

DBpedia-TKG: Capturing Wikipedia’s Evolution as Temporal Knowledge Graphs

Marvin Hofer^{1,2}[0000–0003–4667–5743], Maximilian Mario Töpfer^{1,2}, Christopher Rost^{1,2}[0000–0003–4217–9312], and Erhard Rahm^{1,2}[0000–0002–2665–1114]

¹ University of Leipzig, Germany

² ScaDS.AI Dresden/Leipzig, Germany

{hofer,rost,rahm}@informatik.uni-leipzig.de,{id37owan}@studserv.uni-leipzig.de

Abstract. This paper introduces the DBpedia Temporal Knowledge Graph (DBpedia-TKG), an extension of the DBpedia extraction process to generate temporal versions of the knowledge graph. DBpedia has long served as a vital resource in the Semantic Web community and as a primary data source for research, offering structured information extracted from Wikipedia. However, it lacks a temporal dimension that captures the evolving nature of knowledge. Our approach addresses this gap by enabling the creation of temporal graph versions that reflect changes in Wikipedia pages across various revisions sourced from the Wikipedia meta-history dumps. Our implementation runs in a containerized data extraction system, scaling across an eight-node cluster to extract the first version in under three days.

Our setup facilitates the generation of distinct DBpedia temporal graph variants through configurable settings, using different page extractors, temporal filters, and DBpedia ontology versions. In our initial evaluation, we present comprehensive statistics demonstrating the impact of Wikipedia changes on the extracted data and provide insights into the temporal diversity of the knowledge graph. Finally, we discuss the potential benefits of DBpedia Temporal KG for various research domains. The first English version consists of around **1.7 billion** extracted triples between **270 million** different time points.

Resource type: Dataset

License: CC BY-SA 4.0

Dataset DOI: <https://doi.org/10.5281/zenodo.14532571>

Code URL: <https://github.com/dbpedia/dbpedia-temporal>

Keywords: Dataset · Temporal Knowledge Graph · DBpedia.

1 Introduction

Knowledge graphs have emerged as pivotal tools for representing and analyzing structured information across various domains, including natural language processing, information retrieval, and data analytics. Among them, temporal knowledge graphs do not merely offer a static snapshot of facts; instead, they capture the full evolution of information, enabling users to explore how knowledge

changes over time. This richer temporal perspective supports advanced analyses, such as identifying trends, understanding the provenance of facts, and examining the unfolding narrative of entities beyond what static knowledge graphs can provide [23,24].

DBpedia [2] stands out as one of the most widely used knowledge graphs, as it is derived from the structured content of Wikipedia. Despite its extensive adoption, DBpedia, like many static knowledge graphs, is limited to representing a single temporal state of the knowledge it encodes. Current versions of DBpedia do not incorporate temporal information, such as when facts were introduced, modified, or removed. As a result, users miss out on insights related to the historical trajectory of entities, the emergence of new facts, and the context behind evolving information.

Wikipedia, the foundational source of DBpedia, is inherently dynamic. Its articles undergo continuous revisions contributed by a global community of editors. Each revision leaves a transactional trace, a record of when certain facts were added, updated, or deleted. Although these revision timestamps do not necessarily reflect the real-world *validity time* of the information (e.g., when a historical event actually occurred or when a population figure was truly valid), they still provide a crucial dimension, which can be captured as *transaction time* in the form of extracted triple lifespans. This dimension captures when knowledge becomes available, observable, or recognized within Wikipedia itself. By incorporating triple transaction times, we can better understand the editorial and informational processes that shape the knowledge graph, even if the valid times of the underlying facts differ or remain uncertain.

In this paper, we introduce an extended version of the DBpedia dataset that embeds temporal dynamics by extracting every available revision of Wikipedia up to a specified date. We employ a *timestamp-based* temporal graph model to represent triple-level changes between revisions (as *transaction-time* between triple lifespans extracted from Wikipedia). This granular approach allows us to retain detailed information about knowledge updates, facilitating temporal querying and in-depth analysis.

We implement a containerized, horizontally scalable data extraction system to efficiently process many Wikipedia revisions in parallel. This design supports horizontal scaling, enabling the system to handle large-scale data extraction tasks effectively. Our approach ensures that the extraction process remains practical and efficient, even as the volume of data continues to grow.

We provide detailed statistics on the extracted dataset, including metrics such as the total number of data values, change insights, and temporal diversity. These insights shed light on the dynamics of knowledge evolution within Wikipedia and serve as a valuable resource for further research.

Our work makes the following key contributions:

- **Temporal Extension of DBpedia:** We present a method for enriching DBpedia with temporal information by capturing and representing every revision from Wikipedia using a named graph-based temporal graph model.

	BEAR	YAGO15K	Wikidata-T	FinBench	ICEWS	TGBL-WIKI
Source	DyLDO	FB+YAGO	Wikidata	Synth.	ICEWS 05-15	Wiki Co-editing
Temp. Dim.	transaction	validity	validity	transaction	validity	transaction
Timespan	58 weeks	1513-2017	25-2020	since 2020	2005-2015	1 month
Timestamps	58	328	198	not specified	4,017	152,757
Model	IC/CB/TB	TB	TB	ST	TB	ST

Table 1. Overview of temporal KGs, including RDF archiving datasets. “Source” indicates where the data originates, “Temp. Dim” specifies whether the dataset uses transaction or validity time, “Timespan” shows the covered years or weeks, and “Timestamps” counts the distinct temporal points or ranges. Under “Model,” IC = Independent Copies/snapshots, CB = Change-Based diffs, TB = Timestamp-Based intervals (start/end time), and ST = Single Timestamp (only one annotated time).

- **Comprehensive Dataset Statistics:** We offer detailed analyses of the extracted temporal dataset, providing insights into the patterns and trends of knowledge evolution.
- **Scalable Extraction System:** We develop a containerized, horizontally scalable data extraction system that leverages Docker Swarm for orchestration and Apache Spark for efficient large-scale data processing of Wikimedia Meta History data.

