Temporal Patterns and Periodicity of Entity Dynamics in the Linked Open Data Cloud

Chifumi Nishioka
ZBW – Leibniz Information Centre for Economics, Germany
Kiel University, Germany
chni@informatik.uni-kiel.de

Ansgar Scherp
ZBW – Leibniz Information Centre for Economics, Germany
Kiel University, Germany
a.scherp@zbw.eu

ABSTRACT

We present initial results of finding temporal patterns of entity dynamics on the Linked Open Data (LOD) cloud. For the analysis, we use the dataset of the three-year observation of the Dynamic Linked Data Observatory. Using k-means++ clustering with Euclidean distance, we reveal the temporal patterns of entity dynamics. In addition, we conduct the first investigation of periodicity in entity dynamics on the LOD cloud. While a large portion of entities are static, a certain number of entities have a temporal pattern with substantial changes. We observe different periodicity with respect to temporal patterns of entity dynamics. Knowing about the temporal patterns and their periodicity is important for applications that are depending on fresh data caches and indices of the distributed LOD cloud. They can concentrate in crawling and refreshing those parts of the LOD cloud, which are a) known to have changes in the past and b) currently have their highest periodical change rate.

1. INTRODUCTION

The Linked Open Data (LOD) cloud is a global information space to represent and connect data. The LOD cloud stores information about different real-world objects or concepts commonly referred to as *entities*, and relations between them. The LOD cloud has been expanding continuously and covers a wide range of domains [7]. Because of this evolution of LOD, it is important to understand the change behavior of the LOD cloud over time for many applications involving data caching and indexing of distributed data sources. In addition, since the most popular type of search queries contains entities [11] and recent search engines allow users to explore entities (e.g., Google Knowledge Graph), it is important to understand temporal dynamics of entities, in order to keep information of entities up to date. There are only few investigations of temporal dynamics of entities in the LOD cloud [6, 2]. However, no work so far has looked into temporal patterns of dynamics and the periodicity of how entities change on the cloud.

K-CAP 2015, October 07 - 10, 2015, Palisades, NY, USA

ISBN 978-1-4503-3849-3/15/10...$15.00

DOI: http://dx.doi.org/10.1145/2815833.2816948

In this paper, we take the initial step to investigate temporal patterns of entity dynamics on the LOD cloud. We discover dominant temporal patterns like Yang et al. [10] did for tweets, blogs, and news articles. As a dataset, we analyze the DyLDO dataset [6], a collection of weekly snapshots of a larger set of LOD documents for the period of three years. We first quantify changes of entities between two successive snapshots using a change metric based on Jaccard distance, like Dividino et al. [2]. Temporal dynamics of each entity is represented as a temporal sequence, where each element is a change value between two successive snapshots computed by the change metric. Subsequently, we cluster the obtained temporal sequences, using the k-means++ clustering algorithm [1] and employing Euclidean distance, which is known to be efficient and with a high accuracy [9]. After optimizing the number of clusters, we observe eight temporal patterns of entity dynamics. Most entities have no change over time or have slight changes. On the other hand, a certain number of entities have a temporal pattern with substantial changes.

2. RELATED WORK

Temporal Changes and Dynamics of LOD. Käfer et al. [6] provided a comprehensive analysis of the changes of LOD based on monitoring 86,696 LOD documents for 29 weeks. They report that the data changes frequently, but found that the schemata are generally static. In contrast to Käfer et al.'s analysis on the LOD cloud, Dividino et al. [2] attempted to grasp the temporal dynamics of contexts. They report that the use of the schemata for describing the entity changes quite a lot. Thus, while the schemata themselves are stable [6], the entities' signature is dynamic [2]. Umbrich et al. [8] formed a labeled directed graph based on LOD, where a node is an entity and analyzed entity dynamics. They applied k-means clustering to group entities with similar dynamics. However, they did not take into account the amount of changes of entities and considered only whether there was a change or not. In addition, they did not optimize the number of clusters and did not provide insights into any temporal patterns nor periodicity of the patterns.

