Knowledge Base Evolution Analysis: A Case Study in the Tourism Domain

Knowledge Base Evolution Analysis: A Case Study in the Tourism Domain

Mohammad Rashid[1], Giuseppe Rizzo[2], Marco Torchiano[1], Nandana Mihindukulasooriya[3], and Oscar Corcho[3]

[1] Politecnico di Torino, Italy
[2] Instituto Superiore Mario Boella, Italy
[3] Universidad Politécnica de Madrid, Spain

Abstract. Stakeholders – curator, consumer, etc. – in the tourism domain routinely need to combine and compare statistical indicators about tourism. In this context, various Knowledge Bases (KBs) have been designed and developed in the Linked Open Data (LOD) cloud in order to support decision-making process in Tourism domain. Such KBs evolve over time: their data (instances) and schemes can be updated, extended, revised and refactored. However, unlike in more controlled types of knowledge bases, the evolution of KBs exposed in the LOD cloud is usually unrestrained, what may cause data to suffer from a variety of issues. This paper attempts to address the impact of KB evolution in tourism domain by showing how entity evolves over time using the 3cixty KB. We show that using multiple versions of the KB through time can help to understand inconsistency in the data collection process.

Keywords: Knowledge Base · Linked Data · Evolution Analysis.

1 Introduction

In the recent years much efforts have been given towards sharing Knowledge Bases (KBs) in the Linked Open Data (LOD) cloud[4]. Large KBs in the tourism domain are often maintained by organizations that act as curators to ensure their quality [9]. These KBs naturally evolve due to several causes: *(i)* resource representations and links that are created, updated, and removed; *(ii)* the entire graph can change or disappear. In general, KBs in the tourism domain are highly complex and dynamic in nature. Decision-makers often rely on forecasting models to predict future demand or on decision support systems to analyze and compare the relevant stakeholders [9]. Whilst most datasets are published as open data, the data publishers continuously try to improve the quality of their data by updating ontologies and data instances or removing obsolete ones. However, unlike in more controlled types of knowledge bases, the evolution of KBs in the tourism domain may suffer from a variety of issues, both at a semantic (contradiction) and at a pragmatic level (ambiguity, inaccuracies). This situation clearly affects negatively data stakeholders such as consumers, curators.

[4] http://lod-cloud.net

Taking into consideration a KB, we believe that understanding this evolution could help to define more suitable strategies for data sources integration, enrichment, and maintenance. One of the common tasks for KB evolution analysis is to perform a detailed data analysis, with data profiling. Data profiling is usually defined as the process of examining data to collect statistics and provide relevant metadata [1]. Based on data profiling we can thoroughly examine and understand a KB, its structure, and its properties before usage.

In this paper, we explored the impact of KB evolution in the tourism domain using the 3cixty KB [10]. The core idea in this work is to use dynamic features from data profiling results for analyzing the evolution of KBs. The main contributions of this work are: (1) a fundamental overview about the topic of KB evolution analysis; and (2) the presentation of the 3cixty KB as a use case to understand the impact of KB resource evolution. Furthermore, we used two entity types to explored the stability characteristics to identify any inconsistency present in the data extraction process. In this context, we created a set of APIs[5] for periodic snapshots generation and maintaining scheduled tasks for automatic and timely checks. We explored KB evolution analysis with *lode:Event*[6] and *dul:Places*[7] entity-type in the 3cixty KB, reporting the benefits of KB evolution analysis.

We continue by describing the details of our use case in Section 2, then we provide technical details about the KB evolution analysis in Section 3 and Stability characteristics in Section 4. Section 5 present an experimental analysis. We outline the related works in Section 6 and conclude in Section 7.