2 Related Work

First, we give an overview of existing temporal graph datasets, , and second, we will present the landscape of existing usages of the DBpedia knowledge extraction approaches focused on page versioning

Various temporal graph datasets and generation tools emphasize domain-specific characteristics, temporal resolution, and the nature of the information they encode. We summarize key characteristics and dimension of other TKGs in Table 1. ICEWS [11] and GDELT, for example, capture geopolitical events and interactions over time, enabling analyses of global dynamics and political relationships. In the knowledge domain, YAGO15K [11] and Wikidata [16] enrich standard knowledge graphs with temporal intervals and modifiers, facilitating historical reasoning. Meanwhile, the Temporal Graph Benchmark (TGB) [14] broadens the scope with datasets from social media, finance, and logistics, supporting robust machine learning experiments on dynamic graphs. EventKG and its generation pipeline enhance temporal completeness by merging events and temporal facts from multiple knowledge bases (e.g., Wikidata, YAGO) [12]. In the RDF realm, BEAR variants and RDF Archiving Benchmarks focus on capturing versioned snapshots, changesets, and the querying of evolving data [10]. Tools like EvoGen and the Advanced LUBM Generator systematically produce synthetic temporal graph datasets with controlled evolution over multiple versions [17]. Finally, LDBC FinBench [19] and its generator target the financial sector with scalable, timestamped networks that simulate complex, time-dependent financial transactions. Together, these datasets and tools highlight the breadth

of temporal graph applications, each with its own temporal granularity, thematic focus, and methodological approach to modeling evolving information.

The original DBpedia (core) dataset was extracted on a half-yearly to yearly basis as a snapshot from the latest pages in Wikimedia projects using the the DBpedia information extraction framework (DIEF) [2]. Later, the release cycle was changed to an autonomous process to provide monthly to quarterly snapshot release by reducing the highly required manual effort for execution and validation [13].

Based on the DBpedia ontology and DIEF, other previous works have focused on extracting data on the history information and change event API from Wikipedia. DBpedia Live [18] extends this framework to enable near real-time extraction of current Wikipedia pages, synchronizing RDF data with the latest updates but without preserving historical versions. The Extraction of Historical Events from Wikipedia [8,9] project extracted historical events from various types of Wikipedia articles, storing them in a database for display on a timeline. However, this work is about tracking the edit history in the articles, and it did not capture the changes of triples based on the Wikipedia updates, not creating a temporal knowledge graph. The DBpedia Wayback Machine [5] allows querying historical versions of individual Wikipedia resources at specific points in time, retrieving articles in RDF format aligned with the DBpedia ontology. It queries revisions via the MediaWiki API but does not construct a temporal graph over the DBpedia data.

In contrast, this work proposes creating a temporal knowledge graph that captures and stores historical changes over time for all Wikipedia articles, including their updates and transaction times. This enables detailed analysis of content evolution and provides a comprehensive benchmark dataset, going beyond the capabilities of previous approaches.

3 Temporal Extraction

The English Wikipedia meta history dump as of 2024-06-01 contains over 1 billion revisions, covering around 70 million pages. Like DBpedia Live, we leverage the DIEF server component to implement a web-service-driven architecture. This approach enables scalable extraction processes and facilitates seamless interaction with the DIEF while avoiding the need to fork or extensively modify the original codebase. We adopt a microservice-oriented approach for the extraction process, deploying multiple instances of the DBpedia extraction server. A central extraction orchestration unit coordinates these instances to ensure efficient execution. The meta-history dump releases provided by Wikimedia are partitioned into 862 separate files. This partitioning enables the parallel processing of multiple files, significantly improving performance and scalability.

Figure 1 illustrates the high-level workflow of the DBpedia Temporal Knowledge Graph (TKG) generation process. The Wikimedia history revision dumps, partitioned into 862 files, contains wiki pages along with their revisions in XML. This partitioning provides a strong foundation for parallel processing by the

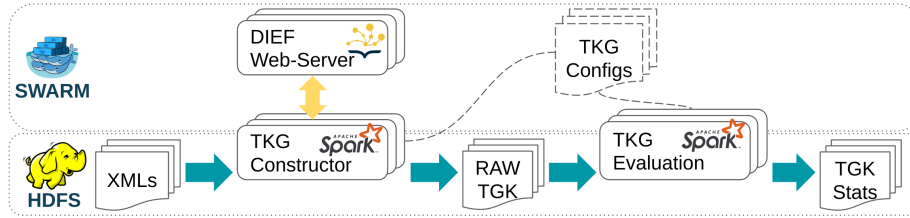


Fig. 1. Architecture and Data Flow Diagram of the DBpedia TKG Extraction.

TKG Constructor. Apache Spark enables efficient distributed processing of the dumps across multiple nodes. A Spark job assigns dump files to individual workers, which process them one at a time by communicating with the HTTP API of a deployed DIEF extraction service while reading from and writing to an Hadoop File System (HDFS). This approach ensures efficient parallelization and reduces overall execution time.

3.1 TKG Configuration Profiles

Our extraction allows and requires several configuration options, grouped into DIEF Extractor Selection, Page Filters, and TKG construction parameters.

- **DIEF Extractor Selection:** The main branch of the DBpedia Extraction framework provides 86 extractor implementations, each generating specific datasets. As some extractors serve a different purpose, intending to extract specific information, they might require different resources than others. For scalability, it will be necessary to select only fast-working extractors.
- **Page Filter:** The Wikimedia dump contains different types of pages. The pages are grouped into namespaces, like article page, user, talk, template, and category pages. Further, the revisions of the Wikidump are created in no regular timeseries. So revisions can be filtered by a timestamp pattern (cron expression), accepting only sequentially fitting revisions.
- **TKG Construction Parameters:** The TKG process will allow further parameters to configure the produced TKG. Currently, parameters that limit the maximum extraction time per revision are supported. Later, post-processing and consistency functions can be added in future work.

