal Knowledge Graph Embeddings with Arbitrary Time Precision. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pages 685–694, New York, NY, USA, oct 2020. Association for Computing Machinery. doi:10.1145/3340531.3412028.

146 Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and others. Dbpedia – a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015. doi:10.3233/SW-140134.

147 Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett, editors, *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, pages 177–187. ACM, 2005. doi:10.1145/1081870.1081893.

148 Siyuan Liao, Shangsong Liang, Zaiqiao Meng, and Qiang Zhang. Learning Dynamic Embeddings for Temporal Knowledge Graphs. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, pages 535–543, New York, NY, USA, mar 2021. Association for Computing Machinery. doi:10.1145/3437963.3441741.

149 M. Lissandrini, D. Mottin, K. Hose, and T. B. Pedersen. Knowledge graph exploration systems: Are we lost? In *Proc. of the 12th Conference on Innovative Data Systems Research, CIDR 2022*, volume 22, pages 10–13, 2022. URL: https://www.cidrdb.org/cidr2022/papers/p40-lissandrini.pdf.

150 Matteo Lissandrini, Davide Mottin, Themis Palpanas, and Yannis Velegrakis. *Data Exploration Using Example-Based Methods*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2018. doi:10.2200/S00881ED1V01Y201810DTM053.

151 Yu Liu, Wen Hua, Kexuan Xin, and Xiaofang Zhou. Context-Aware Temporal Knowledge Graph Embedding. In Reynold Cheng, Nikos Mamoulis, Yizhou Sun, and Xin Huang, editors, *Web Information Systems Engineering – WISE 2019*, Lecture Notes in Computer Science (LNCS), pages 583–598, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-34223-4_37.

152 Freddy Lécué, Simone Tallevi-Diotallevi, Jer Hayes, Robert Tucker, Veli Bicer, Marco Luca Sbodio, and Pierpaolo Tommasi. Smart traffic analytics in the semantic web with STAR-CITY: Scenarios, system and lessons learned in Dublin City. *Journal of Web Semantics*, 27-28:26–33, 2014. doi:10.1016/J.WEBSEM.2014.07.002.

153 Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *7th biennial conference on innovative data systems research*. CIDR Conference, 2014. URL: http://cidrdb.org/cidr2015/Papers/CIDR15_Paper1.pdf.

154 Sofía Maiolo, Lorena Etcheverry, and Adriana Marotta. Data profiling in property graph databases. *ACM J. Data Inf. Qual.*, 12(4):20:1–20:27, 2020. doi:10.1145/3409473.

155 Alessandro Margara, Gianpaolo Cugola, Dario Collavini, and Daniele Dell'Aglio. Efficient temporal reasoning on streams of events with DOTR. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science (LNCS)*, pages 384–399. Springer, 2018. doi:10.1007/978-3-319-93417-4_25.

156 Cynthia Matuszek, John Cabral, Michael Witbrock, and John DeOliveira. An introduction to the syntax and content of cyc. In *Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, Papers from the 2006 AAAI Spring Symposium, Technical Report SS-06-05, Stanford, California, USA, March 27-29, 2006*, pages 44–49. AAAI, 2006. URL: http://www.aaai.org/Library/Symposia/Spring/2006/ss06-05-007.php.

157 Christian Meilicke. *Alignment incoherence in ontology matching*. PhD Thesis, University of Mannheim, 2011. URL: https://ub-madoc.bib.uni-mannheim.de/29351.

158 Albert Meroño-Peñuela, Efstratios Kontopoulos, Sándor Darányi, and Yiannis Kompatsiaris. A study of intensional concept drift in trending DBpedia concepts. In *SEMANTICS workshops*, volume 2063 of *CEUR workshop proceedings*. CEUR-WS.org, 2017. URL: https://ceur-ws.org/Vol-2063/dal-paper3.pdf.

159 Robert Meusel, Petar Petrovski, and Christian Bizer. The WebDataCommons Microdata, RDFa and Microformat Dataset Series. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig A. Knoblock, Denny Vrandecic, Paul Groth, Natasha F. Noy, Krzysztof Janowicz, and Carole A. Goble, editors, *The Semantic Web - ISWC 2014 -*

13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I, volume 8796 of *Lecture Notes in Computer Science (LNCS)*, pages 277–292. Springer, 2014. doi:10.1007/978-3-319-11964-9_18.

160 Pasquale Minervini, Erik Arakelyan, Daniel Daza, and Michael Cochez. Complex query answering with neural link predictors (extended abstract). In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5309–5313. ijcai.org, 2022. doi:10.24963/IJCAI.2022/741.

