A Framework for the Automatic Adaptation of RDF-based Semantic Annotations

Enio de Jesus Pontes Monteiro[0000−0001−9992−864X] and Julio Cesar dos Reis[0000−0002−9545−2098]

Institute of Computing, University of Campinas, Campinas, SP, Brazil
eniojpmonteiro@hotmail.com, jreis@ic.unicamp.br

Abstract. Access and use of semantically defined metadata based on RDF repositories can benefit several types of computational tasks. However, RDF triples tend to undergo modifications as new releases of the repositories appear, which implies a challenging scenario for RDF-based generated annotations. In this context, existing annotations need to be updated according to the evolution of undergoing knowledge base used for their definitions. In this paper, we propose an adaptation framework for updating semantic annotations defined from structured RDF data. Our adaptation approach relies on modifications detected in the evolution of RDF knowledge bases. We design and formalize adaptation operations which are applied to update annotation states. We present and formalize the framework and discuss existing open challenges in our research task.

Keywords: Metadata · RDF · Ontology · Semantic Annotations · LOD

1 Introduction

The generation of metadata (data about data) on Web documents, videos and images using existing Resource Description Framework (RDF) knowledge bases plays a key role to computational systems. This type of metadata is called semantic annotations [1] and it consists of RDF resources that make the meaning of Web elements interpretable to machines. Semantic annotations are essential elements to help systems better interpret, integrate, and retrieve information considering the explicit meaning shared by machines.

In the last years, a large number of interconnected RDF knowledge bases have emerged describing various types of resources in a structured way, e.g., Dbpedia [2]. The knowledge presented in knowledge bases described in RDF is constantly evolving. This phenomenon of the evolution of the base can directly affect the existing associated annotations (already created) since it can make them invalid.

Existing literature has presented methodologies to address this problem. The studies that address the automatic detection of inconsistent annotations [3–6] perform the identification of concepts that changed from a release j of a Knowledge Organization Systems (KOS) [7] to its release $j + 1$ and its associated set of annotations. These studies do not support the correction of outdated annotations. However, methods have

been developed to address the maintenance of outdated annotations [8–11]. Nevertheless, our literature analysis did not detect investigations addressing the maintenance of annotations considering their generation at the instance level of concepts.

In this article, we propose a framework capable of identifying and applying maintenance actions in semantic annotations affected by the evolution of RDF knowledge bases as automatically as possible. Our maintenance process comprises the execution of three steps (cf. Secion 2).

The remaining of this article is organized as follows: Section 2 presents our framework including its description and formalization; Section 3 provides a discussion on existing open challenges in our research. Finally, Section 4 draws the conclusion remarks.

2 ANNOLOD framework for annotation adaptation

The key contribution of this research consists of a framework capable of executing modifications to update annotations as automatically as possible. We propose the framework ANNOLOD to support maintaining instance annotations in RDF repositories. In our study, we adapted the annotation model proposed by Cardoso *et al.* [6] in order to consider instance-based generated annotations. We defined our model as $\mathcal{ISAM} = (D, \mathcal{O}^j, \mathcal{R}^j, \mathcal{A}, SemRel, U_f)$, such that:

- D: It consists of a set of documents $D = \{dj, ..., dn\}$.
- $\mathcal{O}^j$: is a ontology in its release j. An ontology $\mathcal{O}$ describes a domain of knowledge in terms of concepts, attributes, and relationships between concepts [13]. Formally, an ontology $\mathcal{O} = (\mathcal{C}_\mathcal{O}, \mathcal{S}_\mathcal{O}, \mathcal{P}_\mathcal{O})$ consists of a set of classes $\mathcal{C}_\mathcal{O}$ interrelated by directed relationships $\mathcal{S}_\mathcal{O}$. Each $c \in \mathcal{C}_\mathcal{O}$ concept has a unique identifier and is associated with a set of attributes $\mathcal{P}_\mathcal{O}(c) = \{a_1, a_2, a_3, ..., a_n\}$.
- $\mathcal{R}^j$: is a RDF repository in its release j with its predicates defined in the ontology $\mathcal{O}^j$. An RDF repository in the context of Linked Open Data (LOD) is a finite set of RDF triples [12]. Formally, $\mathcal{R} = (t_1, t_2, t_3, ..., t_n)$. In a RDF repository, a triple associates two nodes (or resources) using a property (predicate). A resource can be defined as an instance of a certain ontology class. In RDF, resources are described using a Uniform Resource Identifier (URI)[1] for the unique identification of resources on the Web. A RDF triple refers to a data entity composed by subject (s), predicate (p), and object (o) defined in the form of $t = (s, p, o)$.
- $\mathcal{A}$: is a set of annotations. A $a \in \mathcal{A}$ is defined as $a = (i, t, d, \mathit{Offset}, rel, p)$, such that, an entity named $i \in d \subset D$ is connected to a triple $t \in \mathcal{R}^j$; *Offset* indicates the position ($start$, end) where i appears in the document d being annotated; $rel \in SemRel$ describes the type of relationship between i and $s \in t$; $p \in U_f$ points out to the previous version of the annotation a_i to keep a tracing of the evolution of the annotation in time.

