T2LD: Interpreting and Representing Tables as Linked Data[⋆]

Varish Mulwad, Tim Finin, Zareen Syed, and Anupam Joshi
University of Maryland, Baltimore County, Baltimore MD USA 21250
{varish1,finin,joshi}@cs.umbc.edu,zareensyed@gmail.com

Abstract. We describe a framework and prototype system for interpreting tables and extracting entities and relations from them, and producing a linked data representation of the table's contents. This can be used to annotate the table or to add new facts to the linked data collection.
Keywords: linked data, human language technology, entity linking

1 Introduction

Vast amounts of information is available in structured forms like spreadsheets, database relations, and tables in documents found on the Web. We describe a framework for interpreting such tables and extracting entities and relations from them. The results can be used to annotate the table or to contribute its contents to linked data (LOD) collections. The process assigns column headers to classs from an appropriate ontology, links table cells (as appropriate) to entities in a LOD collection, and identifies relations between columns and links them to ontology properties. The resulting table interpretation can be used to confirm existing facts in an LOD collection and to propose new facts to be added.

Using the table headers and the data in its rows and columns, we query existing knowledge bases (KBs) including Wikitology [1] and DBpedia to select the best class labels for each column, which is then used to identify potential entity links for table cells. A classifier selects the best entity from the list and a second classifier decides whether the evidence is strong enough to link the cell value to it. Relations in the table are discovered using the generated column classes, linked entities, and KB information bases. Our implemented prototype was evaluated using a collection of tables from Google Squared, Wikipedia and tables found on the Web.

Caferella et al. [2] estimated that the Web contains around 14.1 billion HTML tables, over 154 million containing high quality relational data. This represents a huge source of knowledge currently unavailable on the Semantic Web. There is a need for systems that can automatically generate LOD from existing sources, be it unstructured (e.g., free text), semi-structured (e.g., text embedded in forms or Wikis) or structured (e.g., data in spreadsheets and databases). Interpreting tables is a problem of interest in many areas such as databases, web systems and the Semantic Web. See [3, 4] for a comprehensive study of related work on interpreting tables and converting databases and spreadsheets into RDF.

[⋆] Research supported in part by a gift from Microsoft Research, a Fulbright fellowship, NSF award IIS-0326460 and the Human Language Technology Center of Excellence.

Existing systems for extracting knowledge from tables [5] require human intervention and do not focus on a complete interpretation of the table, nor integrating the table with linked open data cloud. This poster paper focuses on an automatic framework for generating an linked RDF which can be integrated into the LOD cloud. The eventual goal of this work is to enrich the LOD cloud by learning new facts and knowledge from tables and publishing it on the Semantic Web.

To develop an overall interpretation of a table, we assign every column header a class label from an appropriate ontology, e.g., the column with header City is assigned a class label *dbpedia-owl:City* from the DBpedia ontology. For the table in Figure 1, we link "Baltimore" to *dbpedia:Baltimore*. Numbers can be mapped as values of properties which can be associated with entities in the table. We also identify the relations implicit between columns, e.g., that *dbpedia-owl:largestCity* seems to hold between the entities denoted by cell values in the first two columns (i.e., city and state). Finally this information is represented in a N3 serialization of RDF.

city	*state*	*mayor*	*population*
Baltimore	MD	S.Dixon	640000
Philadelphia	PA	M.Nutter	1500000
Washington	DC	A.Fenty	595000
New York	NY	M.Bloomberg	8400000
Boston	MA	T.Menino	610000

Fig. 1: In simple tables column headers suggests the type of data stored in columns and cell values denote instances of that type.

2 *T2LD* Framework

Given an table as input, the *T2LD* framework [6] begins with the process of assigning a class label to every column in the table. For all the cell values in every column of the table, the algorithm for assigning class labels (see Algorithm 1 in [3]) submits a complex query to the Wikitology knowledge base to determine the type of each cell value in the column. Each class label from the set of possible class labels obtained from query results is scored. The class label with the highest score is chosen as the class label to be associated with the column. We predict class labels from four vocabularies - DBpedia Ontology, Freebase, WordNet, and Yago.

Using the class labels as additional evidence, for every table cell, the algorithm for linking table cell to entities (see Algorithm 2 in [3] for detailed algorithm), re-queries the KB. For every table cell, the KB returns the top N possible entities. For each of the top N entities, the algorithm generates a feature vector consisting of the entity's KB score, entity's Wikipedia page length, entity's page rank, the Levenshtein distance between the entity and the string in the query and the Dice score between the entity and the string. The set feature vectors for each table cell are ranked using a SVM-Rank classifier. To the highest rank feature vector from SVM rank, two more features are added - the SVM rank score of the feature vector and the difference

MAP	columns
$m = 1$	11.53%
$0 < m < 1$	69.23%
$m = 0$	19.24%

Recall	columns
$r = 1$	46.15%
$0 < r < 1$	34.61%
$r = 0$	19.24%

Fig. 2: The percentage of columns with various MAP and recall scores.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix dbpedia-owl: <http://dbpedia.org/ontology/> .
@prefix dbpprop: <http://dbpedia.org/property/> .

"City"@en is rdfs:label of dbpedia-owl:City .
"State"@en is rdfs:label of dbpedia-owl:AdminstrativeRegion .