161 Vera Zaychik Moffitt and Julia Stoyanovich. Temporal graph algebra. In Tiark Rompf and Alexander Alexandrov, editors, *Proceedings of The 16th International Symposium on Database Programming Languages, DBPL 2017, Munich, Germany, September 1, 2017*, pages 10:1–10:12. ACM, 2017. doi:10.1145/3122831.3122838.

162 Boris Motik, Yavor Nenov, Robert Piro, Ian Horrocks, and Dan Olteanu. Parallel Materialisation of Datalog Programs in Centralised, Main-Memory RDF Systems. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 129–137. AAAI Press, 2014. doi:10.1609/AAAI.V28I1.8730.

163 Boris Motik, Yavor Nenov, Robert Edgar Felix Piro, and Ian Horrocks. Incremental Update of Datalog Materialisation: the Backward/Forward Algorithm. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 1560–1568. AAAI Press, 2015. doi:10.1609/AAAI.V29I1.9409.

164 Yavor Nenov, Robert Piro, Boris Motik, Ian Horrocks, Zhe Wu, and Jay Banerjee. RDFox: A Highly-Scalable RDF Store. In Marcelo Arenas, Óscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d'Aquin, Kavitha Srinivas, Paul Groth, Michel Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, and Steffen Staab, editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*, volume 9367 of *Lecture Notes in Computer Science (LNCS)*, pages 3–20. Springer, 2015. doi:10.1007/978-3-319-25010-6_1.

165 Mark Newman, Albert-László Barabási, and Duncan J. Watts. *The Structure and Dynamics of Networks*. Princeton University Press, dec 2011. doi:10.1515/9781400841356.

166 Chifumi Nishioka and Ansgar Scherp. Temporal Patterns and Periodicity of Entity Dynamics in the Linked Open Data Cloud. In Ken Barker and José Manuél Gómez-Pérez, editors, *Proceedings of the 8th International Conference on Knowledge Capture, K-CAP 2015, Palisades, NY, USA, October 7-10, 2015*, pages 22:1–22:4. ACM, 2015. doi:10.1145/2815833.2816948.

167 Chifumi Nishioka and Ansgar Scherp. Information-theoretic Analysis of Entity Dynamics on the Linked Open Data Cloud. In Elena Demidova, Stefan Dietze, Julian Szymanski, and John G. Breslin, editors, *Proceedings of the 3rd International Workshop on Dataset PROFIling and fEderated Search for Linked Data (PROFILES '16) co-located with the 13th ESWC 2016 Conference, Anissaras, Greece, May 30, 2016*, volume 1597 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016. URL: https://ceur-ws.org/Vol-1597/PROFILES2016_paper2.pdf.

168 Chifumi Nishioka and Ansgar Scherp. Keeping linked open data caches up-to-date by predicting the life-time of RDF triples. In Amit P. Sheth, Axel Ngonga, Yin Wang, Elizabeth Chang, Dominik Slezak, Bogdan Franczyk, Rainer Alt, Xiaohui Tao, and Rainer Unland, editors, *Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, August 23-26, 2017*, pages 73–80. ACM, 2017. doi:10.1145/3106426.3106463.

169 Chifumi Nishioka and Ansgar Scherp. Analysing the Evolution of Knowledge Graphs for the Purpose of Change Verification. In *12th IEEE International Conference on Semantic Computing, ICSC 2018, Laguna Hills, CA, USA, January 31 - February 2, 2018*, pages 25–32. IEEE Computer Society, 2018. doi:10.1109/ICSC.2018.00013.

170 Natasha F. Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. Industry-scale knowledge graphs: Lessons and challenges. *ACM Queue*, 17(2):20, 2019. doi:10.1145/3329781.3332266.

171 Fabrizio Orlandi, Damien Graux, and Declan O'Sullivan. Benchmarking RDF Metadata Representations: Reification, Singleton Property and RDF. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 233–240, jan 2021. ISSN: 2325-6516. doi:10.1109/ICSC50631.2021.00049.

172 Anthony M Orme, Haining Yao, and Letha H Etzkorn. Indicating ontology data quality, stability, and completeness throughout ontology evolution. *Journal of Software Maintenance and Evolution: Research and Practice*, 19(1):49–75, 2007. doi:10.1002/SMR.341.

173 Kristian Otte, Kristian Simoni Vestermark, Huan Li, and Daniele Dell'Aglio. Towards A question answering system over temporal knowledge graph embedding. In Mehwish Alam and Michael Cochez, editors, *Proceedings of the workshop on deep learning for knowledge graphs (DL4KG 2022) co-located with the 21th international semantic web conference (ISWC 2022), virtual conference, online, october 24, 2022*, volume 3342 of *CEUR workshop proceedings*. CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3342/paper-1.pdf.

