Annotating Web Tables Through Ontology Matching

Vasilis Efthymiou[1,2], Oktie Hassanzadeh[1], Mohammad Sadoghi[1], and Mariano Rodriguez-Muro[1]

[1] IBM T.J. Watson Research Center [2] University of Crete, Greece
{vefthymi,hassanzadeh,msadoghi,mrodrig}@us.ibm.com

1 Introduction

Web tables have been proven to constitute valuable sources of information for applications, ranging from Web search, to data discovery in spreadsheet software and KB augmentation [1]. A requirement for those applications is to understand the semantics of Web tables and potentially match their contents with existing URIs in the Web of Data, a process known as Web table annotation [4].

Recent works on Web table annotation follow an iterative approach between instance- and schema-level refinements, until convergence [6, 7]. In this work, we annotate Web tables using ontology matching. As this field has solid tools and benchmarks[3], we design a framework that provides the required input to any ontology matching tool, resulting in Web table annotations. Moreover, our blocking enables even the less scalable ontology matching tools provide annotations to large-scale KBs, such as DBpedia. The contributions of our work are:

– We introduce a generic and scalable framework for Web table annotation using existing ontology alignment systems.
– We evaluate our framework and compare the results against state-of-the-art Web table annotation tools, with promising results.
– Our framework can be extended as a benchmark for ontology matching tools.

2 Matching Framework

Model. We assume that each table row describes a real-world entity, and each column represents a property. Each cell of the *header row* defines the name of a property, except the cell of the *label column*, which defines the name of the table's class. All the entities in the table are instances of this class. The values of a column can be either literals, or references to other entities, corresponding to dataype, or object properties, respectively. To make this distinction, we sample the data types of each column, also identifying the label column, as in [6]. In a second scan, we create a new instance of the table class for each row, whose property values are the cell contents of this row for the respective column.

Blocking. To enable ontology matching tools that do not scale well be applicable in this framework, and to improve the efficiency of matching tools that do scale, we have applied a pre-processing step of candidate mappings selection, known as blocking [2]. Specifically, we retain from DBpedia, the target ontology, only those instances whose labels match with the labels of our table's instances.

[3] http://oaei.ontologymatching.org/

Finally, we call an ontology matching tool with the table ontology and the DBpedia ontology after blocking, as input, and return the mapping results.

Evaluation. We evaluate our approach using the instance mappings of the T2D gold standard[4] and LogMap [5], one of the most efficient ontology matching tools [3]. Our MapReduce-based framework annotates and evaluates the whole corpus in less than 4 minutes. Table 1 presents the micro-averaged recall, precision, and F-measure results, against T2K [6] and two baselines: **DBpedia lookup.** For each entity label in our table, we use top-1 DBpedia lookup[5] result as annotation. **DBpedia lookup refined.** We keep the type of the top-1 lookup result for each cell in a first scan of the table, and then the top-5 most frequent types for each column as acceptable types. Then, we perform a second lookup, restricting the results to the acceptable types, and use the top-1 result as the annotation.

Table 1. Results over T2D gold standard. Blocking results in parentheses.

Method	**Recall**	**Precision**	**F-measure**
DBpedia lookup	0.73	0.79	0.76
DBpedia lookup refined	0.76	0.86	0.81
T2K	0.76	0.90	0.82
Ontology matching	0.57 (0.71)	0.89 (0.32)	0.70 (0.44)

The results show that our framework, using LogMap, suggests a good number of correct results, with high precision. In the future, we plan to improve blocking and extend our model to provide a first alignment, which can be utilized by many ontology matching tools. Our goal is to provide an ontology matching benchmark for instance-, class- and property-mappings, that can result in a new track in the upcoming OAEI campaigns.

References

1. S. Balakrishnan, A. Y. Halevy, B. Harb, H. Lee, J. Madhavan, A. Rostamizadeh, W. Shen, K. Wilder, F. Wu, and C. Yu. Applying WebTables in Practice. In *CIDR*, 2015.
2. V. Christophides, V. Efthymiou, and K. Stefanidis. *Entity Resolution in the Web of Data*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool Publishers, 2015.
3. E. Daskalaki, G. Flouris, I. Fundulaki, and T. Saveta. Instance matching benchmarks in the era of linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 39:1–14, 2016.
4. O. Hassanzadeh, M. J. Ward, M. Rodriguez-Muro, and K. Srinivas. Understanding a large corpus of web tables through matching with knowledge bases: an empirical study. In *ISWC*, pages 25–34, 2015.
5. E. Jiménez-Ruiz and B. Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *ISWC*, pages 273–288, 2011.
6. D. Ritze, O. Lehmberg, and C. Bizer. Matching HTML tables to dbpedia. In *WIMS*, pages 10:1–10:6, 2015.
7. Z. Zhang. Towards efficient and effective semantic table interpretation. In *ISWC*, pages 487–502, 2014.

[4] http://webdatacommons.org/webtables/goldstandard.html
[5] http://wiki.dbpedia.org/projects/dbpedia-lookup