3.2 Data Model

Temporal graphs model the evolution of graph structures over time, offering key benefits in versioning across various domains such as social networks [22], transportation networks [15], and evolving knowledge graphs. Approaches differ in how they represent time, ranging from snapshot-based methods that store discrete historical states to timestamped models associating temporal metadata with nodes or edges and time-interval graphs that capture valid periods. While

Property Graph Models (PGM) [1] typically store temporal information as properties, extended models like the Temporal Property Graph Model (TPGM) [20] integrate time as a core concept. In RDF contexts, maintaining temporal versions is achieved through snapshot strategies (either full materialization or delta-based), or by enriching triples with temporal information [21,24]. The latter approach is mostly done via explicit reification, implicit abstraction, or named graphs [3]. The named graph model naturally attaches temporal metadata to RDF statements by using their graph identifiers as subjects in annotation triples within a separate meta-graph.

While this diversity of graph models provides a broad spectrum, the later usage might still require a different model. Therefore, we decided first to extract and build a simple tabular serialization of our temporal graph, where the revision and timespans for each extracted triple from the Wikipedia dump are tracked.

```

1 head,      rel,      tail,      tStart,    tEnd,      rStart, rEnd
2 dbo:Leipzig, dbo:populationTotal, "510043"^^<xsd:integer>, 2019-08-13, 2019-10-28, 389, 647
3 dbo:Leipzig, dbo:populationTotal, "605407"^^<xsd:integer>, 2019-10-28, 2021-04-11, 647, 250
4 ... # headers, datetime and revision ids are shortened

```

We argue that it is easy to build different temporal graph representations in the Property Graph Model (PGM) and Resource Description Framework (RDF) from this TKG extraction table with columns like `head`, `relation`, `tail`, `timeStart`, `timeEnd`, `revisionStart`, and `revisionEnd` because this structure aligns with some graph requirements. The `head`, `relation`, and `tail`³ define entities and relationships, while `timeStart` and `timeEnd` capture triple lifespan, and `revisionStart` and `revisionEnd` track data provenance by holding the ids of the first revision with this triple and the revision removing this triple, respectively. Generating these graphs primarily requires grouping data by the head or property columns and applying a map operation to create nodes, edges, or triples with temporal and revision attributes. This approach also facilitates the creation of unique identifiers for each relationship or entity instance, enabling seamless transformation into expressive temporal graph models.

```

1 <dbr/Leipzig> <dbo/populationTotal> "510043" <tkg/..389-..647> . # lifespan over ?g below
2 <tkg/..389-..647> <tkg/extractedFrom> "2019-08-13 12:02:50"^^<xsd:dateTime <tkg> .
3 <tkg/..389-..647> <tkg/extractedUntil> "2019-10-28 19:35:51"^^<xsd:dateTime <tkg> .
4
5 <dbr/Leipzig> <dbo/populationTotal> "605407" <tkg/..647-..250> . # lifespan over ?g below
6 <tkg/..647-..250> <tkg/extractedFrom> "2019-10-28 19:35:51"^^<xsd:dateTime <tkg> .
7 <tkg/..647-..250> <tkg/extractedUntil> "2021-04-11 18:51:06"^^<xsd:dateTime <tkg> .

```

To demonstrate a possible conversion, we show a snippet of the tabular TKG data converted to the named graph RDF representation⁴ for triple-level timestamp annotations.

³ The tail is either the (datatype-)literal or URI part of the produced DIF NTriple

⁴ Due to size limits @Zenodo, we released a NQ version separately (Code URL)

3.3 Current Limitations

The current generator exhibits several limitations that require consideration. Although the generation process supports horizontal scaling, not all extractors are consistently applied. This inconsistency is primarily due to certain extractors depending on external services, such as text extraction tools, which may not always be available. Additionally, the generator does not account for changes in the ontology. The extraction process was conducted using the ontology and mappings version as of 2024-06-01, and any updates to the ontology necessitate a full re-computation. Compounding this issue, ontology versions are not aligned with the timestamps of data revisions, which can lead to discrepancies.

The DBpedia release process also involves additional preprocessing steps, adding complexity to the workflow. Furthermore, some external or third-party datasets, such as SD types, are included in the final classic releases, introducing dependencies that are not entirely transparent. Lastly, while changes reflected in the DBpedia TKG are based on edits, the validity of a statement is not intrinsically bound to the timestamp of those edits, raising concerns about temporal accuracy in the generated knowledge graph.

4 Evaluation

We deployed the extraction on a docker-swarm environment consisting of 8 Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz Nodes with 36 cores (72 threads), 376GB RAM, and 20GBit network bandwidth. For our extraction architecture deployment, we used the following configuration:

- **DBpedia ontology version:** 2024-10-29
- **DBpedia mappings version:** 2024-10-29
- **Revision filters:** Only *article* and *category* pages but all revisions.
- **Construction parameters:** DIEF server request timeout of 10 seconds.

Extractor Selection. For our initial TKG extraction version, we selected the following available DIEF extractors:

- **Labels:** The unique name of a DBpedia resource extracted from the page title as *rdfs:label* triple.
- **Infobox Generic:** Basic infobox key-value pair extraction using simple rules to parse the values as text, numbers, dates, and links to other entities.
- **Anchor Text:** Links from a page, as either *WikiPageWikiLink* for resources in Wikipedia (DBpedia) or as *ExternalLink* for external resources.
- **Categories:** Generates category information for pages of the category *namespace* using the Simple Knowledge Organization System (SKOS) vocabulary.
- **Mappings:** The mappings extraction of DBpedia using predefined mappings between Infobox-templates to the DBpedia ontology and applies better value and datatype normalization as well.
- **Metadata*:** Extracts page metadata information, like the original *WikiPageID* for the extracted resource.

With a scale factor of 16 DIEF services per Node (128 total), and eight TKG-Construction processes, each using 16 DIEF services, the extraction of the first FULL TKG version was completed in less than 3 days ($\sim 52\text{h}$), with an average of 4300 revisions per second (~ 33 per TKG Constructor). We provide more statistics on resource consumption and scale-factors in the repository under the evaluation benchmark section.