174 Anil Pacaci, Angela Bonifati, and M. Tamer Özsu. Regular path query evaluation on streaming graphs. In David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo, editors, *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pages 1415–1430. ACM, 2020. doi:10.1145/3318464.3389733.

175 Anil Pacaci, Angela Bonifati, and M. Tamer Özsu. Evaluating complex queries on streaming graphs. In *38th IEEE International Conference on Data*

Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022, pages 272–285. IEEE, 2022. doi:10.1109/ICDE53745.2022.00025.

176 Heiko Paulheim and Aldo Gangemi. Serving dbpedia with DOLCE - more than just adding a cherry on top. In Marcelo Arenas, Óscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d'Aquin, Kavitha Srinivas, Paul Groth, Michel Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, and Steffen Staab, editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, volume 9366 of *Lecture Notes in Computer Science (LNCS)*, pages 180–196. Springer, 2015. doi:10.1007/978-3-319-25007-6_11.

177 Olivier Pelgrin, Luis Galárraga, and Katja Hose. Towards fully-fledged archiving for RDF datasets. *Semantic Web Journal*, 12(6):903–925, 2021. doi:10.3233/SW-210434.

178 Olivier Pelgrin, Ruben Taelman, Luis Galárraga, and Katja Hose. GLENDA: Querying RDF Archives with full SPARQL. In *Proceedings of the 25th Extended Semantic Web Conference (ESWC)*, 2023. doi:10.1007/978-3-031-43458-7_14.

179 Olivier Pelgrin, Ruben Taelman, Luis Galárraga, and Katja Hose. Scaling Large RDF Archives To Very Long Histories. In *2023 IEEE 17th International Conference on Semantic Computing (ICSC)*, pages 41–48, Laguna Hills, CA, USA, feb 2023. IEEE. doi:10.1109/ICSC56153.2023.00013.

180 Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. YAGO 4: A Reason-able Knowledge Base. In Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez, editors, *The Semantic Web*, pages 583–596. Springer, 2020. doi:10.1007/978-3-030-49461-2_34.

181 Romana Pernisch, Daniele Dell'Aglio, and Abraham Bernstein. Toward Measuring the Resemblance of Embedding Models for Evolving Ontologies. In *Proceedings of the 11th on Knowledge Capture Conference*, pages 177–184, Virtual Event USA, dec 2021. ACM. doi:10.1145/3460210.3493540.

182 Romana Pernisch, Daniele Dell'Aglio, and Abraham Bernstein. Beware of the hierarchy — An analysis of ontology evolution and the materialisation impact for biomedical ontologies. *Journal of Web Semantics*, 70:100658, jul 2021. doi:10.1016/J.WEBSEM.2021.100658.

183 Romana Pernisch, Daniele Dell'Aglio, Mirko Serbak, Rafael S. Gonçalves, and Abraham Bernstein. Visualising the effects of ontology changes and studying their understanding with ChImp. *Journal of Web Semantics*, 74:100715, oct 2022. doi:10.1016/J.WEBSEM.2022.100715.

184 Romana Pernischová, Florian Ruosch, Daniele Dell'Aglio, and Abraham Bernstein. Stream processing: The matrix revolutions. In Thorsten Liebig, Achille Fokoue, and Zhe Wu, editors, *Proceedings of the 12th International Workshop on Scalable Semantic Web Knowledge Base Systems co-located with 17th International Semantic Web Conference, SSWS@ISWC 2018, Monterey, California, USA, October 9, 2018*, volume 2179 of *CEUR Workshop Proceedings*, pages 15–27. CEUR-WS.org, 2018. URL: https://ceur-ws.org/Vol-2179/SSWS2018_paper2.pdf.

185 Matthew Perry, Prateek Jain, and Amit P. Sheth. SPARQL-ST: Extending SPARQL to Support Spatiotemporal Queries. In Naveen Ashish and Amit P. Sheth, editors, *Geospatial Semantics and the Semantic Web*, volume 12, pages 61–86, Boston, MA, 2011. Springer US. doi:10.1007/978-1-4419-9446-2_3.

186 Danh Le Phuoc, Minh Dao-Tran, Josiane Xavier Parreira, and Manfred Hauswirth. A native and adaptive approach for unified processing of linked streams and linked data. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Fridman Noy, and Eva Blomqvist, editors, *The semantic web - ISWC 2011 - 10th international semantic web conference, bonn, germany, october 23-27, 2011, proceedings, part I*, volume 7031 of *Lecture Notes in Computer Science (LNCS)*, pages 370–388. Springer, 2011. doi:10.1007/978-3-642-25073-6_24.