"Baltimore"@en is rdfs:label of dbpedia:Baltimore .
dbpedia:Baltimore a dbpedia-owl:City .
"MD"@en is rdfs:label of dbpedia:Maryland .
dbpedia:Maryland a dbpedia-owl:AdministrativeRegion .

dbpprop:LargestCity rdfs:domain dbpedia-owl:AdminstrativeRegion .
dbpprop:LargestCity rdfs:range dbpedia-owl:City .
```

Fig. 3: A example of N3 representation of a table as linked data

in SVM-Rank scores between the top two feature vectors. Based on this new feature vector, a second SVM classifier decides whether to link the table cell to this top ranked entity or not. If the evidence is not strong enough, it is likely that the table cell is a new entity not present in the KB; this step is useful in discovery of new entities in a given table. If the evidence is strong enough, the table cell is linked to the top ranked entity returned by SVM-Rank.

We also present a preliminary approach for identifying relations between table columns (see Algorithm 3 in [3]). The algorithm generates a set of candidate relations from the relations that exist between the strings in each row of the two columns. Each candidate relation is scored and the relation with the highest score is selected to represent relation between the two columns. We have also developed a preliminary template in N3 (see Figure 3), which is a compact and human readable serialization of RDF for representing tables as LOD.

3 Evaluation and Conclusion

Our implemented prototype was evaluated against 15 tables obtained from Google Squared, Wikipedia and from a collection of tables extracted from the Web. Excluding the columns with numbers, the 15 tables have 52 columns and 611 entities for evaluation of our algorithms. We used a subset of 23 columns for evaluation of relation identifcation between columns.

In the first evaluation of the algorithm for assigning class labels to columns, we compared the ranked list of possible class labels generated by the system against the list of possible class labels ranked by the evaluators. As shown in Figure 2 for *80.76%* of the columns the Mean Average Precision (MAP) between the system and evaluators list is greater than 0 which indicates that there was at least one relevant label in the top three of the system ranked list. Also seen in Figure 2, for *75%* of the columns, the recall of the algorithm was greater than or equal to 0.6. We also assessed whether our predicted class labels were reasonable based on the judgment of human subjects (see [3]). *76.92* % of the class labels predicted were considered correct by the evaluators. The accuracy in

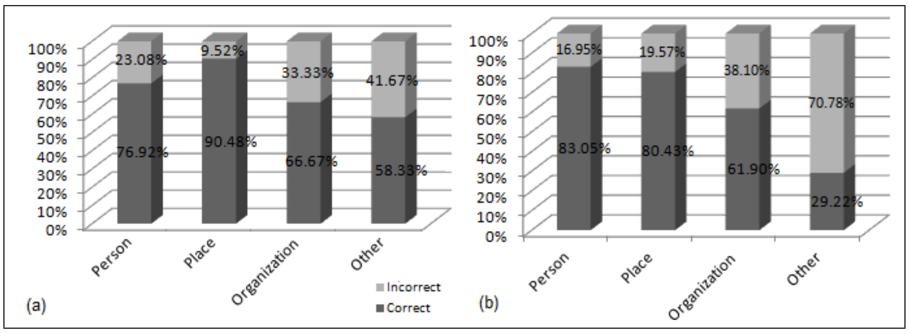

Fig. 4: Category wise accuracy for "column correctness" is shown in (a) and for entity linking in (b)

each of the four categories is shown in Figure 4. *66.12* % of the table cell strings were correctly linked by our algorithm for linking table cells. The breakdown of accuracy based on the categories is shown in Figure 4. Our dataset had 24 new entities and our algorithm was able to correctly predict for all the 24 entities as new entities not present in the KB. We did not get encouraging results for relationship identification with an accuracy of 25 % (see [3] for details).

Our existing system performs reasonably well in selecting appropriate types for columns and linking cell values to LOD entities. We have preliminary results for identifying and encoding the relationships implicit in the columns as well. Our current work is focused on improving relationship discovery and generating new facts and knowledge from tables that contain entities not present in the LOD knowledge bases.

References

1. Finin, T., Syed, Z.: Creating and Exploiting a Web of Semantic Data. In: Proc. 2nd Int. Conf. on Agents and Artificial Intelligence, Springer (2010)
2. Cafarella, M.J., Halevy, A.Y., Wang, Z.D., Wu, E., Zhang, Y.: Webtables: exploring the power of tables on the web. PVLDB **1** (2008) 538–549
3. Mulwad, V.: T2LD - An automatic framework for extracting, interpreting and representing tables as Linked Data. Master's thesis, U. of Maryalnd, Baltimore County (2010)
4. Sahoo, S.S., Halb, W., Hellmann, S., Idehen, K., Thibodeau Jr, T., Auer, S., Sequeda, J., Ezzat, A.: A survey of current approaches for mapping of relational databases to rdf. Technical report, W3C (2009)
5. Han, L., Finin, T., Parr, C., Sachs, J., Joshi, A.: RDF123: from Spreadsheets to RDF. In: Seventh International Semantic Web Conference, Springer (2008)
6. Syed, Z., Finin, T., Mulwad, V., Joshi, A.: Exploiting a Web of Semantic Data for Interpreting Tables. In: Proceedings of the Second Web Science Conference. (2010)
7. Mulwad, V., Finin, T., Syed, Z., Joshi, A.: Using linked data to interpret tables. In: Proc. First Int. Workshop on Consuming Linked Data. (2010)