2 Use Case: The 3cixty KB

3cixty is a knowledge base that describes cultural and tourist information. This knowledge base was initially developed within the 3cixty project[8], which aimed to develop a semantic web platform to build real-world and comprehensive knowledge bases in the domain of culture and tourism for a few cities. The entire approach has been tested first in the occasion of the Expo Milano 2015 [8], where a specific knowledge base for the city of Milan was developed, and has now been refined with the development of knowledge bases for the cities of Nice, London, Singapore, and Madeira island. They contain descriptions of events, places (sights and businesses), transportation facilities and social activities, collected from numerous static, near- and real-time local and global data providers, including Expo Milano 2015 official services in the case of Milan, and numerous social media platforms. The generation of each city-driven 3cixty KB follows a strict data integration pipeline, that ranges from the definition of the data model, the selection of the primary sources used to populate the knowledge base, till the data reconciliation used for generating the final stream of cleaned data that

[5] The source code is available at https://github.com/rifat963/KBDataObservatory
[6] http://linkedevents.org/ontology/Event
[7] http://www.ontologydesignpatterns.org/ont/dul/DUL.owl
[8] https://www.3cixty.com

is then presented to the users via multi-platform user interfaces. The quality of the data is today enforced through a continuous integration system that only verifies the integrity of the data semantics [10].

3 Knowledge Base Evolution Analysis

The evolution of a KB can be analyzed using fine-grained "change" detection at low-level or using "dynamics" of a dataset at high-level. Fine-grained changes of KB sources are analyzed with regard to their sets of triples, set of entities, or schema signatures [5]. For example, fine-grained analysis at the triple level between two snapshots of a KB can detect which triples from the previous snapshots have been preserved in the later snapshots. Furthermore, it can detect which triples have been deleted, or which ones have been added. On the other hand, the dynamic feature of a dataset give insights into how it behaves and evolves over a certain period [5]. Ellefi *et al.* [1] explored the dynamic features considering the use cases presented by Käfer *et al.* [2].

KB evolution analysis using dynamic feature help to understand the changes applied to an entire KB or parts of it. It has multiple dimensions regarding the dataset update behavior, such as frequency of change, changes pattern, changes impact and causes of change. More specifically, using dynamicity of a dataset, we can capture those changes that happen often; or changes that the curator wants to highlight because they are useful or interesting for a specific domain or application; or changes that indicate an abnormal situation or type of evolution [6, 5]. The kind of evolution that a KB is subjected to depends on several factors such as:

- *Frequency of update:* KBs can be updated almost continuously (e.g. daily or weekly) or at long intervals (e.g. yearly);
- *Domain area:* depending on the specific domain, updates can be minor or substantial. For instance, social data is likely to be subject to wide fluctuations than encyclopedic data, which are likely to undergo smaller knowledge increments;
- *Data acquisition:* the process used to acquire the data to be stored in the KB and the characteristics of the sources may influence the evolution; for instance, updates on individual resources cause minor changes when compared to a complete reorganization of a data source infrastructure such as a change of the domain name;
- *Link between data sources:* when multiple sources are used for building a KB, the alignment and compatibility of such sources affect the overall KB evolution. The differences of KBs have been proved to play a crucial role in various curation tasks such as the synchronization of autonomously developed KB versions, or the visualization of the evolution history of a KB [6] for more user-friendly change management.

Taking into account above mentioned factors, the benefit of KB evolution analysis can be two-fold [3]: (1) quality control and maintenance; and (2) data

exploitation. Considering quality control and maintenance, KB evolution can help to identify common issues such as broken links or URI changes that create inconsistencies in the dataset. On the other hand, data exploitation can provide valuable insights regarding dynamics of the data, domains, and the communities that explore operational aspects of evolution analysis [3].

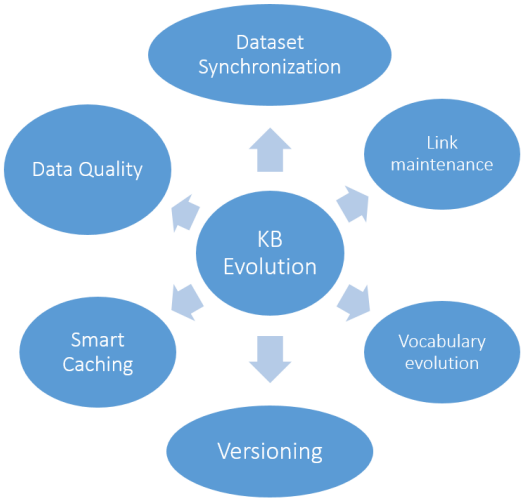


Fig. 1: Use cases of Knowledge Base evolution analysis.

