DORIS: Discovering Ontological Relations In Services

Maria Koutraki[1,2], Dan Vodislav[2], and Nicoleta Preda[1]

[1] PRiSM CNRS, University of Versailles, Versailles, France
`firstname.lastname@prism.uvsq.fr`
[2] ETIS CNRS, University of Cergy-Pontoise, Cergy-Pontoise, France
`firstname.lastname@u-cergy.fr`

Abstract. We propose to demonstrate DORIS, a system that maps the schema of a Web Service automatically to the schema of a knowledge base. Given only the input type and the URL of the Web Service, DORIS executes a few probing calls, and deduces an intensional description of the Web service. In addition, she computes an XSLT transformation function that can transform a Web Service call result in XML to RDF facts in the target schema. Users will be able to play with DORIS, and to see how real-world Web Services can be mapped to large knowledge bases of the Semantic Web.

1 Introduction

Recent years have seen the rise of RDF knowledge bases (KBs) that are automatically extracted from the Web. These KBs contain facts such as which artist sang which song, which city is located in which country, or which actor stars in which movie. However, due to their automated construction, many KBs are inherently incomplete. One way to deal with this is to tap into additional sources, such as e.g. Web Services (WSs). In this work we focus on WSs that export their data via REST APIs. The `programmableweb.com` Web site (the Web's de-facto repository of WSs) records more than 12,000 WS APIs, of which more than 70% are based on the REST architecture. The large number of important data providers in the list (e.g. Amazon, Google, musicbrainz) shows that the concept is accepted by the industry. The exported data covers a large variety of domains pertinent to the construction of KBs: books, music, movies, geographic databases, transportation networks, social media.

A WS is essentially a parametrized query that is executed on a remote data source. The service is called by accessing a special URL. Figure 1 shows the description of a WS published by `isbndb.com`. Given an author, the service returns information such as his books and his awards. The author is specified as an *input value* as part of the URL. The class *author* is called the *input type*.

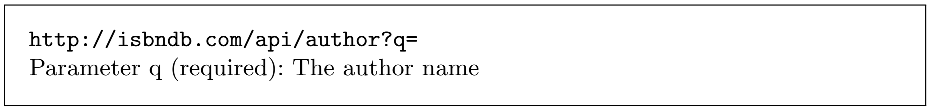

`http://isbndb.com/api/author?q=`
Parameter q (required): The author name

Fig. 1: An informal description of a REST Web Service on a Web page

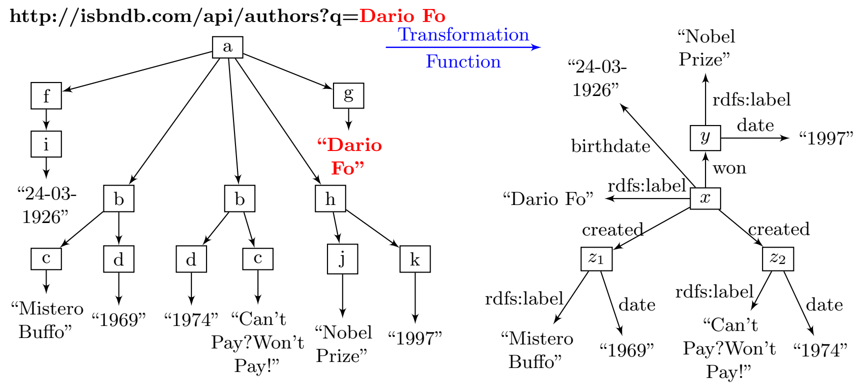


Fig. 2: A call result (left), and its mapping to the KB (right).

If this URL is accessed, the WS will respond with the result of the query, typically with an XML document. In the example, if we call the service with the input value "Dario Fo", we may receive as answer the XML tree depicted in Figure 2 (left).

