MTab: Matching Tabular Data to Knowledge Graph with Probability Models

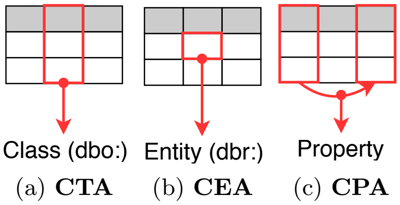
Phuc Nguyen[1,2], Natthawut Kertkeidkachorn[3],
Ryutaro Ichise[1,2,3], and Hideaki Takeda[1,2]

[1] National Institute of Informatics, Japan
[2] SOKENDAI (The Graduate University for Advanced Studies), Japan
[3] National Institute of Advanced Industrial Science and Technology, Japan

Abstract. This paper presents the design of our system, namely MTab, for Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2019). MTab combines the voting algorithm and the probability model to solve critical bottlenecks of the matching task. Results on SemTab 2019 show MTab obtains the promising performance.

1 Introduction

Tabular Data to Knowledge Graph Matching (SemTab 2019) [4] is a challenge on matching semantic tags from table elements to knowledge bases (KBs), especially DBpedia. Fig. 1 depicts the three sub-tasks for SemTab 2019. Given a table data, **CTA** (Fig. 1a) is the task of assigning a semantic type (e.g., a DBpedia class) to a column. In **CEA** (Fig. 1b), a cell is linked to an entity in KB. The relation between two columns is assigned to a property in KB in **CPA** (Fig. 1c).



(a) **CTA** (b) **CEA** (c) **CPA**

Fig. 1. Tabular Data Matching to Knowledge Base (DBpedia)

2 Approach

To address the three tasks of the challenge, we designed our system (MTab) by the 4-steps pipeline as shown in Fig. 2.

Step 1 is to pre-process a table data by predicting languages of the table with fasttext [1], correcting spelling, predicting data types (e.g., number or text), and searching relevant entities in DBpedia. Due to the heterogeneous problem, we utilize entity searching on many services including DBpedia Lookup, DBpedia

[4] http://www.cs.ox.ac.uk/isg/challenges/sem-tab/

Copyright 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

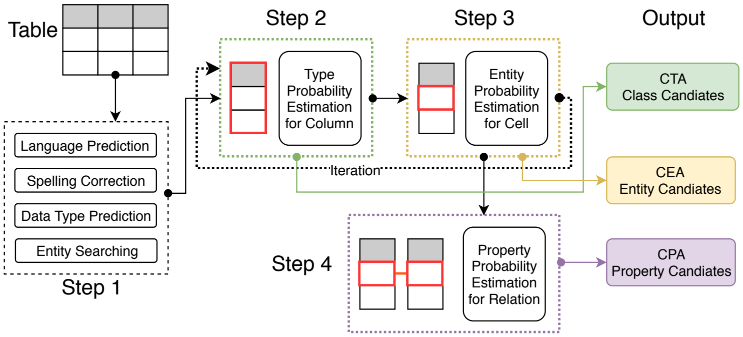


Fig. 2. The design of MTab framework

endpoint. Also, we search relevant entities on Wikipedia and Wikidata by redirected links to DBpedia to increase the possibility of finding the relevant entities. We assume that cells in a column have the same type. We then use information from Step 1 to estimate the probability of the types for the column in Step 2. The type candidate which has the highest probability is the result for **CTA** task. In Step 3, the result of Step 2 and information of Step 1 are used to estimate the probability for entities. Similarly, the entity candidate which has the highest probability is the result for **CEA** task. In Step 4, we use the result from Step 3 to estimate the property between two entities, and then, adopt the voting technique to estimate the probability for all rows of two columns. The result for **CPA** is the highest probability of property candidate in Step 4. We repeatedly execute Step 2, 3 and 4 to find the best candidates for columns, cells, and the relation between two columns.

3 Results and Conclusion

Table 1 reports the overall results of MTab for three matching tasks. Overall, these results show that MTab achieves a promising performance for the three Tabular data matching tasks. The MTab performance might be explained in part by searching cell values from multiple services to increase the possibility of finding the relevant entities, and adopting the iteration procedure to boost the overall performance for the three tasks.

Table 1. Results of MTab on Round 1 Data of SemTab 2019

Task	F1	Precision	Recall
CEA	0.816	0.799	0.834
CTA	0.934	0.926	0.942
CPA	0.594	0.698	0.516

References

1. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: EACL 2017. pp. 427–431. ACL (April 2017)