Figure 1 illustrates common use cases [2] of knowledge base evolution using dynamic features. The use cases are explained in detail below.

- *Dataset Synchronization:* In any KB, large quantity of data need to be replicated and maintained at external sources. Furthermore, these data sources need to be in periodic synchronization with the original data sources [2].
- *Link maintenance:* In KB, data are made of statements that link between resources. Due to KB updates, often resources are erroneously removed or change semantics, without taking necessary steps to update their dependents resources. This creates the need to take appropriate action for link maintenance.
- *Vocabulary evolution:* Ontologies, vocabularies, and data schemata in a KB are often inconsistent and lack metadata information. Mihindukulasooriya *et al.* [4] present an empirical study of ontology evolution and data quality. They explicitly pointed out that changes in the ontology depend on the development process and on the community involved in the creation of the knowledge base. Furthermore, they also pointed the drawbacks of finding practical guidelines and best practices for ontology evolution.
- *Versioning:* It is relevant for ontologies, vocabularies, and data schemata in a KB, whose semantics may change over time to reflect usage [2]. Within

this context, KB evolution analysis can show how changes propagate and help to design a stable versioning methodology.
- *Smart Cashing:* Query optimization and live querying approaches need a smart cashing approach for dereferencing and sources discovery. KB evolution analysis can help to identify which sources can be cached to save time and resources, how long cached data can be expected to remain valid, and whether there are dependencies in the cache [2].
- *Data Quality:* One of the key use case is to ensure a good quality of data in a KB. Since data instances are often derived from autonomous, evolving, and increasingly large data providers, it is impractical to do manual data curation, and at the same time, it is very challenging to do the continuous automatic assessment of data quality. In this context, using the KB evolution analysis, we can explore the data quality issues in tourism domain.

Based on Ellefi *et al.* [1], we present the key dynamic features for KB evolution analysis.

- *Lifespan*: knowledge bases contain information about different real-world objects or concepts commonly referred as entities. Lifespan measures change patterns of a knowledge base. Change patterns help to understand the existence and kinds of categories of updates or change behavior. Also, lifespan represents the period when a certain entity is available.
- *Update history*: it contains basic measurement elements regarding the knowledge base update behavior such as frequency of change. The frequency of change measures the update frequency of KB resources. For example, the instance count of an entity type for various versions.
- *Stability*: it helps to understand to what extent the performed update impacts the overall state of the knowledge base. Furthermore, the degree of changes helps to understand what are the causes for change triggers as well as the propagation effects.

4 Stability Characteristics

On the basis of the dynamic feature [1], a further conjecture poses that the growth of the knowledge in a mature KB ought to be stable. We define this KB growth measure as *stability characteristic*. A simple interpretation of the stability of a KB is monitoring the dynamics of knowledge base changes. This measure could be useful to understand high-level changes by analyzing KB growth patterns. Within this context, this measure explores two main areas: (1) evolution of resources and (2) impact of the erroneous removal of resources in a KB.

We argue that quality issues can be identified through monitoring lifespan of an RDF KBs. We can measure growth level of KB resources (instances) by measuring changes presented in different releases. In particular, knowledge base growth can be measured by detecting the changes over KB releases utilizing trend analysis such as the use of simple linear regression. Based on the comparison between observed and predicted values, we can detect the trend in the

KB resources, thus detecting anomalies over KB releases if the resources have a downward trend over the releases.

We derive KB lifespan analysis regarding change patterns over time. To measure the KB growth, we applied linear regression analysis of entity counts over KB releases. In the regression analysis, we checked the latest release to measure the normalized distance between an actual and a predicted value. In particular, in the linear regression we used entity count (y_i) as dependent variable and time period (t_i) as independent variable. Here, $n = total\ number\ of\ KB\ releases$ and $i = 1...n$ present as the time period.

We start with a linear regression fitting the count measure of the class (C):

$$y = at + b$$

The residual can be defined as:

$$residual_i(C) = a \cdot t_i + b - count_i(C)$$

We define the normalized distance as:

$$ND(C) = \frac{residual_n(C)}{mean(|residual_i(C)|)}$$

Based on the normalized distance, we can measure the KB growth of a class C as:

$$Stability(C) = \begin{cases} 1\ if ND(C) \geq 1 \\ 0\ if ND(C) < 1 \end{cases}$$

The value is 1 if the normalized distance between actual value is higher than the predicted value of type C, otherwise it is 0. In particular, if the KB growth measure has the value of 1 then the KB may have an unexpected growth with unwanted entities otherwise the KB remains stable.