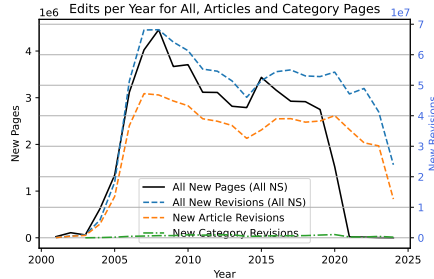


Fig. 2. Newly added wiki pages & revisions per year.

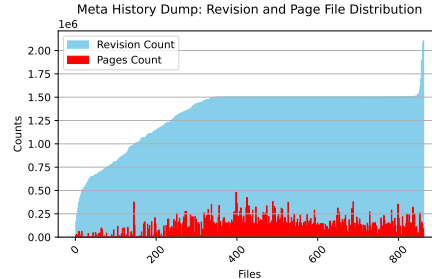


Fig. 3. Revision and Page counts of the 862 Wikipedia input dump files.

Table 2 shows an overview of the processed input wiki data, the number of selected pages, revisions, and the size of the produced output TKG (as CSV version). From the 812 million selected revisions, only 12,7 million were of the type category page, and the rest article pages. Due to the current set restrictions of a limit of 10 seconds per revision extraction (DIEF request timeout), less than 0.0005% revisions had been skipped.

Input size	Input pages	Input revs.	Select pages	Select revs.	Fail/Skip	Output size
25.5TB	60,810,684	1,076,085,956	20,186,460	812,076,920	<0,0005%	293GB

Table 2. Wikidump Input and TKG Output statistics (size for uncompressed data).

Figure 2 shows the creation of new pages and revisions for all namespaces per year, including new article revisions and new category revisions per year in the dataset time range from 2001 to 2024. As the number of revisions is an order of magnitude larger than the number of pages, we have used a different scale (blue) for better visualization. Figure 3 visualizes the distribution of pages and revisions per dump file. The number of pages increases continuously with the number of dumps, while the number of revisions is fixed at around 1.5 million. The partition shows a similar distribution of revisions per dump file, supporting the decision to scale the extraction by parallelizing the process by extracting TKG data from multiple input files simultaneously.

4.1 Dataset Overview

	Version Triples	Unique Triples	Nodes	Types	Relations	Timestamps
DBP-TKG_{FULL}	1,663,152,137	1,088,513,441	26,182,006	794	375,129	271,455,850
DBP-TKG_{WPL}	856,809,538	493,981,357	17,828,226	*	1	228,412,407
DBP-TKG_{DBO}	213,764,478	152,975,581	25,779,983	501	1617	65,385,286
DBP-TKG_{CAT}	12,892,255	12,558,676	2,356,961	1	3	6,714,398

Table 3. Statistics of the extracted DBP-TKGs, including the full main version and subgraphs of different vocabularies. Timestamps range from 2001 to 2024.

Before delving into specific insights on the temporal changes within the graph, we first present summaries of the extracted data to provide a comprehensive overview.

Table 3 shows statistics of the fully generated TKG. Further, we provide insights on three subgraphs we extracted from the full main version by selecting different vocabularies for property and type URIs:

- 1) DBP-TKG_{DBO} is a subgraph, only using the DBpedia ontology vocabulary,
- 2) DBP-TKG_{CAT} is a subgraph of the extracted category system, and
- 3) DBP-TKG_{WPL} is a subgraph only with the Wikipedia page wiki links, which is in size almost half of the extracted total triples, underlining the dynamicity of edits on page revision, which is to be expected as one of the most edits in Wikipedia in the sense of structured information (besides text edits).

4.2 Changes over time

Exploring temporal dynamics of the generated TKG is crucial for understanding and analyzing its structural evolution. This subsection illustrates four key aspects of these changes: the yearly growth of the data, yearly degree distribution, yearly triple changes, and their lifespans. By examining such characteristics, we aim to provide a comprehensive overview of the graph’s progression over time and its implications for downstream applications.

Yearly Growth. Figure 4 shows the aggregated yearly growth of triples, unique entities, types, relations, and versions, of yearly snapshot in the covered time range from 2001 to 2024. Triples, entities, and versions continuously increase over the lifespan, while the unique relations have their maximum in 2015 and converge on a value of $\sim 80k$ for the remaining years. The unique types converge way earlier (around 2010), since barely no new types were introduced since then. Additionally, since types are extracted from infoboxes, which started to be widely used around 2005, the first types occur in the respective snapshot.

Yearly Out Degree Distribution. Another way to visualize the temporal diversity of the extracted data is by representing the evolution metric of the out-degree. Figure 5 shows the changes in the degree distribution of the graph for each of the 24 yearly snapshots. The data for this is from the FULL version and

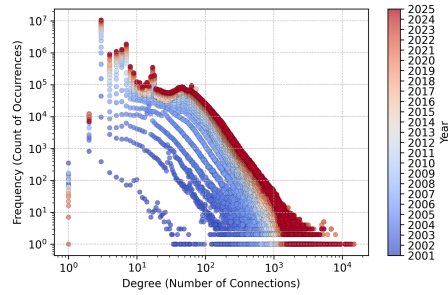
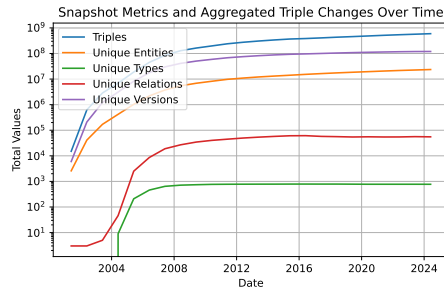


Fig. 4. Yearly growth over the 24 extracted time snapshots of DBP-TKG_{FULL}. **Fig. 5.** Out degree distribution of the 24 yearly snapshot versions.

consists of datatype and object properties, as removing the datatype properties did not change much on the visualization outcome. The graph is getting more connected over time as the available pages and infobox templates in Wikipedia also grow each year, making it easier to provide more extractable information on each revision for new and older page.