187 Danh Le Phuoc, Thomas Eiter, and Anh Lê Tuán. A scalable reasoning and learning approach for neural-symbolic stream fusion. In *AAAI*, pages 4996–5005. AAAI Press, 2021. doi:10.1609/AAAI.V35I6.16633.

188 Alessandro Piscopo, Lucie-Aimée Kaffee, Chris Phethean, and Elena Simperl. Provenance information in a collaborative knowledge graph: An evaluation of wikidata external references. In Claudia d'Amato, Miriam Fernández, Valentina A. M. Tamma, Freddy Lécué, Philippe Cudré-Mauroux, Juan F. Sequeda, Christoph Lange, and Jeff Heflin, editors, *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, volume 10587 of *Lecture Notes in Computer Science (LNCS)*, pages 542–558. Springer, 2017. doi:10.1007/978-3-319-68288-4_32.

189 Alessandro Piscopo, Chris Phethean, and Elena Simperl. What makes a good collaborative knowledge graph: Group composition and quality in wikidata. In Giovanni Luca Ciampaglia, Afra J. Mashhadi, and Taha Yasseri, editors, *Social Informatics - 9th International Conference, SocInfo 2017, Oxford, UK, September 13-15, 2017, Proceedings, Part I*, volume 10539 of *Lecture Notes in Computer Science (LNCS)*, pages 305–322. Springer, 2017. doi:10.1007/978-3-319-67217-5_19.

190 Alessandro Piscopo and Elena Simperl. Who models the world?: Collaborative ontology creation and user roles in wikidata. *Proc. ACM Hum. Comput. Interact.*, 2(CSCW):141:1–141:18, 2018. doi:10.1145/3274410.

191 Alessandro Piscopo and Elena Simperl. Who Models the World?: Collaborative Ontology Creation and User Roles in Wikidata. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–18, nov 2018. doi:10.1145/3274410.

192 Alessandro Piscopo and Elena Simperl. What we talk about when we talk about wikidata quality: a literature survey. In Björn Lundell, Jonas Gamalielsson, Lorraine Morgan, and

Gregorio Robles, editors, *Proceedings of the 15th International Symposium on Open Collaboration, OpenSym 2019, Skövde, Sweden, August 20-22, 2019*, pages 17:1–17:11. ACM, 2019. doi:10.1145/3306446.3340822.

193 Mason A. Porter, Jukka-Pekka Onnela, and Peter J. Mucha. Communities in Networks. *Computing Research Repository (CoRR)*, abs/0902.3788, 2009. arXiv:0902.3788.

194 Giulia Preti, Matteo Lissandrini, Davide Mottin, and Yannis Velegrakis. Mining patterns in graphs with multiple weights. *Distributed Parallel Databases*, 39(2):281–319, 2021. doi:10.1007/S10619-019-07259-W.

195 Eric Prud'hommeaux, Iovka Boneva, Jose Emilio Labra Gayo, and Gregg Kellogg. Shape expressions language 2.1. Technical report, W3C Shape Expressions Community Group, oct 2019. URL: http://shex.io/shex-semantics/.

196 Kashif Rabbani, Matteo Lissandrini, and Katja Hose. Extraction of validating shapes from very large knowledge graphs. *Proceedings of the VLDB Endowment*, 16(5):1023–1032, 2023. doi:10.14778/3579075.3579078.

197 Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001. doi:10.1007/S007780100057.

198 Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II 15*, pages 177–185. Springer, 2016. doi:10.1007/978-3-319-46547-0_19.

199 Yuan Ren and Jeff Z. Pan. Optimising ontology stream reasoning with truth maintenance system. In Craig Macdonald, Iadh Ounis, and Ian Ruthven, editors, *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 831–836. ACM, 2011. doi:10.1145/2063576.2063696.

200 Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. *ACM Transactions on Knowledge Discovery from Data*, 15(2):14:1–14:49, jan 2021. doi:10.1145/3424672.

201 Christopher Rost, Kevin Gómez, Matthias Täschner, Philip Fritzsche, Lucas Schons, Lukas Christ, Timo Adameit, Martin Junghanns, and Erhard Rahm. Distributed temporal graph analytics with GRADOOP. *VLDB Journal*, 31(2):375–401, 2022. doi:10.1007/S00778-021-00667-4.

202 Anisa Rula, Amrapali Zaveri, Elena Simperl, and Elena Demidova. Editorial: Special issue on quality assessment of knowledge graphs dedicated to the memory of amrapali zaveri. *Journal of Data and Information Quality*, 12(2), may 2020. doi:10.1145/3388748.