5 Experimental Analysis

Experimental Settings: The 3cixty KB is continuously changing with frequent updates (daily updates). We target *lode:Event* and *dul:Places* class for Stability analysis. The distinct instance count for each class is presented in Table 1a. We manually collected 9 snapshots from 2016-03-11 to 2016-09-09. In addition, we collected daily snapshots for *lode:Event* type starting from 2017-07-19 till 2017-09-27. Table 1b reports the entity count of *lode:Event* type using periodic snapshots generation.

Stability Characteristics: We applied a linear regression over the eight releases for the *lode:Event*-type and *dul:Place*-type entities (Figure 2a and 2b).

From the linear regression, the KB has a total of $n = 8$ releases where the 8^{th} predicted value for *lode:Event* $y^{'}_{event_8} = 3511.548$ while the actual value=689. Similarly, for *dul:Place* $y^{'}_{place_8} = 47941.57$ and the actual value=44968.

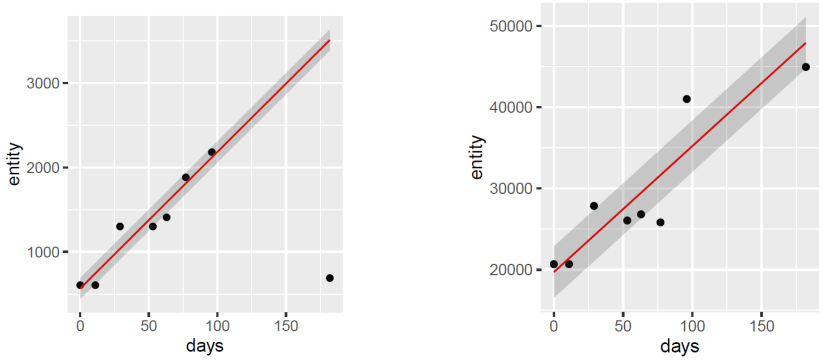
Table 1: 3cixty KB Dataset Summary.

(a) *lode:Event* and *dul:Place* type.

Release	lode:Event	dul:Places
2016-03-11	605	20,692
2016-03-22	605	20,692
2016-04-09	1,301	27,858
2016-05-03	1,301	26,066
2016-05-13	1,409	26,827
2016-05-27	1,883	25,828
2016-06-15	2,182	41,018
2016-09-09	689	44,968

(b) Periodic snapshots of *lode:Event* class.

Release	Entity Count
2017-07-27	114,054
2017-07-28	114,542
2017-07-29	114,544
2017-07-30	114,544
other rows are omitted for brevity	
2017-09-14	188,967
2017-09-15	192,116
2017-09-16	154,745



(a) lode:Event (b) dul:Place

Fig. 2: 3cixty two classes Stability measure.

The residuals, e_{events_8}= $|689 - 3511.548| = 2822.545$ and e_{places_8}= $|44968 - 49741.57| = 2973.566$. The mean of the residuals, $e_{event_i} = 125.1784$ and $e_{place_i} = 3159.551$, where $i = 1...n$.

So the normalized distance for, 8^{th} *lode:Event* entity $ND_{event} = \frac{2822.545}{125.1784} = 22.54818$ and *dul:Place* entity $ND_{place} = \frac{2973.566}{3159.551} = 0.9411357$.

For the *lode:Event* class, $ND_{events} \geq 1$ so the stability measure value = 1. However, for the *dul:Place* class, $ND_{places} < 1$ so the stability measure value =0 .

In the case of the *lode:Event* class, it clearly presents anomalies as the number of distinct entities drops significantly on the last release. To further validate our assumption we performed manual inspection on the last release of *lode:Event* entity type. We observed that entities that are present in 2016-06-06 are missing in 2016-09-09. Thus, it leads to a Stability Characteristics value of 1. We further investigated the value chain leading to the generation of the KB and we found an error in the reconciliation algorithm for 2016-09-09 release.