Finding temporal patterns. In contrast to our analysis on LOD, Yang et al. [10] investigated patterns of temporal variation of social media, using the K-Spectral Centroid (K-SC) clustering. Their analysis found six temporal patterns of attention of online content in tweets, blogs, and news articles. In addition, they focused on clustering only sequences of social media activities that have a steep rise and fall. Although there are a lot of clustering algorithms developed for temporal sequences such as K-SC, we employ

Table 1: Symbol Notation

t	a point in time
G_t	a set of quads captured at t (a snapshot at t)
$\mathbb{G}$	the set of all G_t (our dataset)
U	a set of all possible URIs
B	a set of all possible blank nodes
L	a set of all possible literals
s	a subject, $s \in U \cup B$
p	a predicate, $p \in U$
o	an object, $o \in U \cup B \cup L$
c	a context of a triple (s, p, o), $c \in U$
(s, p, o, c)	a quad

k-means++ clustering with Euclidean distance as Wang et al. [9] reported it as scalable with a high accuracy. We leave the use of more sophisticated algorithms for future work.

3. PROBLEM DEFINITION

Let G_t be a LOD snapshot, i.e., a set of all quads captured at a certain point in time t. The set of all snapshots is $\mathbb{G} = \{G_{t_1}, G_{t_2}, \ldots, G_{t_N}\}$, where N denotes the number of snapshots. A quad $(s, p, o, c) \in G_t$ consists of a RDF triple where s, p, and o correspond to a subject, predicate, and object, and a context c, i.e., the URI on the web where the RDF triple was retrieved. Formally, we consider the sets of all possible URIs U, blank nodes B, and literals L. In a quad (s, p, o, c), the subject $s \in U \cup B$ is a URI or a blank node, the predicate $p \in U$ a URI, the object $o \in U \cup B \cup L$ a URI, a blank node, or a literal, and the context $c \in U$ a URI. An entity is a real-world object or concept. We consider all unique subject URIs s appearing in G_t as a set of entities observed at a point in time t. An entity s at a point in time t is described by a set of quads $G_t(s) ::= \{(s, p, o, c) \mid (s, p, o, c) \in G_t\}$. Thus, an entity s consists of the quads described by the outgoing properties (including those for $p =$ `rdf:type`). Table 1 summarizes the symbol notations.

In the following, we aim at finding temporal patterns of entity dynamics, by analyzing how $G_t(s)$ varies over time.

4. METHODOLOGY

First, we define a function for computing the entity dynamics. Subsequently, we describe how we find patterns and periodicity of the entity dynamics.

4.1 Entity Dynamics

We define a function δ that quantifies the degree of dynamics of entities between two points in time and outputs a value between 0 and 1. Following Dividino et al. [2], we first measure the degree of *changes* of an entity s between two points in time t_1 and t_2 as shown in Equation 1.

$$\delta(s, t_1, t_2) ::= 1 - \frac{|G_{t_1}(s) \cap G_{t_2}(s)|}{|G_{t_1}(s) \cup G_{t_2}(s)|} \quad (1)$$

Table 2 shows a small example of LOD snapshots. The change of the entity `db:Green_Village` between t_1 and t_2 is computed as $\delta($`db:Green_Village`$, t_1, t_2) = 0.66$.

In order to represent the entity dynamics (cf. [2]), we define a temporal sequence $\Delta(s)$ of entity changes as shown in Equation 2.

$$\Delta(s) ::= \big(\delta(s, t_1, t_2), \delta(s, t_2, t_3), \ldots, \delta(s, t_{N-1}, t_N)\big) \quad (2)$$

For example, the temporal sequence of the entity `db:John_Brown` is defined as $\Delta($`db:John_Brown`$) = (1.00, 0.50)$.

Table 2: An example of snapshots

a dataset at time t_1		
db:Green_Village	rdf:type	db:City
db:Green_Village	db:population	224123
a dataset at time t_2		
db:John_Brown	rdf:type	foaf:Person
db:John_Brown	db:location	db:Green_Village
db:John_Brown	db:works	db:Green_Institute
db:Green_Village	rdf:type	db:City
db:Green_Village	db:population	223768
a dataset at time t_3		
db:John_Brown	rdf:type	foaf:Person
db:John_Brown	db:location	db:Green_Village
db:John_Brown	db:works	db:Green_University
db:Green_Village	rdf:type	db:City
db:Green_Village	db:population	223540

4.2 Temporal Patterns of Entity Dynamics

Clustering. We aim at finding temporal patterns of entity dynamics and cluster temporal sequences $\Delta(s)$ of an entity s as defined in Equation 2. To this end, we use the k-means++ clustering algorithm [1] with Euclidean distance as this combination is a good trade-off between efficiency and accuracy of the results [9]. We interpret the centroids of the resulted clusters as dominant temporal patterns of entity dynamics.

