Towards Updating Wikipedia via DBpedia Mappings and SPARQL[⋆]

Albin Ahmeti, Javier D. Fernández, Axel Polleres, and Vadim Savenkov
Vienna University of Economics and Business, Vienna, Austria
{name.surname}@wu.ac.at

Abstract. DBpedia is a community effort that has created the most important cross-domain datasets in RDF, a focal point of the Linked Open Data (LOD) cloud. In its core there is a set of declarative mappings extracting the data from Wikipedia *infoboxes* and tables into the RDF. However, while DBpedia focuses on publishing knowledge in a machine-readable way, little attention has been paid to the benefits of supporting machine updates. This greatly restricts the possibilities of automatic curation of the DBpedia data that could be semi-automatically propagated to Wikipedia, and also prevents maintainers from evaluating the impact of their edits on the consistency of knowledge. Excluding the DBpedia taxonomy from the editing cycle is a major drawback which we aim to address. This paper starts a discussion of DBpedia making a case for a benchmark for Ontology Based Data Management (OBDM). As we show, although based on fairly restricted mappings (which we cast as a variant of nested tgds here) and minimalistic TBox language, accommodating DBpedia updates is intricate from different perspectives, ranging from conceptual (what is an adequate semantics for DBpedia SPARQL updates?) to challenges related to the user interface design.

1 Formalization of the OBDM Setting

We define the declarative **WikiDBpedia framework** (WDF) as a pair $(\mathcal{M}, \mathcal{T})$ where $\mathcal{M}$ is a schema mapping between the structured Wiki data and DBpedia [8], and $\mathcal{T}$ is a DBpedia TBox. Specifically, $\mathcal{M}$ is a triple $(\mathbf{W}, \mathbf{T}, \Sigma)$ based on a set Σ of *nested tuple generating dependences* (tgds) [5, 7] of a special form translating the Wiki schema $\mathbf{W}$ into an ABox over the DBpedia vocabulary $\mathbf{T}$. A *WDF instance* of a WDF $(\mathcal{M}, \mathcal{T})$ is a Wiki instance I satisfying $\mathbf{W}$. We now specify the language used to formalize the TBox $\mathcal{T}$, the tgds language of Σ and the Wiki schema $\mathbf{W}$.

DBpedia ontology language. DBpedia uses a fragment of OWL 2 RL profile which we call DBP. The fragment includes OWL keywords subClassOf (which we abbreviate as sc) and subPropertyOf (sp), domain and range (respectively, dom and rng), inversePropertyOf (inv), disjointWith (dw), propertyDisjointWith (pdw) and functionalProperty (func). At present, only data properties are declared as functional in DBpedia and no roles are declared inverse functional. Inference rules for the ontology language DBP are summarized in Fig. 1. Application of these rules terminates and thus allows for materialization of the ABox.

Infobox schema W. Each Wiki page is identified by a URI which translates to a subject IRI in DBpedia. A page can contain several *infoboxes* of distinct *types*. We model this semistructured data store using a relational schema $\mathbf{W}$ with two ternary relations $W_i =$ UTI and $W_d =$ IPV, attribute I storing infobox identifiers, U page URI, T infobox type, and P and V being property names resp. values. That is, unlike the real Wiki where infoboxes may belong to different pages or be separate tables of distinct types,

[⋆] An extended version of this paper including additional details is available in [3].

A sc $B : A(x) \rightarrow B(x)$	P sp $Q : P(x,y) \rightarrow Q(x,y)$	P dom $A : P(x,y) \rightarrow A(x)$
P rng $A : P(x,y) \rightarrow A(y)$	P inv $Q : P(x,y) \rightarrow Q(y,x)$	A dw $B : A(x) \wedge B(x) \rightarrow \perp$
P pdw $Q : P(x,y) \ \wedge \ Q(x,y) \rightarrow \perp$		func $: P(x,y) \wedge P(x,z) \wedge y \neq z \rightarrow \perp$

Fig. 1. Rule representation of DBP.

we use an auxiliary surrogate key I to horizontally partition the single key-value store W_d. Our schema $\mathbf{W}$ assumes key constraints UT $\rightarrow$ I, IP $\rightarrow$ V and the inclusion dependency $W_d[\mathsf{I}] \subseteq W_i[\mathsf{I}]$. Two kinds of values are allowed in $\mathbf{W}$: labelled nulls and constants, whereby only constants will be transferred to the DBpedia by the mappings as explained below.

Mapping constraints Σ. The specification [1] distinguishes several types of DBpedia mappings summarized in Table 1 along with their figures in the English DBpedia. All these mappings can be represented as *nested tgds* [5, 7] extended with *negation and constraints* in the antecedents for capturing the conditional mappings and interpreted functions in the conclusions of implications, in the case of calculated mappings handling, e.g., dates or geo coordinates. A crucial limitation of the mapping language (which we call *DBpedia tgds*) is the *impossibility of comparisons between infobox property values*. Infobox type W_i.T and property names W_d.P must be specified explicitly.