In Figure 2a, the *lode:Event* class growth remains constant until it has errors in the last release. It has higher distance between actual and predicted value based on the *lode:Event*-type entity count. However, in the case of *dul:Place*-type, the actual entity count in the last release is near the predicted value. We can assume that on the last release, the 3cixty KB has improved the quality of data generation matching the expected growth. Figure 3 illustrates the stability measures for *lode:Event* entity type periodic snapshots. The last three snapshots (2017-09-14,2017-09-15,2017-09-16) stability measures has a value of 1 which indicates an exponential growth compared to predicted values.

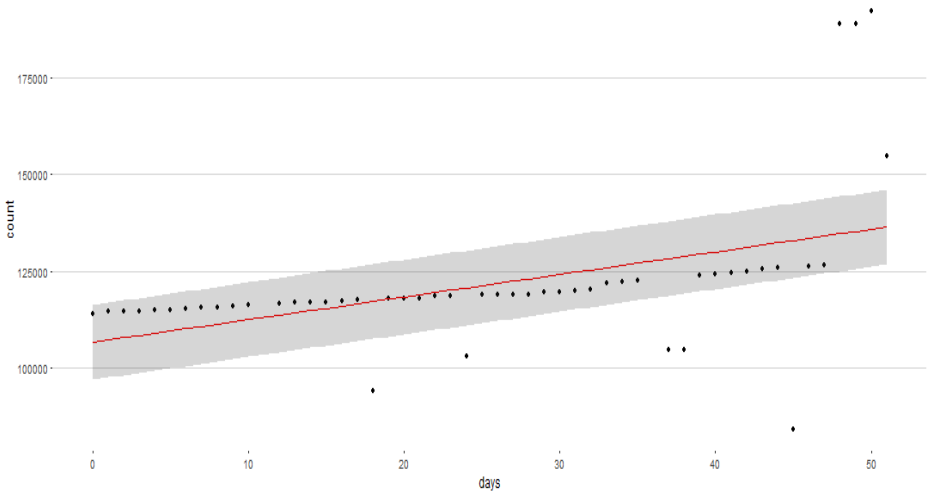


Fig. 3: KB Stability measure for 3cixty *lode:Event* class using periodic snapshots dataset.

6 Related Work

Taking into account changes over time, every dataset can be dynamic. In this context, Käfer *et al.* [2] design a Linked Data Observatory to monitor linked data dynamics. Umbrich *et al.* [11] present a comparative analysis on LOD datasets dynamics. In particular, they analyzed entity dynamics using a labeled directed graph based on LOD, where a node is an entity that is represented by a subject. In addition, Umbrich *et al.* [12] present a comprehensive survey based on technical solutions for dealing with changes in datasets of the Web of Data. Issues in curated RDF(S) have been addressed by Papavasileiou et al. [6]. They introduce a high-level language of changes and its formal detection and application semantics, as well as a corresponding change detection algorithm, which

satisfies these needs for RDF(S) KBs. Ellefi *et al.* [1] present a comprehensive overview of the RDF dataset profiling feature, methods, tools, and vocabularies. They present dataset profiling in a taxonomy and illustrate the links between the dataset profiling and feature extraction approaches. It enables easy and efficient navigation among versions, automated processing, and analysis of changes. They also include cross-snapshot queries (spanning across different versions), as well as queries involving both changes in schema and instance. Zabilith et al. [13] conducted an extensive work at the ontology level detection, representation, and management of the changes.

Pernelle et al .[7] present an approach that allows to detect and represent elementary and complex changes that can be detected only on the data level. In this work, we use linear regression for detecting changes present in the KB. However, instead of using a clustering technique [5] based on entities temporal pattern we mainly focus on presenting linear regression analysis to detect trend present in the KB. Clustering techniques can be of help to summarize the temporal changes in a dataset, but they are computationally expensive considering millions of entities present in a KB. In this regard, only using data profiling results as features we reduce the computational complexity of the task because we reduce the volume of data to process.