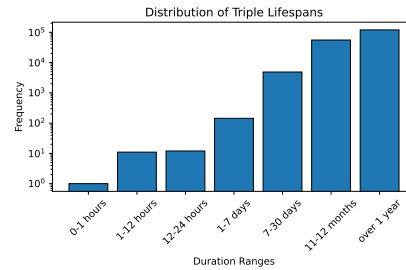
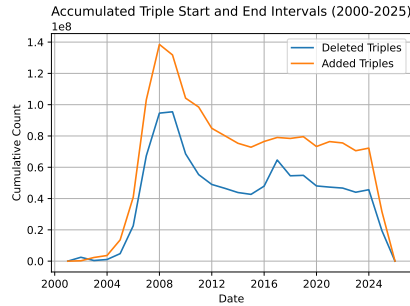


Fig. 6. Yearly triple changes, by additions and deletions over time.

Fig. 7. Frequency of different triple lifespans. Excluding open ends.

Triple Changes. Figure 6 shows triple additions and deletions for a time granularity of 1 year. This includes triple changes not captured by the time snapshots, as a triple can be added or removed multiple times between two snapshots. The pattern of these changes aligns with the pattern in Figure 2, showing that new revisions often reflect in extracted triple.

Triple Lifespan. The triple lifespans ranged from 1 second to over 23 years, whereas the latter mainly were Wikipedia page wiki links. Figure 7 provides a distribution of triple lifespans, where it is evident that most triples, when introduced by a revision, are rarely or never changed. We removed the open-ended triples from this statistic (over 300M) and only considered closed intervals.

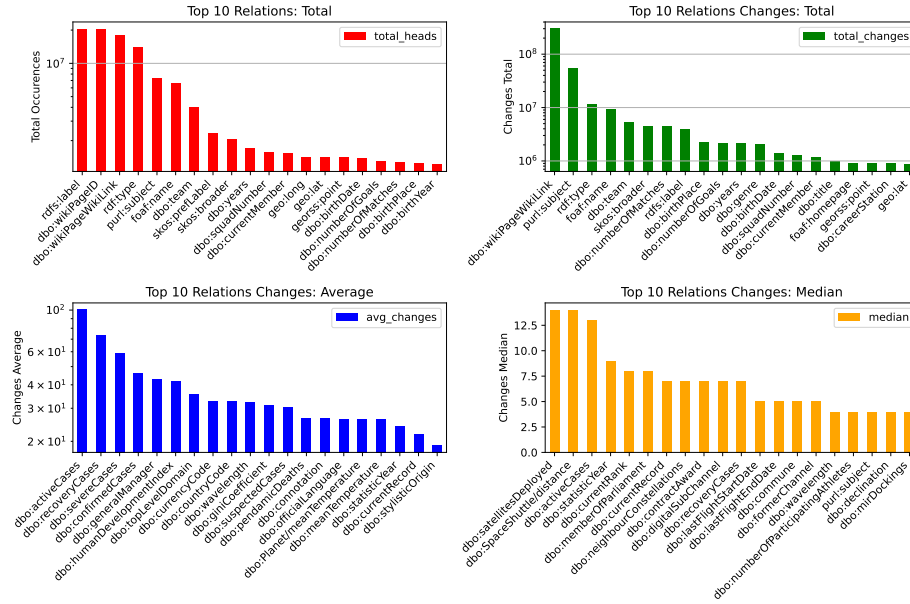


Fig. 8. Statistics on total usage and property changes for the Top 10 most used, total, average, and median changes per property.

4.3 Property Changes

As a last insight, we wanted to investigate the properties that are affected highly and slowly by new revisions. We select the top 10 most used properties, the top 10 most changed, and the top 10 with the highest average and highest median changes per page properties (Figure 8). The full statistics on each property are available as supplement data, in the dataset repository. A change is defined as the number of different value-lifespans pairs for a subject-predicate pair minus one.

Properties like `rdfs:label`, `rdf:type`, `foaf:name`, and `dbo:birthDate` appear in large numbers of nodes (high total usage) but have relatively low median and often zero 25th percentile changes. This suggests that while these core descriptive properties are very common, they tend to remain stable over time for most entities. For instance, `rdfs:label` is extremely widely used (for over 20 million total entities) but has a median of 0 changes, proving that once a page title is set, it rarely changes (as intended). This is consistent with the idea that basic factual attributes (names, labels, and core types) don’t fluctuate dramatically over time.

The property `dbo:wikiPageWikiLink`, which represents the interlinking of pages, shows an enormous number of total changes (over 299 million total changes) and a very high total usage count, which aligns with the insights on the DBP-TKG_{WPL} subgraph. However, the median changes per node is just 1, and the

75th percentile is only 11 changes. This strongly suggests a highly skewed distribution: a small subset of nodes (pages) undergo frequent and substantial link rewiring, while the majority remain relatively stable. In other words, some entities or topics are "hotspots" of editorial activity, while others see only occasional changes.

On the list ordered by average changes, you see properties like *dbo:activeCases*, *dbo:recoveryCases*, *dbo:confirmedCases*, and other health/statistics-related properties. These have extremely high average changes per node but are associated with comparatively fewer total entities. This implies a small subset of nodes (possibly related to weekly or dynamically updated data, such as public health statistics or economic indicators) undergo frequent and large-scale updates. This mirrors real-world phenomena where certain data points, like COVID-19 case numbers, economic figures, or population counts—are inherently more volatile over time.

Another pattern is properties like *dbo:numberOfGoals*, *dbo:currentMember*, or *dbo:numberOfMatches* show moderate to high total changes and usage counts. Sports- or performance-related data (like a player’s number of goals or current club membership) change as careers progress. The distributions and quantiles of changes hint that while some players’ data may remain stable for long periods, others experience more frequent updates, possibly reflecting transfers, new statistics, and continuous record-keeping.