203 Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. ChronoR: Rotation Based Temporal Knowledge Graph Embedding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6471–6479, may 2021. Number: 7. doi:10.1609/AAAI.V35I7.16802.

204 Christian Sageder. Annotating entities with finegrained types in austrian court decisions. In *Further with Knowledge Graphs - Proceedings of the 17th International Conference on Semantic Systems, SEMANTiCS 2017, Amsterdam, The Netherlands, September 6-9, 2021*, volume 53 of *Studies on the Semantic Web*, pages 139–153. IOS Press, 2021. doi:10.3233/SSW210041.

205 Satya S. Sahoo, Christopher Thomas, Amit Sheth, William S. York, and Samir Tartir. Knowledge modeling and its application in life sciences: A tale of two ontologies. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 317–326, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1135777.1135826.

206 Fabio Saracco, Riccardo Di Clemente, Andrea Gabrielli, and Tiziano Squartini. Randomizing bipartite networks: the case of the World Trade Web. *Scientific Reports*, 5(1):10595, jun 2015. doi:10.1038/srep10595.

207 Cristina Sarasua, Alessandro Checco, Gianluca Demartini, Djellel Eddine Difallah, Michael Feldman, and Lydia Pintscher. The evolution of power and standard wikidata editors: Comparing editing behavior over time to predict lifespan and volume of edits. *Computer Supported Cooperative Work (CSCW)*, 28(5):843–882, 2019. doi:10.1007/S10606-018-9344-Y.

208 Lukas Schmelzeisen, Corina Dima, and Steffen Staab. Wikidated 1.0: An evolving knowledge graph dataset of wikidata's revision history. In Lucie-Aimée Kaffee, Simon Razniewski, and Aidan Hogan, editors, *Proceedings of the 2nd Wikidata Workshop (Wikidata 2021) co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual Conference, October 24, 2021*, volume 2982 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021. URL: https://ceur-ws.org/Vol-2982/paper-11.pdf.

209 Stefan Schmid, Cory Henson, and Tuan Tran. Using knowledge graphs to search an enterprise data lake. In Pascal Hitzler, Sabrina Kirrane, Olaf Hartig, Victor de Boer, Maria-Esther Vidal, Maria Maleshkova, Stefan Schlobach, Karl Hammar, Nelia Lasierra, Steffen Stadtmüller, Katja Hose, and Ruben Verborgh, editors, *The Semantic Web: ESWC 2019 Satellite Events - ESWC 2019 Satellite Events, Portorož, Slovenia, June 2-6, 2019, Revised Selected Papers*, volume 11762 of *Lecture Notes in Computer Science (LNCS)*, pages 262–266. Springer, 2019. doi:10.1007/978-3-030-32327-1_46.

210 Patrik Schneider, Daniel Alvarez-Coello, Anh Le-Tuan, Manh Nguyen Duc, and Danh Le Phuoc. Stream reasoning playground. In Paul Groth, Maria-Esther Vidal, Fabian M. Suchanek, Pedro A. Szekely, Pavan Kapanipathi, Catia Pesquita, Hala Skaf-Molli, and Minna Tamper, editors, *The Semantic Web - 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29 - June 2, 2022, Proceedings*, volume 13261 of *Lecture Notes in Computer Science (LNCS)*, pages 406–424. Springer, 2022. doi:10.1007/978-3-031-06981-9_24.

211 Vedran Sekara, Arkadiusz Stopczynski, and Sune Lehmann. Fundamental structures of dynamic social networks. *Proceedings of the national academy of sciences*, 113(36):9977–9982, 2016. doi:10.1073/pnas.1602803113.

212 Tong Shen, Fu Zhang, and Jingwei Cheng. A comprehensive overview of knowledge graph completion. *Knowledge-Based Systems*, 255:109597, nov 2022. doi:10.1016/J.KNOSYS.2022.109597.

213 Ying Shen, Daoyuan Chen, Buzhou Tang, Min Yang, and Kai Lei. EAPB: entropy-aware path-based metric for ontology quality. *Journal of Biomedical Semantics*, 9(1):20, 2018. doi:10.1186/s13326-018-0188-7.

214 Robert H Shumway, David S Stoffer, and David S Stoffer. *Time series analysis and its applications*, volume 3. Springer, 2000. doi:10.1007/978-3-319-52452-8.

215 Philipp Singer, Denis Helic, Behnam Taraghi, and Markus Strohmaier. Detecting Memory and Structure in Human Navigation Patterns Using Markov Chain Models of Varying Order. *PLoS ONE*, 9(7):e102070, jul 2014. doi:10.1371/journal.pone.0102070.