Optimization of the Number of Clusters. We find the optimal number of clusters k by applying an adopted cluster quality score from Dutta et al. [3]. The score considers two factors, intra-cluster and inter-cluster sparseness. The cluster quality score CQ is defined as Equation 3.

$$CQ = \sum_{c_l \in C} \frac{comp(c_l)}{iso(C)}, \quad (3)$$

where $comp(c_l)$ computes the compactness of a cluster defined in Equation 4 below:

$$comp(c_l) = \max\{dist(\Delta(s_i), \Delta(s_j)\}, \forall s_i, s_j \in c_l \quad (4)$$

It computes the distance of arbitrary pairs of temporal sequences of entities $\Delta(s_i)$ and $\Delta(s_j)$ in the cluster c_l and returns the maximum distance between two entities observed in c_l. We employ the Euclidean distance as we do for clustering. Furthermore, $iso(C)$ computes the minimum distance of any two entities from different clusters in C. It is defined in Equation 5 below:

$$iso(C) = \min\{dist(\Delta(s_i), \Delta(s_j))\}, \forall s_i \in c_l; \forall s_j \in c_m; c_l \neq c_m \quad (5)$$

Ideally, every cluster should contain similar elements (i.e., low *comp*) and the distance between elements from different clusters should be high (i.e., high *iso*). Thus, a lower CQ indicates a better clustering.

Periodicity Detection from Temporal Patterns. Periodicity detection is the task to discover the pattern at which a temporal sequence is periodic. For instance, temporal sequences $(1, 3, 2, 1, 3, 2)$ and $(1, 2, 1, 2, 1, 2)$ have the periodicity 3 and 2, respectively. Computing periodicity over the observed entity dynamics ensures generalizability of the observed temporal patterns. We employ a convolution-based

Table 3: Statistics of the DyLDO dataset. The bottom row provides the average number of quads per snapshot and the standard deviation.

# of snapshots	165
# of contexts	2,230
# of quads per snapshot	610,824.50 (± 14,745.10)

algorithm proposed by Elfeky et al. [4]. The algorithm outputs periodicity candidates with confidence scores.

5. EXPERIMENT AND RESULT

We first describe the dataset used for the experiment. Subsequently, we show the temporal patterns following the methodology described above and discuss the results.

5.1 Dataset

We use the Dynamic Linked Data Observatory (DyLDO) dataset[1]. The DyLDO dataset has been created to monitor a fixed set of contexts. The dataset is composed of 165 weekly snapshots over three years from May 2012 to July 2015. For more detailed information about the dataset, we refer to [6]. For the sake of consistency and scalability, we first identify the subset of 2,230 contexts that appear in all snapshots as shown in Table 3. The contexts originate from 163 pay-level domains (PLDs). A PLD is a sub-domain of a public top-level domain, for which users usually pay. They are extracted from the data using the Guava tool [2]. From the 2,230 contexts observed over all snapshots, we extract on average 610,824.50 quads (SD: 14,745.10) per snapshot. We use those subsets of the original snapshots in our experiment. Please note, for reasons of readability we continue using the term snapshots in the following when referring to the subsets of the original snapshots.

5.2 Result and Discussion

Entities in DyLDO. We find a total of 1,091,737 entities from all the snapshots, i. e., we form the union of all entities over all snapshots. 65,044 entities (5.96%) appear in all snapshots. For the sake of consistency and scalability for clustering, we focus on analyzing the temporal dynamics of the 65,044 entities. Those entities come from 696 PLDs, i. e., many entities span multiple PLDs.