7 Conclusions and Future Work

We have focused on the use case of supporting tourist data producers and consumers using KB evolution analysis in their activities of data collection and integration process. Knowledge about Linked Data dynamics[9] is essential for a broad range of applications such as effective caching, link maintenance, and versioning [2]. However, less focus has been given towards understanding knowledge base resource changes over time to detect anomalies over various releases in the tourism domain. More specifically, we consider coarse-grained analysis as an essential requirement to capture any inconsistency present in the dataset. Although coarse-grained analysis cannot detect all possible inconsistencies, it helps to identify common issues such as erroneous deletion of resources in the data extraction and integration processes.

In this context, the focus of this work is to automate the timely process of KB change detection without user intervention based on evolution analysis. We have designed a set of APIs for monitoring KB evolution. More specifically, we explore the lifespan of an entity type using stability characteristics using simple linear regression model. In particular, it can help to detect unexpected growth or impact of the erroneous removal of resources in a KB. To verify our assumption, we discovered entities with anomalies in the 3cixty KB and perform further inspection. For *lode:Event* entities, we identified a large number of instances missing due to an algorithmic error in the data extraction pipeline. However, a further exploration of the KB evolution analysis is needed, and we consider this

[9] https://www.w3.org/wiki/DatasetDynamics

as a future research activity. We want to explore further *(i)* which factors are affecting KB growth and *(ii)* validating the stability measure.

References

1. Ellefi, M.B., Bellahsene, Z., Breslin, J., Demidova, E., Dietze, S., Szymanski, J., Todorov, K.: Rdf dataset profiling-a survey of features, methods, vocabularies and applications. Semantic Web (2017)
2. Käfer, T., Abdelrahman, A., Umbrich, J., O'Byrne, P., Hogan, A.: Observing linked data dynamics. In: Extended Semantic Web Conference. pp. 213–227. Springer (2013)
3. Meimaris, M., Papastefanatos, G., Pateritsas, C., Galani, T., Stavrakas, Y.: A framework for managing evolving information resources on the data web. arXiv preprint arXiv:1504.06451 (2015)
4. Mihindukulasooriya, N., Poveda-Villalón, M., García-Castro, R., Gómez-Pérez, A.: Collaborative ontology evolution and data quality-an empirical analysis. In: OWL: Experiences and Directions–Reasoner Evaluation, pp. 95–114. Springer (2016)
5. Nishioka, C., Scherp, A.: Information-theoretic analysis of entity dynamics on the linked open data cloud. In: PROFILES@ ESWC (2016)
6. Papavasileiou, V., Flouris, G., Fundulaki, I., Kotzinos, D., Christophides, V.: High-level change detection in rdf (s) kbs. ACM Transactions on Database Systems (TODS) **38**(1), 1 (2013)
7. Pernelle, N., Saïs, F., Mercier, D., Thuraisamy, S.: Rdf data evolution: efficient detection and semantic representation of changes. In: Semantic Systems-SEMANTiCS2016. pp. 4–pages (2016)
8. Rizzo, G., et. al.: 3cixty@expo milano 2015: Enabling visitors to explore a smart city. In: 14^th International Semantic Web Conference (ISWC), Semantic Web Challenge (2015)
9. Sabou, M., Braşoveanu, A.M., Arsal, I.: Supporting tourism decision making with linked data. In: Proceedings of the 8th International Conference on Semantic Systems. pp. 201–204. ACM (2012)
10. Troncy, R., Rizzo, G., Jameson, A., Corcho, O., Plu, J., Palumbo, E., Hermida, J.C.B., Spirescu, A., Kuhn, K.D., Barbu, C., et al.: 3cixty: Building comprehensive knowledge bases for city exploration. Web Semantics: Science, Services and Agents on the World Wide Web (2017)
11. Umbrich, J., Decker, S., Hausenblas, M., Polleres, A., Hogan, A.: Towards dataset dynamics: Change frequency of linked open data sources (2010)
12. Umbrich, J., Villazón-Terrazas, B., Hausenblas, M.: Dataset dynamics compendium: A comparative study (2010)
13. Zablith, F., Antoniou, G., d'Aquin, M., Flouris, G., Kondylakis, H., Motta, E., Plexousakis, D., Sabou, M.: Ontology evolution: a process-centric survey. The knowledge engineering review **30**(1), 45–75 (2015)