216 Alessandro Solimando, Ernesto Jiménez-Ruiz, and Giovanna Guerrini. Minimizing conservativity violations in ontology alignments: algorithms and evaluation. *Knowledge and Information Systems (KAIS)*, 51(3):775–819, 2017. doi:10.1007/S10115-016-0983-3.

217 Hyun-Je Song and Seong-Bae Park. Enriching Translation-Based Knowledge Graph Embeddings Through Continual Learning. *IEEE Access*, 6:60489–60497, 2018. Conference Name: IEEE Access. doi:10.1109/ACCESS.2018.2874656.

218 T. G. Stavropoulos, S. Andreadis, E. Kontopoulos, and I. Kompatsiaris. SemaDrift: A hybrid method and visual tools to measure semantic drift in ontologies. *Journal of Web Semantics*, jun 2018. doi:10.1016/j.websem.2018.05.001.

219 Thanos G. Stavropoulos, Efstratios Kontopoulos, Albert Meroño-Peñuela, Stavros Tachos, Stelios Andreadis, and Yiannis Kompatsiaris. Cross-domain semantic drift measurement in ontologies using the semadrift tool and metrics. In Jeremy Debattista, Jürgen Umbrich, Javier D. Fernández, Anisa Rula, Amrapali Zaveri, Anastasia Dimou, and Wouter Beek, editors, *Joint proceedings of the 3rd Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW 2017) and the 4th Workshop on Linked Data Quality (LDQ 2017) co-located with 14th European Semantic Web Conference (ESWC 2017), Portorož, Slovenia, May 28th-29th, 2017*, volume 1824 of *CEUR Workshop Proceedings*, pages 59–72. CEUR-WS.org, 2017. URL: https://ceur-ws.org/Vol-1824/mepdaw_paper_5.pdf.

220 Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. Association for Computing Machinery. doi:10.1145/1242572.1242667.

221 Xibo Sun, Shixuan Sun, Qiong Luo, and Bingsheng He. An in-depth study of continuous subgraph matching. *Proceedings of the VLDB Endowment*, 15(7):1403–1416, 2022. doi:10.14778/3523210.3523218.

222 Ruben Taelman, Thibault Mahieu, Martin Vanbrabant, and Ruben Verborgh. Optimizing storage of RDF archives using bidirectional delta chains. *Semantic Web*, 13(4):705–734, may 2022. doi:10.3233/SW-210449.

223 Ruben Taelman, Miel Vander Sande, Joachim Van Herwegen, Erik Mannens, and Ruben Verborgh. Triple storage for random-access versioned querying of RDF archives. *Journal of Web Semantics*, 54:4–28, jan 2019. doi:10.1016/J.WEBSEM.2018.08.001.

224 Thomas Pellissier Tanon, Camille Bourgaux, and Fabian M. Suchanek. Learning how to correct a knowledge base from the edit history. In Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia, editors, *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1465–1475. ACM, 2019. doi:10.1145/3308558.3313584.

225 Thomas Pellissier Tanon and Fabian M. Suchanek. Querying the edit history of wikidata. In Pascal Hitzler, Sabrina Kirrane, Olaf Hartig, Victor de Boer, Maria-Esther Vidal, Maria Maleshkova, Stefan Schlobach, Karl Hammar, Nelia Lasierra, Steffen Stadtmüller, Katja Hose, and Ruben Verborgh, editors, *The Semantic Web: ESWC 2019 Satellite Events - ESWC 2019 Satellite Events, Portorož, Slovenia, June 2-6, 2019, Revised Selected Papers*, volume 11762 of *Lecture Notes in Computer Science (LNCS)*, pages 161–166. Springer, 2019. doi:10.1007/978-3-030-32327-1_32.

226 Jonas Tappolet and Abraham Bernstein. Applied temporal RDF: efficient temporal querying of RDF data with SPARQL. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, *The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings*, volume 5554 of *Lecture Notes in Computer Science (LNCS)*, pages 308–322. Springer, 2009. doi:10.1007/978-3-642-02121-3_25.

227 Samir Tartir, Ismailcem Arpinar, and Amit Sheth. *Ontological Evaluation and Validation*, pages 115–130. Springer, sep 2010. doi:10.1007/978-90-481-8847-5_5.

228 Yi Tay, Anh Luu, and Siu Cheung Hui. Non-parametric estimation of multiple embeddings for link prediction on dynamic knowledge graphs. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 31(1), feb 2017. doi:10.1609/aaai.v31i1.10685.