Temporal Patterns of Entity Dynamics. From the 65,044 entities that appear in all snapshots, we observe that 50,275 entities (77.29%) do not change once over the entire period of time. Thus, we concentrate on clustering the remaining 14,769 entities. We run the k-means++ clustering, varying the number of clusters k from 2 to 10, and set the number of iterations to 1,000,000. Using the cluster quality score defined in Equation 3, we find that $k = 7$ is the optimal number of clusters. Figure 1 plots centroids of each of the seven clusters. The interpretation of the clustering output is such that each cluster represents a characteristic temporal pattern.

We show how many entities belong to each cluster in Table 4. C_s denotes a set of entities which have no change over all snapshots. In addition, the top three candidates of

[1]http://swse.deri.org/dyldo/, last access on 07/22/2015
[2]https://github.com/google/guava/wiki/Release19, last access on 08/24/2015

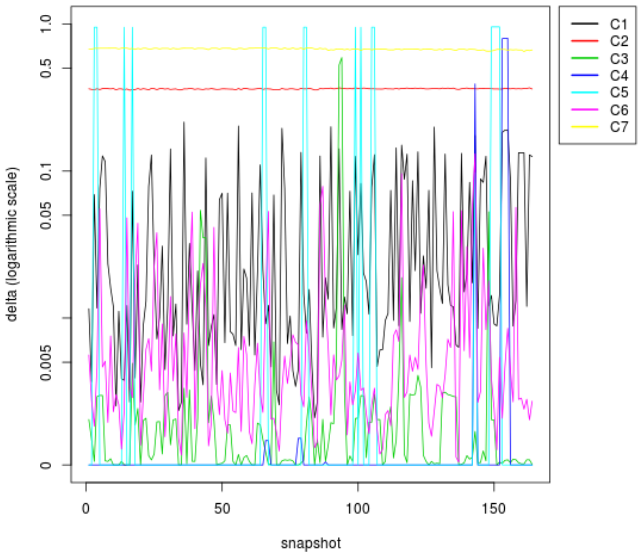

Figure 1: Temporal patterns of entity dynamics.

the periodicity are given with the ratios compared to the confidence score of the top candidate, which represent how different the second and third candidate are from the top candidate in terms of confidence scores. We observe that most entities belong to $C1$, where entities are changed periodically. Although the top periodicity candidate is 66 in Table 4, the confidence score of the second one 8 is close to the top one. The confidence score indicates how likely a certain computed periodicity is the actual periodicity of the temporal sequence. Thus, we can say that entities in $C1$ repeat a change behavior at every 66 weeks. Similarly, the second largest cluster $C6$ has periodical changes, but the degree of changes is smaller than one of the cluster $C1$. The clusters $C2$ and $C7$ share a similar duration in terms of their temporal pattern, where entities continuously change over all snapshots. However, the amplitudes of changes are different between these clusters. A small number of entities belong to the clusters $C3$ and $C4$. Here, the entities only have a small amount of changes but these changes happen continuously at every week (periodicity is one) as shown in Table 4. However, substantial changes are also observed at some points in time as depicted in Figure 1. The cluster $C5$ can be seen as an outlier, since it only contains one entity, whose subject URI is legislation.gov.uk/uksi/2010/1158/article/2/made.

Pay-level Domains of Entities in Clusters. Finally, we investigate for the resulting clusters which pay-level domains (PLD) they cover. We provide dominant PLDs of the 65,044 entities at the first row of Table 5. A large portion of entities come from `ontologydesignpatterns.org`.

Table 5 provides dominant PLDs with respect to clusters and ratios occupied by each PLD in a cluster. We observe that the majority (more than 60%) of entities come from up to three different dominant PLDs in most clusters, even when the number of entities in a cluster is large (e.g., $C1$, C_s). For example, one can see that `ontologydesignpatterns.org` and `w3.org` are dominant in C_s. Ontology Design Patterns define ontology design patterns as generic solutions of recurring modeling problems [5], which should be by definition stable. And W3 contains RDF, RDFS, and SKOS

Table 4: Resulting clusters and the top three candidates of the periodicity of the temporal patterns. In parentheses, ratios compared to the confidence score of the top candidate are given, which represents how different the second and third candidate are from the top candidate in terms of confidence scores. A higher value indicates that the periodicity is as appropriate as the top one.