In this way the providers can keep control of their data, while users and applications can receive answers to their knowledge requests. This *partial openness* may be the reason for the success of WSs. If a user issues too many requests, she is blocked. One consequence of this design is that it is impossible to crawl the data of the WS exhaustively.

Challenges. The main challenge when dealing with WSs is that the data that the service returns is not necessarily in the schema of the KB. In fact, even different WSs typically use different schemas. The goal is, therefore, to map the output of a WS call (Figure 2 (left)) into facts in the schema of the KB (Figure 2 (right)). This involves not just an alignment of the schema of the WS with the schema of the KB, but also a transformation function, which translates an XML tree into an RDF graph.

This is difficult for several reasons: First, labels of intermediary nodes in the call results are usually vacuous and do not give away any semantics. In the example, they are just labeled "a", "b", and "c". Second, it is not clear which nodes in the call result correspond to entities in the KB schema. In the example, one can guess that the nodes labeled with "b" correspond to entities. However, the nodes labeled with "f" correspond to nothing in particular. Third, call results usually contain data for several types of entities. The key challenge is to detect which node corresponds to which entity. Finally, the edges of the XML tree have to be mapped to relation names in the schema of the KB. This is particularly difficult because sometimes entire paths in the XML tree correspond to one single relation in the schema of the KB.

Contribution. In this paper, we demonstrate DORIS [2], an approach that, given a KB and a WS, deduces both the schema mapping and the transformation function automatically. As input, DORIS requires only the KB, the WS URL, and the input type. The central idea is to probe the WS with a few sample inputs from the KB, and to analyze the overlap of the XML call result with the KB in order to deduce the alignments. Technically speaking, DORIS describes the WS as a view with binding patterns over a global database schema [1]. The particular contribution of this demo proposal is the graphical interface that lets users play with the system, and map real-world WSs automatically to some of the largest KBs of the Semantic Web.

Related Work. Schema alignment approaches such as [3] could be considered to map WSs to KBs. However, they fail because the XML result tree of the WS does not give away which nodes correspond to entities in the KB and which don't. Worse, our scenario requires aligning multi-hop relations with single-hop relations, which is out of the scope of current schema alignment approaches. Closest to our work, [4] derives intensional descriptions for WSs. However, the approach is semi-automatic, and requires the user's assistance during the process. Furthermore, it assumes an implicit translation of call results into tables, meaning that a WS returns properties for only one class of entities. Our approach, in contrast, can deal with the general case where WSs return nested descriptions of entities of different types, such as authors, books, and book editions.

2 Demonstration

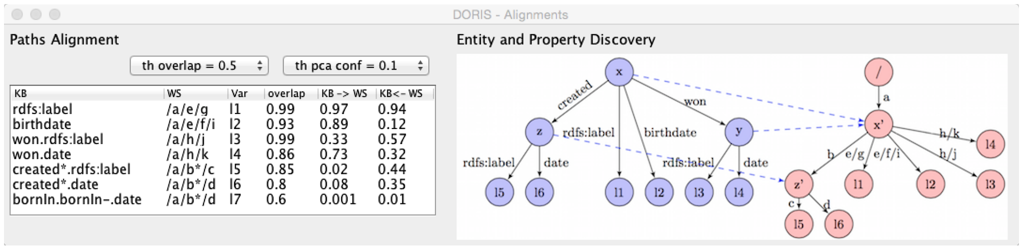
We now describe our approach DORIS. With the proposed demonstration, the user will be able to trace every step that DORIS takes in our graphical interface. The interface will illustrate a suite of strategies that we have investigated, and the design decisions we made.

The user first chooses a target KB. Our demo currently supports DBpedia, YAGO, and the KB of the French National Library BNF. Then, the user chooses a WS that she wishes to map to the KB. This can be any WS that the user came across (as the one depicted in Figure 1), as long as we may expect some overlap of entities between the WS and the KB. DORIS requires as input the URL of the WS, and the input type. We provide a list of 50 example WSs, covering the domains of music, books, movie, and geolocations. The demonstration of DORIS proceeds in 4 steps.