229 Christoph Tempich and Raphael Volz. Towards a benchmark for semantic web reasoners - an analysis of the DAML ontology library. In York Sure and Óscar Corcho, editors, *EON2003, Evaluation*

of Ontology-based Tools, Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools held at the 2nd International Semantic Web Conference ISWC 2003, 20th October 2003 (Workshop day), Sundial Resort, Sanibel Island, Florida, USA, volume 87 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003. URL: https://ceur-ws.org/Vol-87/EON2003_Tempich.pdf.

230 Mattias Tiger and Fredrik Heintz. Stream reasoning using temporal logic and predictive probabilistic state models. In *TIME*, pages 196–205. IEEE Computer Society, 2016. doi:10.1109/TIME.2016.28.

231 Stein L. Tomassen and Darijus Strasunskas. An ontology-driven approach to web search: Analysis of its sensitivity to ontology quality and search tasks. In *Proceedings of the 11th International Conference on Information Integration and Web-Based Applications & Services*, iiWAS '09, pages 130–138, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1806338.1806368.

232 Riccardo Tommasini, Pieter Bonte, Femke Ongenae, and Emanuele Della Valle. RSP4J: an API for RDF stream processing. In Ruben Verborgh, Katja Hose, Heiko Paulheim, Pierre-Antoine Champin, Maria Maleshkova, Óscar Corcho, Petar Ristoski, and Mehwish Alam, editors, *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, volume 12731 of *Lecture Notes in Computer Science (LNCS)*, pages 565–581. Springer, 2021. doi:10.1007/978-3-030-77385-4_34.

233 Riccardo Tommasini, Pieter Bonte, Emanuele Della Valle, Femke Ongenae, and Filip De Turck. A query model for ontology-based event processing over RDF streams. In Catherine Faron-Zucker, Chiara Ghidini, Amedeo Napoli, and Yannick Toussaint, editors, *Knowledge Engineering and Knowledge Management - 21st International Conference, EKAW 2018, Nancy, France, November 12-16, 2018, Proceedings*, volume 11313 of *Lecture Notes in Computer Science (LNCS)*, pages 439–453. Springer, 2018. doi:10.1007/978-3-030-03667-6_28.

234 Riccardo Tommasini, Davide Calvaresi, and Jean-Paul Calbimonte. Stream reasoning agents: Blue sky ideas track. In Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor, editors, *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pages 1664–1680. International Foundation for Autonomous Agents and Multiagent Systems, 2019. URL: http://dl.acm.org/citation.cfm?id=3331894.

235 Riccardo Tommasini, Mohamed Ragab, Alessandro Falcetta, Emanuele Della Valle, and Sherif Sakr. Bootstrapping the publication of linked data streams. In Mari Carmen Suárez-Figueroa, Gong Cheng, Anna Lisa Gentile, Christophe Guéret, C. Maria Keet, and Abraham Bernstein, editors, *Proceedings of the ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26-30, 2019*, volume 2456 of *CEUR Workshop Proceedings*, pages 29–32. CEUR-WS.org, 2019. URL: https://ceur-ws.org/Vol-2456/paper8.pdf.

236 Riccardo Tommasini, Mohamed Ragab, Alessandro Falcetta, Emanuele Della Valle, and Sherif Sakr. A first step towards a streaming linked data life-cycle. In Jeff Z. Pan, Valentina A. M. Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II*, volume 12507 of *Lecture Notes in Computer Science (LNCS)*, pages 634–650. Springer, 2020. doi:10.1007/978-3-030-62466-8_39.

237 Riccardo Tommasini, Sherif Sakr, Emanuele Della Valle, and Hojjat Jafarpour. Declarative languages for big streaming data. In Angela Bonifati, Yongluan Zhou, Marcos Antonio Vaz Salles, Alexander Böhm, Dan Olteanu, George H. L. Fletcher, Arijit Khan, and Bin Yang, editors, *Proceedings of the 23rd International Conference on Extending Database Technology, EDBT 2020, Copenhagen, Denmark, March 30 - April 02, 2020*, pages 643–646. OpenProceedings.org, 2020. doi:10.5441/002/EDBT.2020.84.

238 Efthymia Tsamoura, David Carral, Enrico Malizia, and Jacopo Urbani. Materializing Knowledge Bases via Trigger Graphs. *Proceedings of the VLDB Endowment*, 14(6):943–956, 2021. doi:10.14778/3447689.3447699.

239 Evangelia Tsoukanara, Georgia Koloniari, and Evaggelia Pitoura. Graph-tempo: An aggregation framework for evolving graphs. In *Proceedings 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28–31*, 2023. doi:10.48786/edbt.2023.79.