For a Wiki instance I, by $\mathcal{M}(I)$ we denote the chase of I with the tgds in $\mathcal{M}$ [7] and by $\mathcal{M} \circ \mathcal{T}(I)$ the closure of $\mathcal{M}(I)$ under the rules in Fig. 1.

Example 1. A tgd formalizing a French DBpedia mapping for clergy:

$$
\begin{aligned}
&\forall U \forall I \big(W_i(U, \text{'fr:Prélat catholique'}, I) \rightarrow \\
&\quad \big(W_d(I, \text{'titre'}, \text{'Pape'}) \rightarrow \exists Y \big(\mathsf{Pope}(U) \wedge \mathsf{occupation}(U,Y) \wedge \mathsf{PersonFunction}(Y) \\
&\qquad \wedge\, \mathsf{title}(Y, \text{'Pape'})) \quad \text{// \textit{“Intermediate node mapping”}} \\
&\qquad \wedge \ldots \\
&\qquad \wedge\, \forall X (W_d(I, \text{'prédécesseur pape'}, X) \rightarrow \mathsf{predecessor}(Y, X))\big) \\
&\quad \ldots \\
&\quad \wedge\, (W_d(I, \text{'titre'}, \text{'Prêtre'}) \rightarrow \mathsf{Priest}(U)) \\
&\quad \wedge\, (\neg W_d(I, \text{'titre'}, \text{'Pape'}) \wedge \ldots \wedge \neg W_d(I, \text{'titre'}, \text{'Prêtre'}) \rightarrow \mathsf{Cleric}(U)) \quad \text{// \textit{“otherwise”}} \\
&\quad \wedge\, \forall X (W_d(I, \text{'nom'}, X) \rightarrow \mathsf{foaf{:}name}(U, X)) \\
&\quad \ldots \\
&\quad \wedge\, \forall X (W_d(I, \text{'nom naissance'}, X) \rightarrow \mathsf{birthName}(U, X)))\big)
\end{aligned}
$$

The specification stipulates that conditions are evaluated in the natural order, and thus every next condition has to include the negation of all preceding conditions. In our case, this is only illustrated by the last, default ("otherwise") case, since the conditions are mutually exclusive. Note also that no universally quantified variable besides the page URI U and the technical infobox identifier I) – i.e., no variable representing an infobox property, called X in the example – can occur in more than two W_d atoms.

One further particularity of the chase with tgds is handling of existentially quantified variables. A usual approach is to instantiate such variables by null values, which could be blank nodes in the case of RDF. The strategy followed by DBpedia is however different: instead of blank nodes, the chase produces fresh IRIs, avoiding clashes with existing page URIs. Already the following problem is worst-case intractable for WDFs:

ABox source consistency ASCONS [2, 6].Parameter: WDF $(\mathcal{M}, \mathcal{T})$. Input: ABox A. Test if $A \cup \mathcal{T} \not\models \perp$ and if a Wiki instance I exists such that $\mathcal{M} \circ \mathcal{T}(I) = A$.

TYPE OF MAPPINGS	DECLARED	DESCRIPTION
Template	958	Map Wiki templates to DBpedia classes.
Property	19,972	Map Wiki template properties to DBpedia properties.
IntermediateNode	107	Generate a blank node with a URI.
Conditional	31	Depend on template properties and their values.
Calculate	23	Compute a function over two properties.
Date	106	Mappings that generate a starting and ending date.

Table 1. Description of DBpedia (English) mappings.

Proposition 1. ASCONS *is NP-complete.*[1]

2 Towards the DBpedia OBDM

The ABox source consistency problem demonstrates one source of complexity for DBpedia update translations, namely accommodating a set of insertions exactly (up to the facts derivable via a TBox).

Definition 1 (Translation of an infobox update). *Let I be a Wiki instance, $e = (e^-, e^+)$ be an infobox update and let $\mathcal{M}$ be a DBpedia mapping. The* translation $\mathcal{M}_I(e)$ of e w.r.t. $\mathcal{M}$ and *I is a DBpedia update $u = (u^-, u^+)$ where $u^- = \mathcal{M} \circ \mathcal{T}(I) \setminus \mathcal{M} \circ \mathcal{T}(e(I))$ and $u^+ = \mathcal{M} \circ \mathcal{T}(e(I)) \setminus \mathcal{M} \circ \mathcal{T}(I)$.*

The inverse translation, casting a DBpedia update as a Wiki update, can be defined similarly, with the difference that such a translation is often not unique or even not existing, for various reasons: (i) many-to-many relations between Wiki and RDF properties: modifying just a single fact can be impossible (ii) updates can cause inconsistencies as directly w.r.t. the previous DBpedia knowledge, as also indirectly, by triggering a conditional mapping rule, causing already existing infobox properties to be transfered to DBpedia, resulting in a clash. Therefore, we define translations based on containment.