Many properties have a median of 0 changes but a potentially large maximum number of changes. This signals a classic "long tail" distribution where most nodes are stable, and only a minority experience high-frequency updates. Thus, the temporal aspect is not uniform across the graph, but it is concentrated in certain nodes or property domains.

The graph captures diverse temporal dynamics, with stable identifiers and rapidly changing metrics showcasing a meaningful and dynamic dataset. High-change properties, like wiki links, offer rich opportunities to study knowledge evolution and editorial patterns over time. However, skewed distributions could challenge analysis by overemphasizing outliers, and the uneven stability of properties may hinder efforts to maintain a uniformly well-curated temporal record.

4.4 Alignment of Edits and Global Events

As we made clear in the introduction (Section 1), the temporal dimension of the extracted triples only reflects their existence as information in Wikipedia as transaction time in the Media-wiki database. However, as we can see, Wikipedia information is still highly updated, affecting the extracted triples differently. In this section, we try to show that for some of these triple changes, the edit frequency is high enough to reflect the evolution of global knowledge over time.

We inspect two different DBpedia ontology properties and their value changes for their assigned page entity. We applied a simple outlier cleaning function to remove outliers based on bad edits. First, we selected the *dbo:activeCases* property due to its prominence in Figure 8, and then focused on the top five entities that underwent the most changes. As shown in Figure 9, this property tracked active

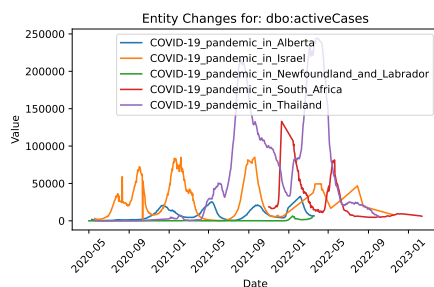


Fig. 9. Top 5 events with most changes for *dbo:activeCases*.

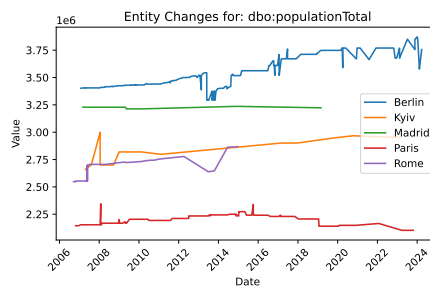


Fig. 10. European countries changes for *dbo:PopulationTotal* (1-5M).

COVID-19 case counts across different regions and was first introduced in May 2020. The identified increases in edits to this property correspond with slight global rises in reported cases (We have not independently verified the accuracy of these numbers).

Next, we choose the property *dbo:population*, which is used to display the population of entities of type place, like cities, regions, and countries. Here, we picked five European cities with this property with a value range between 1 and 5 million (see Figure 10). Over the years, Wikipedia’s population data for cities like Rome, Paris, Berlin, Madrid, and Kyiv has fluctuated frequently and sometimes erratically, reflecting editorial debates, inconsistent sourcing, and shifting definitions rather than true demographic changes. It is also noticeable that for Rome and Madrid, the value (triple) was removed in 2015 and 2019, never being reintroduced by new revisions (possibly due to an infobox issue). Stable periods emerge, typically following the establishment of a credible source, but are interrupted by frequent edits due to new external information, differing editor methodologies, or disputes. As a result, the historical population figures documented on Wikipedia should be viewed as a record of evolving consensus among volunteers rather than an authoritative or directly reliable time series of actual population trends.

5 Dataset Discussion

In its current form, the DBpedia-TKG dataset primarily targets research communities working with temporal knowledge graphs (TKGs). While refining data quality through cleaning, subset selection, and outlier detection remains a challenge, this large-scale, real-world temporal resource enables research previously constrained by static snapshots. The dataset provides a foundation for temporal entity matching, question answering (QA), evolutionary analysis, temporal machine learning, and benchmarking, pushing the field toward more dynamic and realistic knowledge graph research.

The DBpedia-TKG dataset enables a broad range of temporal knowledge graph tasks and methodologies that were previously limited by static represen-

tations. With time-annotated facts, evolving entities such as cities experiencing administrative changes (e.g., Leipzig’s district reformation), political figures with tenure changes (e.g., U.S. presidents), or companies merging or splitting (e.g., Google’s reorganization into Alphabet Inc.) can be more accurately tracked. These capabilities improve dynamic identity resolution and time-sensitive entity recognition [4].

Temporal annotations enable time-restricted question-answering (QA) and more nuanced reasoning over evolving facts. For example, “*Which countries have changed their head of state between 2010 and 2020?*”⁵ illustrates retrieving fact evolution instead of just the latest state, shifting focus toward causality-aware, historically accurate answers beyond static entity retrieval.

However, it is essential to note that the temporal annotations in DBpedia-TKG reflect transaction time when a fact was added, changed, or removed in Wikipedia, not necessarily its real-world validity time. For example, a population figure added in 2022 may refer to census data from 2020 (see Section 4.4). Therefore, temporal patterns in the DBpedia-TKG reflect the editorial history and visibility of facts, rather than their factual occurrence in the real world. We encourage users to keep this distinction in mind when interpreting time-based analyses or time-sensitive tasks such as temporal QA and historical reasoning.

Beyond entity-level tasks, the dataset supports evolutionary analysis and temporal graph mining by capturing node, edge, and community dynamics. These insights enable pattern discovery, long-term trend analysis, and the study of structural evolution in dynamic knowledge graphs [6]. For instance, monitoring Wikipedia categories over time can reveal trends such as: “*How did the number of Nobel Prize winners per country evolve over the last 10 years?*”.

In machine learning research, timestamped data facilitates training temporal embeddings and time-aware models for link prediction, event forecasting, and anomaly detection [25,14,11]. These models better capture evolving relationships and predict structural changes over time. Open questions remain, for example: “*How can large-scale TKGs handle contradictory facts from Wikipedia’s evolving editorial process?*”, or “*How accurately can TKG-based models forecast real-world changes based on past trends?*”.