The adaptation was carried out, because the model defined in Cardoso *et al.* [6] did not have all the required elements to conceive our maintenance actions. We observed

[1] https://www.w3.org/wiki/URI

the need to add the attributes i, t, d, $Offset$, rel, and p in the definition of an annotation a. In the original model of Cardoso *et al.* [6], they were defined at set level of $\mathcal{A}$. Table 1 provides an instance annotation example by adopting our adapted model ($\mathcal{ISAM}$). The mention of the scientist "*Albert Einstein*", present in a given textual document was linked (annotated) to its semantic definition (formal RDF resource "Albert Einstein")[2] formally coded in the DBpedia.

Table 1. Annotation example based on our $\mathcal{ISAM}$ model

$\mathcal{A}$		D	$\mathcal{R}^j \rightarrow \mathcal{O}^j$			$SemRel$	U_f
a_3	i	d	t	$offset$		rel	p
				$start$	end		
	Albert Einstein	46	<Albert Einstein, rdf:type, Person>	1	15	$sameAs$	a_2

The proposed framework defines annotation adaptation actions to be performed automatically when an RDF dataset used to create semantic annotations evolves (*i.e.*, a new release is generated). These actions are necessary to keep annotations consistent and up to date over time. The necessary input consists of the interconnected initial datasets, being $\mathcal{R}^j$ and $\mathcal{R}^{j+1}$ (its new release that can affect existing annotations) and the existing in place $\mathcal{A}^j$ annotations. The ultimate goal of the framework is to obtain the updated $\mathcal{A}^{j+1}$ annotations according to the new release $\mathcal{R}^{j+1}$ dataset. The framework performs a series of steps (cf. Algorithm 1). Each step is explained in further details (cf. Steps A, B, and C).

Algorithm 1: Annotation Maintenance

Require: $\mathcal{R}^j$, $\mathcal{R}^{j+1}$, $\mathcal{A}^j$
1: $\mathcal{A}^{j+1} \leftarrow \varnothing$
2: $\Delta \leftarrow detectChanges(\mathcal{R}^j, \mathcal{R}^{j+1})$
3: $\mathcal{A}^{aff} \leftarrow recAffAnnotations(\Delta, \mathcal{A}^j)$
4: $\mathcal{A}^{unaff} \leftarrow recUnAffAnnotations(\Delta, \mathcal{A}^j)$
5: $\mathcal{A}^{j+1} \leftarrow \mathcal{A}^{j+1} \cup \mathcal{A}^{unaff}$
6: **for all** $a_i \in \mathcal{A}^{aff}$ **do**
7: $\quad \mathcal{A}^{j+1} \leftarrow \mathcal{A}^{j+1} \cup applyAction(\Delta, a_i)$
8: **end for**
9: **return** $\mathcal{A}^{j+1}$

Step A: this step consists of detecting a series of modifications that occurred in a given time period based on two releases of an RDF dataset (line 2 in Algorithm 1). This operation is known as Δ because it computes the difference between the two datasets, recognizing added, removed, and not updated elements. Changes can be of the simple type (such as unit actions of adding or removing triples), or complex operations (update actions) of the knowledge stored in the datasets.

Step B: this step consists of recognizing and filtering the annotations affected by the changes of those that are not affected (lines 3 to 4 in Algorithm 1). An annotation

[2] http://dbpedia.org/page/Albert_Einstein

can be created, removed or updated. In our solution, annotations that share a subject (s) with a t_k triple in which $t_k \in \Delta$ are considered affected by change modifications. These are considered outdated annotations and maintenance actions on the annotations must be applied to them. We assume that unaffected annotations can be directly reused in composing the final set of updated annotations and added to the final set of $\mathcal{A}^{j+1}$ annotations (line 5 in Algorithm 1). However, annotations classified as affected is further handled by our framework. This involves investigating which and how computed change operations influence the definition of existing annotations.

Step C: this step involves applying corrective actions (lines 6 to 8 in Algorithm 1) to the affected and outdated annotations detected in step B. For example, an action type may be a "reanotation". In this case, an annotation $a_i \in \mathcal{A}^j$ defined on the basis of a triple t_k adapted its subject (s). The framework generates as a final result a stable and semantically consistent set of annotations $\mathcal{A}^{j+1}$ (line 9 in Algorithm 1) concerning the updated RDF data in the new dataset $\mathcal{R}^{j+1}$.

3 Open Research Challenges

The key research challenge at step A (cf. Section 2) is to understand what kind of changes at the level of instances can affect existing annotations. For example, to what extent does the removal of a triple RDF (used in defining an annotation) impact the consistency of such an annotation?