Definition 2 (Update containment). *The* syntactic containment $u_1 \subseteq u_2$ *holds when $u_1^+ \subseteq u_2^+$ and $u_1^- \subseteq u_2^-$ is the case. This containment is applicable to pairs of Wiki updates. Given an instance I of a WDF $(\mathcal{M}, \mathcal{T})$ the* WDF containment $u \subseteq_I e$ *between the Wiki update e and the DBpedia update u holds if $u \subseteq \mathcal{M}_I(e)$. The proper update containment relations $\subset$ and $\subset_I$ are defined analogously.*

For the heterogeneous pair u, e of updates as above, we say that e minimally contains u, *written $u \subseteq_{I_{min}} e$, if* (i) $e \not\models_I \bot$ *and* (ii) *for every Wiki update e' with $e' \subset e$, $u \not\subseteq_I e'$ or $e' \models_I \bot$ is the case; if $e' \subset e$ implies $u \not\subseteq_I e'$ (that is, the option $e' \models_I \bot$ is eliminated), e is said to* faithfully contain u, *written $u \subseteq_{I_{fth}} e$. We also use $u \subseteq_{I_{ex}} e$ and $u =_I e$ as shorthands for $(u \subseteq \mathcal{M}_I(e)) \wedge (\mathcal{M}_I(e) \subseteq u)$.*

Intuitively, minimal containment ensures that all insertions and deletions performed by e are necessary either to implement u or to restore the ABox consistency after implementing u. In contrast, faithful containment deprecates extending u purely for the sake of restoring the consistency. The notions of minimal and faithful adapt the semantics considered in [4] in a much simpler setting of SPARQL ABox udpates, where no mappings have been present.

Using the above definition, the decision version of the OBDM [9] problem can be defined as follows:

Source revision SREV for the WDF $(\mathcal{M}, \mathcal{T})$ and $\diamond \in \{min, fth, ex\}$. Input: WDF instance I, DBpedia update u, Wiki update e. Test if $u \subseteq_{I_\diamond} e$ holds.

[1] See [3] for a proof sketch.

The source revision problem is a special case of belief revision problem tailored to the OBDM setting, in which the mapping and the TBox are considered fixed and the ABox is derived: that is, only the infobox data can be actually modified.

3 Discussion and Practical Outlook

OBDM related problems tend to be intractable w.r.t. the worst case complexity even for simple mapping and ontology languages, such as those underlying DBpedia. Our initial experiments with the translation of SPARQL updates in this setting (discussed in [3]) demonstrate however, that worst-case scenarios leading to intractability of update handling are seldom realized in the current DBpedia version. From a practical point of view, the following considerations appear crucial. First, it is the *inherent ambiguity of update translation*; mappings often create a many-to-one or many-to-many relationships between infobox and DBpedia properties. Second, concisely presenting a large number of options to the user is a challenge, hence an *automatic selection* resp. *ranking of update translations* is required. The crucial part of these services is to provide the user with the clear and concise justifications for the ranking or automatic selection, based on the already present data or previously resolved updates. Finally, being a curated system, Wiki also requires curated updates. Thus, *splitting a SPARQL update into small independent pieces* to be verified by Wiki maintainers is needed as well.

Little attention has been paid so far to the benefits that the semantic infrastructure can bring to maintain the wiki content. In fact, the DBpedia mapping language has to the best of our knowledge never formalized as a rule language, which this paper does. Our early practical experiments with a DBpedia-based OBDM prototype show that likely not the worst case complexity of update translation is a major challenge in such a system, but defining a reasonable DBpedia-enabled maintenance process, comprehensible user interface, and automatic aid in resolving ambiguities due to the robust design of DBpedia mappings.

Acknowledgements. This paper is funded by the Austrian Science Fund (FWF): M1720-G11, and by the Vienna Science and Technology Fund (WWTF), project ICT12-SEE.

References

1. DBpedia Mappings. http://mappings.dbpedia.org/, 2015. [accessed 29.02.15].
2. S. Abiteboul and O. Duschka. Complexity of answering queries using materialized views. In *Proc. PODS '98*, pp.254–263, ACM, 1998.
3. A. Ahmeti, J. Fernández, A. Polleres, and V. Savenkov. Towards Updating Wikipedia via DBpedia Mappings and SPARQL. Working Papers on Information Systems, Information Business and Operations, 01/2016. WU Vienna University of Economics and Business. Available at http://epub.wu.ac.at/view/p_series/S1
4. A. Ahmeti, D. Calvanese, A. Polleres, and V. Savenkov. Handling inconsistencies due to class disjointness in SPARQL updates. In *Proc. ESWC '16. to appear.*, Springer, 2016.
5. A. Fuxman, M. Hernández, C. Howard Ho, R. Miller, P. Papotti, and L. Popa. Nested mappings: Schema mapping reloaded. In *VLDB*, pages 67–78, 2006.
6. G. Grahne, A. Moallemi, and A. Onet. Recovering exchanged data. In *PODS '15*, 2015.
7. Ph. Kolaitis, R. Pichler, E. Sallinger, and V. Savenkov. Nested dependencies: structure and reasoning. In *PODS'14*, pp 176–187, ACM, 2014.
8. J. Lehmann, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.
9. M. Lenzerini. Ontology-based data management. In *Proc. CIKM '11*, pp.5–6, ACM, 2011.