240 Stella Verkijk, Ritte Roothaert, Romana Pernisch, and Stefan Schlobach. Do you catch my drift? In *Proceedings of the 12th on Knowledge Capture Conference*, Pensecola, FL, USA, dec 2023. ACM. doi:10.1145/3587259.3627555.

241 Raphael Volz, Steffen Staab, and Boris Motik. Incrementally Maintaining Materializations of Ontologies Stored in Logic Databases. *Journal on Data Semantics II*, 2:1–34, 2005. doi:10.1007/978-3-540-30567-5_1.

242 Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014. doi:10.1145/2629489.

243 Przemyslaw Andrzej Walega, Mark Kaminski, Dingmin Wang, and Bernardo Cuenca Grau. Stream reasoning with DatalogMTL. *Journal of Web Semantics*, 76:100776, 2023. doi:10.1016/J.WEBSEM.2023.100776.

244 Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, dec 2017. doi:10.1109/TKDE.2017.2754499.

245 Richard Y Wang, Mostapha Ziad, and Yang W Lee. *Data quality*, volume 23. Springer Science & Business Media, 2006. doi:10.1007/b116303.

246 Shenghui Wang, Stefan Schlobach, and Michel Klein. Concept drift and how to identify it. *Journal of Web Semantics*, 9(3):247–265, sep 2011. `doi:10.1016/J.WEBSEM.2011.05.003`.

247 Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, jun 1998. `doi:10.1038/30918`.

248 Patricia L. Whetzel, Natalya Fridman Noy, Nigam H. Shah, Paul R. Alexander, Csongor Nyulas, Tania Tudorache, and Mark A. Musen. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic Acids Research*, 39(Web-Server-Issue):541–545, jul 2011. `doi:10.1093/NAR/GKR469`.

249 Tianxing Wu, Arijit Khan, Melvin Yong, Guilin Qi, and Meng Wang. Efficiently embedding dynamic knowledge graphs. *Knowledge-Based Systems*, 250:109124, 2022. `doi:10.1016/J.KNOSYS.2022.109124`.

250 Zhe Yang, Dalu Zhang, and Chuan Ye. Evaluation metrics for ontology complexity and evolution analysis. In *2006 IEEE International Conference on e-Business Engineering (ICEBE 2006), 24-26 October 2006, Shanghai, China*, pages 162–170. IEEE Computer Society, 2006. `doi:10.1109/ICEBE.2006.48`.

251 Carlo Zaniolo, Shi Gao, Maurizio Atzori, Muhao Chen, and Jiaqi Gu. User-friendly temporal queries on historical knowledge bases. *Information and Computation*, 259(3):444–459, 2018. `doi:10.1016/J.IC.2017.08.012`.

252 Dalu Zhang, Chuan Ye, and Zhe Yang. An evaluation method for ontology complexity analysis in ontology evolution. In Steffen Staab and Vojtech Svátek, editors, *Managing Knowledge in a World of Networks, 15th International Conference, EKAW 2006, Podebrady, Czech Republic, October 2-6, 2006, Proceedings*, volume 4248 of *Lecture Notes in Computer Science (LNCS)*, pages 214–221. Springer, 2006. `doi:10.1007/11891451_20`.

253 Fu Zhang, Ke Wang, Zhiyin Li, and Jingwei Cheng. Temporal data representation and querying based on RDF. *IEEE Access*, 7:85000–85023, 2019. `doi:10.1109/ACCESS.2019.2924550`.

254 Lei (Nico) Zheng, Christopher M. Albano, Neev M. Vora, Feng Mai, and Jeffrey V. Nickerson. The roles bots play in wikipedia. *Proceedings of the ACM on Human Computer Interaction (HCI)*, 3(CSCW), nov 2019. `doi:10.1145/3359317`.

255 Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural Bellman-Ford networks: A general graph neural network framework for link prediction. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29476–29490. Curran Associates, Inc., 2021. `doi:10.48550/arXiv.2106.06935`.

256 Antoine Zimmermann, Nuno Lopes, Axel Polleres, and Umberto Straccia. A general framework for representing, reasoning and querying with annotated Semantic Web data. *Journal of Web Semantics*, 11:72–95, mar 2012. `doi:10.1016/J.WEBSEM.2011.08.006`.

257 Mussab Zneika, Claudio Lucchese, Dan Vodislav, and Dimitris Kotzinos. Summarizing linked data RDF graphs using approximate graph pattern mining. In Evaggelia Pitoura, Sofian Maabout, Georgia Koutrika, Amélie Marian, Letizia Tanca, Ioana Manolescu, and Kostas Stefanidis, editors, *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016*, pages 684–685. OpenProceedings.org, 2016. `doi:10.5441/002/EDBT.2016.86`.