In step B (cf. Section 2), a key challenging research refers to how to accurately categorize an annotation as inconsistent based on computed change operations. In this sense, we need to investigate to which extent semantics defined in the annotations are affected by the observed changes. For example, the removal of the associated triple may be a typical case that affects the annotation. However, if there are other types of changes related to the triple subject, it is necessary to further understand how they make the annotation semantically inconsistent.

The main challenge in step C (cf. Section 2) refers to the definition and correct application of annotation adaptation actions to ensure the updating of semantically consistent affected annotations. The definition of actions (adaptation operations) requires investigating techniques that express the necessary conditions for their application.

4 Conclusion

The real value of semantically-enabled computer systems lays on the reliability of semantic annotations. This work studied how to keep RDF-based annotations up-to-date according to the evolution of RDF repositories. We proposed a framework for the (semi-)automatic maintenance of semantic annotations affected by RDF data evolution. Our defined adaptation algorithm works on the basis of change operations automatically identified in the evolution of RDF datasets. We are currently further investigating the adaptation actions, their formalization and applicability. Next steps involve the full implementation of a software tool for the adaptation of RDF-based semantic annotations maintenance. We also plan to conduct thorough experimental analyses with real-world datasets.

Acknowledgments

This work is supported by the São Paulo Research Foundation (FAPESP) (Grants #2017/02325-5, #2019/14582-8 and #2013/08293-7)[3].

References

1. Oren, E., Möller, K., Scerri, S., Handschuh, S., Sintek, M.: What are semantic annotations?. DERI Galway, **9**, 62 (2006)
2. Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., Ives Z.: DBpedia: A Nucleus for a Web of Open Data. In: Aberer K. et al. (eds) The Semantic Web. ISWC 2007, ASWC 2007. LNCS, vol 4825, pp 722-735. Springer, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
3. Maynard, D. Peters, W., d'Aquin, M., Sabou, M.: Change management for metadata evolution. In: Proceedings of the International Workshop on Ontology Dynamics (IWOD-07), pp. 27–40. IWOD-07, Innsbruck (2007)
4. Gross A., Hartung M., Kirsten T., Rahm E.: Estimating the Quality of Ontology-Based Annotations by Considering Evolutionary Changes. In: Paton N.W., Missier P., Hedeler C. (eds) Data Integration in the Life Sciences. DILS 2009. LNCS, vol 5647, pp 71-87. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02879-3_7
5. Köpke J., Eder J.: Semantic Invalidation of Annotations Due to Ontology Evolution. In: Meersman R. et al. (eds) On the Move to Meaningful Internet Systems: OTM 2011. OTM 2011. LNCS, vol 7045, pp 763-780. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25106-1_25
6. Cardoso, S. D., Pruski, C., Da Silveira, M., Lin, Y. C., Groß, A., Rahm, E., Reynaud-Delaître, C.: Leveraging the Impact of Ontology Evolution on Semantic Annotations. In: Blomqvist E., Ciancarini P., Poggi F., Vitali F. (eds) Knowledge Engineering and Knowledge Management. EKAW 2016. LNCS, vol 10024, pp 68-82. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49004-5_5
7. Hodge, G.: Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files. 1st edn. ERIC, Washington, DC (2000)
8. Luong PH., Dieng-Kuntz R. (2006) A Rule-based Approach for Semantic Annotation Evolution in the CoSWEM System. In: Koné M.T., Lemire D. (eds) Canadian Semantic Web. Semantic Web and Beyond (Computing for Human Experience), vol 2, pp 103-120. Springer, Boston, MA . https://doi.org/10.1007/978-0-387-34347-1_7
9. Park, Y.R., Kim, J., Lee, H.W. et al.: GOChase-II: correcting semantic inconsistencies from Gene Ontology-based annotations for gene products. BMC Bioinformatics **12**, S40 (2011).
10. Cardoso, S. D., Chantal, R. D., Da Silveira, M., Pruski, C.: Combining rules, background knowledge and change patterns to maintain semantic annotations. In: AMIA Annual Symposium Proceedings, pp. 505–514. AMIA, Washington, D.C (2017)
11. Cardoso, S., Reynaud-Delaître, C., Da Silveira, M., Lin, Y. C., Gross, A., Rahm, E., Pruski, C.: Evolving semantic annotations through multiple versions of controlled medical terminologies. Health Technol **8**, 361–376 (2018)
12. Faisal S., Endris K.M., Shekarpour S., Auer S., Vidal ME.: Co-evolution of RDF Datasets. In: Bozzon A., Cudre-Maroux P., Pautasso C. (eds) Web Engineering. ICWE 2016. LNCS, vol 9671, pp 225-243. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38791-8_13
13. Gruber, T. R.: Toward principles for the design of ontologies used for knowledge sharing?. International journal of human-computer studies, **43**(5-6), 907-928 (1995)

[3] The opinions expressed in this work do not necessarily reflect those of the funding agencies.