Beyond modeling challenges, the temporal dimension of DBpedia-TKG enables advanced benchmarking and query assessment by extending existing DBpedia workloads. Time-specific subsets, filters, date comparisons, and interval constraints support more realistic queries (e.g., “*Retrieve all changes to dbr:Leipzig from 2010 to 2020*”), refining RDF archiving strategies and testing scalability in SPARQL-based frameworks, as inspired by benchmarks like BEAR [10] and FinBench [19]. Notably, DBpedia-TKG is significantly larger than previous resources and captures real-world Wikipedia edits, providing a rich, evolving testbed for both fundamental research and practical applications in temporal knowledge graphs.

⁵ Mentioned and additional example queries (with SPARQL) can be found at <https://github.com/dbpedia/dbpedia-temporal/blob/main/examples/queries.md>

Currently, the temporal extension cannot fully replace main DBpedia snapshots due to missing extractors and the absence of a post-processing step. Future work may address data completeness by integrating additional extractors (validated for scalability and performance) and adapting the existing postprocessing workflow, including DIFE filters, to handle data inconsistencies, e.g., removing improbable value changes based on lower and upper percentiles. Aligning DBpedia ontology versions with revisions via DBpedia Archivo [7] could further enhance consistency, and refining evaluation metrics would improve result quality. Supporting multiple Wikipedia language editions would enable multilingual TKGs, while extending DBpedia Live could pave the way for real-time updates. Lastly, incrementally adding new triple changes based on a previous TKG configuration profile could streamline updates and boost efficiency.

In the meantime, we will try to produce only two updates per year, given the size of the extracted data and its current primary role as a research dataset.

6 Conclusion

This paper introduces the DBpedia Temporal Knowledge Graph (DBpedia-TKG), an extension to the DBpedia dataset. By incorporating temporal dynamics, our approach captures the evolution of knowledge over time in Wikipedia with the DBpedia ontology. The DBpedia-TKG offers a scalable and configurable solution, enabling the extraction of temporal datasets from Wikipedia’s extensive meta-history dumps. Our first complete extraction spans 1.7 billion triples with 270 million distinct change time points.

DBpedia-TKG provides a transformative resource for applications ranging from entity disambiguation to temporal graph analytics by enabling researchers to explore the dynamics of knowledge evolution. We anticipate that this dataset will catalyze new advancements in (temporal) knowledge graph research and can allow innovative methodologies that account for the dynamic nature of real-world data.

Despite its initial successes, several challenges and limitations remain, including incorporating additional extractors, handling ontology versioning, and addressing inconsistencies in extracted data. Future directions include the generation of multilingual TKGs, real-time temporal updates via DBpedia Live, and integration with temporal property graph frameworks to leverage advanced analytic capabilities.

The documentation and source code is publicly available in the GitHub repository (see code URL in abstract). Besides, service deployment and execution, it also contains example scripts and tooling for generating the snapshots, subgraphs, and running the metrics generation.

Acknowledgments. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany and by the Sächsische Staatsministerium für Wissenschaft Kultur und Tourismus in the program Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: ScaDS.AI.

References

1. Angles, R.: The property graph database model. In: Olteanu, D., Poblete, B. (eds.) Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management, Cali, Colombia, May 21-25, 2018. CEUR Workshop Proceedings, vol. 2100. CEUR-WS.org (2018), <https://ceur-ws.org/Vol-2100/paper26.pdf>
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: Dbpedia: A nucleus for a web of open data. In: The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Lecture Notes in Computer Science, vol. 4825, pp. 722–735. Springer (2007). https://doi.org/10.1007/978-3-540-76298-0_52, https://doi.org/10.1007/978-3-540-76298-0_52
3. Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. *J. Web Semant.* **3**(4), 247–267 (2005)
4. Christen, P., Gayler, R.W.: Adaptive temporal entity resolution on dynamic databases. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II. Lecture Notes in Computer Science, vol. 7819, pp. 558–569. Springer (2013). https://doi.org/10.1007/978-3-642-37456-2_47, https://doi.org/10.1007/978-3-642-37456-2_47
5. Fernández, J.D., Schneider, P., Umbrich, J.: The dbpedia wayback machine. In: Polleres, A., Pellegrini, T., Hellmann, S., Parreira, J.X. (eds.) Proceedings of the 11th International Conference on Semantic Systems, SEMANTiCS 2015, Vienna, Austria, September 15-17, 2015. pp. 192–195. ACM (2015). <https://doi.org/10.1145/2814864.2814889>, <https://doi.org/10.1145/2814864.2814889>
6. Fournier-Viger, P., He, G., Cheng, C., Li, J., Zhou, M., Lin, J.C., Yun, U.: A survey of pattern mining in dynamic graphs. *WIRES Data Mining Knowl. Discov.* **10**(6) (2020). <https://doi.org/10.1002/WIDM.1372>, <https://doi.org/10.1002/widm.1372>
7. Frey, J., Streitmatter, D., Götz, F., Hellmann, S., Arndt, N.: Dbpedia archivo: A web-scale interface for ontology archiving under consumer-oriented aspects. In: Blomqvist, E., Groth, P., de Boer, V., Pellegrini, T., Alam, M., Käfer, T., Kieseberg, P., Kirrane, S., Meroño-Peñuela, A., Pandit, H.J. (eds.) Semantic Systems. In the Era of Knowledge Graphs - 16th International Conference on Semantic Systems, SEMANTiCS 2020, Amsterdam, The Netherlands, September 7-10, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12378, pp. 19–35. Springer (2020). https://doi.org/10.1007/978-3-030-59833-4_2, https://doi.org/10.1007/978-3-030-59833-4_2
8. Gandon, F., Boyer, R., Corby, O., Monnin, A.: Materializing the editing history of wikipedia as linked data in dbpedia. In: Kawamura, T., Paulheim, H. (eds.) Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 19, 2016. CEUR Workshop Proceedings, vol. 1690. CEUR-WS.org (2016), <https://ceur-ws.org/Vol-1690/paper41.pdf>
9. Gandon, F., Boyer, R., Corby, O., Monnin, A.: Wikipedia editing history in dbpedia: Extracting and publishing the encyclopedia editing activity as linked data. In: 2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, Omaha, NE, USA, October 13-16, 2016. pp. 479–482. IEEE Computer