	# of entities	Top three periodicity		
		1	2	3
$C1$	12,982	66 (1.00)	8 (0.99)	42 (0.96)
$C2$	168	23 (1.00)	18 (0.99)	24 (0.99)
$C3$	35	1 (1.00)	51 (0.21)	50 (0.17)
$C4$	12	1 (1.00)	2 (0.50)	12 (0.26)
$C5$	1	1 (1.00)	51 (0.62)	77 (0.56)
$C6$	1,541	56 (1.00)	48 (0.85)	27 (0.81)
$C7$	30	37 (1.00)	34 (0.99)	36 (0.99)
C_s	50,275	-		

vocabularies that hardly change, too. Entities from `legislation.gov.uk` are distributed across different clusters and dominate them. It is the official database of the statute law of the UK and each entity indicates a law. They have different patterns depending on the type of the law. We assume that the type of law is the reason why they have different dynamics and thus are distributed over the clusters. However, investigating this in more detail requires further analyses.

6. CONCLUSION

We present the initial results of analyzing temporal patterns of entity dynamics on the LOD cloud. Using the k-means++ clustering, we find seven temporal patterns of entity dynamics on a large LOD datset over a period of three years. In addition, we conduct the first investigation of periodicity in entity dynamics of the LOD cloud. The insights provided on temporal patterns and periodicity of the LOD cloud dynamics is important for applications that depend on timely updates of data caches and indexes of the LOD cloud. They can conduct targeted crawls for fresh data to those parts of the cloud that a) are known to have frequent changes and b) have a peak in their change rate according to the periodicity. In the future, we will scale up the analysis and use more elaborated clustering algorithms.

7. REFERENCES

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *SODA*, pages 1027–1035. SIAM, 2007.

[2] R. Dividino, T. Gottron, A. Scherp, and G. Gröner. From changes to dynamics: dynamics analysis of linked open data sources. In *PROFILES*, 2014.

[3] A. Dutta, C. Meilicke, and H. Stuckenschmidt. Enriching structured knowledge with open information. In *WWW*, pages 267–277. W3C, 2015.

[4] M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid. Periodicity detection in time series databases. *TKDE*, 17(7):875–887, 2005.

[5] A. Gangemi. Ontology design patterns for semantic web content. In *ISWC*, pages 262–276. Springer, 2005.

Table 5: Pay-level domains (PLDs) of entities in the clusters. PLDs shown until the sum of ratios in each cluster (except "others") exceeds 60%.

	Pay-level domains	Ratio
all	`ontologydesignpatterns.org`	40%
	`legislation.gov.uk`	11%
	`w3.org`	9%
	others	40%
$C1$	`legislation.gov.uk`	30%
	`utexas.edu`	28%
	`data.gov.uk`	20%
	others	22%
$C2$	`openei.org`	42%
	`legislation.gov.uk`	35%
	others	23%
$C3$	`legislation.gov.uk`	49%
	`kit.edu`	29%
	others	22%
$C4$	`legislation.gov.uk`	100%
$C5$	`legislation.gov.uk`	100%
$C6$	`legislation.gov.uk`	23%
	`bbc.co.uk`	21%
	`ivan-herman.net`	11%
	`ntu.ac.uk`	6%
	`reegle.info`	5%
	others	34%
$C7$	`vivoweb.org`	13%
	`kanzaki.com`	10%
	`fao.org`	10%
	`utexas.edu`	10%
	`qudt.org`	10%
	`buzzfeed.com`	10%
	others	37%
C_s	`ontologydesignpatterns.org`	52%
	`w3.org`	11%
	others	37%

[6] T. Käfer, A. Abdelrahman, J. Umbrich, P. O'Byrne, and A. Hogan. Observing linked data dynamics. In *ESWC*, pages 213–227. Springer, 2013.

[7] M. Schmachtenberg, C. Bizer, and H. Paulheim. Adoption of the linked data best practices in different topical domains. In *ISWC*, pages 245–260. Springer, 2014.

[8] J. Umbrich, M. Karnstedt, and S. Land. Towards understanding the changing web: Mining the dynamics of linked-data sources and entities. In *KDML*, 2010.

[9] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, 2013.

[10] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, pages 177–186. ACM, 2011.

[11] X. Yin and S. Shah. Building taxonomy of web search intents for name entity queries. In *WWW*, pages 1001–1010. ACM, 2010.