1. **Probing:** The service is called with several entities from the KB. The result is a set of sample call results.
2. **Path Alignment:** Discover *root-to-text nodes* paths in the call results. Discover *input-to-literals* paths in the KB. Align the paths in the call results with the paths from the KB.
3. **Entity and Property Discovery:** Identify the entities from the KB and their properties that are encoded in the call results. Find the nodes, respective the paths in the call results that correspond to them.
4. **Parameterized Query & Transformation Function:** Build the view and the transformation function as a XSLT script.

The demonstration will illustrate first different binding selection strategies, which aim to minimise the number of empty call results. For this purpose, we try to learn the overlap between the WS and the KB. The interface will also allow experimenting with different numbers of calls.

The second and the third step are concerned with identifying complex objects encoded in call results. We do not make any assumption about an implicit translation from trees to RDF fragments. The intuition behind the second step is that we can align the root of a call result to the entity in the KB that was used



th overlap = 0.5
th pca conf = 0.1

KB	WS	Var	overlap	KB -> WS	KB<- WS
rdfs:label	/a/e/g	l1	0.99	0.97	0.94
birthdate	/a/e/f/i	l2	0.93	0.89	0.12
won.rdfs:label	/a/h/j	l3	0.99	0.33	0.57
won.date	/a/h/k	l4	0.86	0.73	0.32
created*.rdfs:label	/a/b*/c	l5	0.85	0.02	0.44
created*.date	/a/b*/d	l6	0.8	0.08	0.35
bornIn.bornIn-.date	/a/b*/d	l7	0.6	0.001	0.01

Fig. 3: Paths Alignment - Entity and Property Discovery.

as input value. This is because the call return data for its input (for simplicity, in this presentation, we assume that the WS has one input). Furthermore, literals are aligned to text nodes as string values are usually encoded as literals in KBs and as text nodes XML documents. Figure 3 (left) illustrates the results of the second step for our running example. The interface will present two strategies for computing the scores of the alignments: (1) measuring the number of calls for which two paths select at least a value in common; and (2) measuring the number of calls for which the set of values selected by one path is subsumed by the set of values selected by the other path.

The third step maps KB entities (RDF resources) to XML nodes. Candidates are pairs of nodes that are traversed by a pair of paths aligned in the second step. For instance, the second step aligned the XML path `/a/b/d` and the KB path `created.date`. The desired outcome is the mapping of the nodes selected by `/a/b` to entities in the range of `created` (to books). The interface will present several strategies based on the conservation of the properties of relations e.g., their functionality. Figure 3 (right) illustrates a solution for our running example. The demonstration will also show how errors caused by biased call results can be detected and corrected.

Finally, DORIS takes the mapping of the previous step and computes a view and a transformation function in the form of an XSLT script. The interface will allow to inspect and to run the script.

3 Conclusion

This paper demonstrates DORIS, a system to automatically align Web Services with Knowledge Bases. Our experiments with 50 services show that DORIS can infer alignments with a F-measure of 81%-100%. All evaluation results, as well as our full paper [2], can be found on `http://oasis.prism.uvsq.fr/`.

Acknowledgments. This work is supported by the research projects EDOP (PATRIMA LabEx) and ALODIS (PEPS FASCIDO 2015 - CNRS).

References

1. Halevy, A.: Answering queries using views – a survey. The VLDB Journal (2001)
2. Koutraki, M., Vodislav, D., Preda, N.: Deriving intensional descriptions for web services. In: CIKM (2015)
3. Suchanek, F.M., Abiteboul, S., Senellart, P.: PARIS: Probabilistic Alignment of Relations, Instances, and Schema. PVLDB (2011)
4. Taheriyan, M., Knoblock, C.A., Szekely, P.A., Ambite, J.L.: Rapidly integrating services into the linked data cloud. In: ISWC (2012)