- Society (2016). <https://doi.org/10.1109/WI.2016.0079>, <https://doi.org/10.1109/WI.2016.0079>
10. Garcia, J.D.F., Umbrich, J., Polleres, A.: Bear: Benchmarking the efficiency of rdf archiving (2015)
 11. García-Durán, A., Dumancic, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. pp. 4816–4821. Association for Computational Linguistics (2018), <https://aclanthology.org/D18-1516/>
 12. Gottschalk, S., Demidova, E.: Eventkg: A multilingual event-centric temporal knowledge graph. In: Gangemi, A., Navigli, R., Vidal, M., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M. (eds.) The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10843, pp. 272–287. Springer (2018). https://doi.org/10.1007/978-3-319-93417-4_18, https://doi.org/10.1007/978-3-319-93417-4_18
 13. Hofer, M., Hellmann, S., Dojchinovski, M., Frey, J.: The new dbpedia release cycle: Increasing agility and efficiency in knowledge extraction workflows. In: Blomqvist, E., Groth, P., de Boer, V., Pellegrini, T., Alam, M., Käfer, T., Kieseberg, P., Kirrane, S., Meroño-Peñuela, A., Pandit, H.J. (eds.) Semantic Systems. In the Era of Knowledge Graphs - 16th International Conference on Semantic Systems, SEMANTiCS 2020, Amsterdam, The Netherlands, September 7-10, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12378, pp. 1–18. Springer (2020). https://doi.org/10.1007/978-3-030-59833-4_1, https://doi.org/10.1007/978-3-030-59833-4_1
 14. Huang, S., Poursafaei, F., Danovitch, J., Fey, M., Hu, W., Rossi, E., Leskovec, J., Bronstein, M.M., Rabusseau, G., Rabbany, R.: Temporal graph benchmark for machine learning on temporal graphs. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023), http://papers.nips.cc/paper_files/paper/2023/hash/066b98e63313162f6562b35962671288-Abstract-Datasets_and_Benchmarks.html
 15. Jiang, W., Luo, J.: Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.* **207**, 117921 (2022). <https://doi.org/10.1016/J.ESWA.2022.117921>, <https://doi.org/10.1016/j.eswa.2022.117921>
 16. Leblay, J., Chekol, M.W.: Deriving validity time in knowledge graph. In: Champin, P., Gandon, F., Lalmas, M., Ipeirotis, P.G. (eds.) Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018. pp. 1771–1776. ACM (2018). <https://doi.org/10.1145/3184558.3191639>, <https://doi.org/10.1145/3184558.3191639>
 17. Meimaris, M., Papastefanatos, G.: The evogen benchmark suite for evolving RDF data. In: Debattista, J., Umbrich, J., Fernández, J.D., Rula, A., Zaveri, A., Knuth, M., Kontokostas, D. (eds.) Joint Proceedings of the 2nd Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW 2016) and the 3rd Workshop on Linked Data Quality (LDQ 2016) co-located with 13th European Semantic Web Conference (ESWC 2016), Heraklion, Crete, Greece, May 30th,

2016. CEUR Workshop Proceedings, vol. 1585, pp. 20–35. CEUR-WS.org (2016), https://ceur-ws.org/Vol-1585/mepdaw2016_paper_03.pdf
18. Morsey, M., Lehmann, J., Auer, S., Stadler, C., Hellmann, S.: Dbpedia and the live extraction of structured data from wikipedia. *Program* **46**(2), 157–181 (2012). <https://doi.org/10.1108/00330331211221828>, <https://doi.org/10.1108/00330331211221828>
 19. Qi, S., Lin, H., Guo, Z., Szárnyas, G., Tong, B., Zhou, Y., Yang, B., Zhang, J., Wang, Z., Shen, Y., Wang, C., Peiravi, P., Gabb, H.A., Steer, B.A.: The LDBC financial benchmark. *CoRR* **abs/2306.15975** (2023). <https://doi.org/10.48550/ARXIV.2306.15975>, <https://doi.org/10.48550/arXiv.2306.15975>
 20. Rost, C., Gómez, K., Täschner, M., Fritzsche, P., Schons, L., Christ, L., Adameit, T., Junghanns, M., Rahm, E.: Distributed temporal graph analytics with GRADOOP. *VLDB J.* **31**(2), 375–401 (2022). <https://doi.org/10.1007/S00778-021-00667-4>, <https://doi.org/10.1007/s00778-021-00667-4>
 21. Wang, H., Tansel, A.U.: Temporal extensions to RDF. *J. Web Eng.* **18**(1-3), 125–168 (2019)
 22. Wang, Z., Li, Z., Yuan, G., Sun, Y., Rui, X., Xiang, X.: Tracking the evolution of overlapping communities in dynamic social networks. *Knowl. Based Syst.* **157**, 81–97 (2018). <https://doi.org/10.1016/J.KNOSYS.2018.05.026>, <https://doi.org/10.1016/j.knosys.2018.05.026>
 23. Zaki, A., Attia, M., Hegazy, D., Amin, S.: Comprehensive survey on dynamic graph models. *International Journal of Advanced Computer Science and Applications* **7**(2) (2016)
 24. Zhang, F., Li, Z., Peng, D., Cheng, J.: RDF for temporal data management - a survey. *Earth Sci. Informatics* **14**(2), 563–599 (2021)
 25. Zhang, Y., Kong, X., Shen, Z., Li, J., Yi, Q., Shen, G., Dong, B.: A survey on temporal knowledge graph embedding: Models and applications. *Knowl. Based Syst.* **304**, 112454 (2024). <https://doi.org/10.1016/J.KNOSYS.2024.112454>, <https://doi.org/10.1016/j.knosys.2024.112454>