LexMa: Tabular Data to Knowledge Graph Matching using Lexical Techniques

Shalini Tyagi[1] and Ernesto Jimenez-Ruiz[1,2]

[1] City, University of London, UK
[2] University of Oslo, Norway
shaliniktyagi@gmail.com,
ernesto.jimenez-ruiz@city.ac.uk

Abstract. With the fundamentals of lives dependent upon the extensive use of the internet-based searches for common life items, there is an ever-growing demand of the quick and meaningful search query systems. This has given the rise of the concept called Semantic Web. There are many challenges in developing the Semantic Web however one fundamental challenge is to design systems to enable the semantic access to the information in tabular data (e.g., Web tables). In this paper, we discuss one such system which has been developed for the automatic annotation of the tabular data using a knowledge graph. We call this system LexMa. Our system is based on lexical matching techniques. LexMa has participated in the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2020).

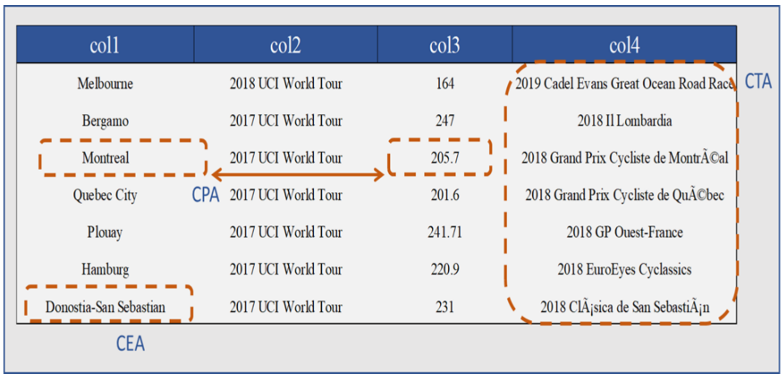
Keywords: Lexical Matching, Web Tables, Cosine Similarity, Semantic Table Interpretation.

1 Introduction

Tabular data to knowledge graph (KG) matching is the procedure of assigning the semantic tags from a KG such as Wikidata to the elements of the tables [2]. However, in the real-world data, it is hard to practice because of missing, noisy or incomplete data [3,8]. SemTab 2020: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching is a challenge for assigning semantic tags from part of the table to Wikidata KG [1]. More specifically, table annotation consists of three tasks such as cell to KG entity annotation (CEA), column to KG class annotation (CTA) and pair of columns to KG property annotation (CPA) [8]. These three tasks are summarized in Figure 1.

2 Methods

We developed the LexMa system to solve the CEA and CTA tasks using basic but efficient lexical techniques.



col1	col2	col3	col4
Melbourne	2018 UCI World Tour	164	2019 Cadel Evans Great Ocean Road Rac
Bergamo	2017 UCI World Tour	247	2018 Il Lombardia
Montreal	2017 UCI World Tour	205.7	2018 Grand Prix Cycliste de MontrÃ©al
Quebec City	2017 UCI World Tour	201.6	2018 Grand Prix Cycliste de QuÃ©bec
Plouay	2017 UCI World Tour	241.71	2018 GP Ouest-France
Hamburg	2017 UCI World Tour	220.9	2018 EuroEyes Cyclassics
Donostia-San Sebastian	2017 UCI World Tour	231	2018 ClÃ¡sica de San SebastiÃ¡n

Fig. 1. The explanation of three different tasks in the challenge.

2.1 Method used for CEA task

In SemTab 2020, the target KG is Wikidata [9,10]. The CEA task is to annotate the cells of the table to the specific entity of the Wikidata KG. The schematic of the overall pipeline used to annotate single cells is shown in Figure 2. For each of the cell values in the table, we first pre-process them by trimming the text in the cell and convert the resultant strings into uppercase. After that the top-5 entities were fetched for each cell value from the Wikidata look-up service [11]. Thereafter, the lexical matching was evaluated based upon the cosine similarity [5] of the encoded one-hot vectors formed out of the fetched entity labels and the cell value. Labels and cell values were split into tokens and stop words were removed before creating the one-hot vectors. There were still considerable numbers of cells returned with empty values as their respective entities could not be found in the Wikidata KG. These missed values were searched in the DBpedia KG via its look-up service and later converted into a (same as) Wikidata entity via the DBpedia SPARQL Endpoint [11].

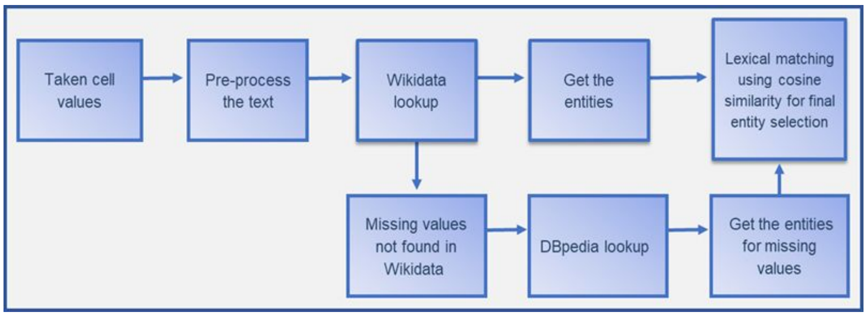

Taken cell values
Pre-process the text
Wikidata lookup
Get the entities
Lexical matching using cosine similarity for final entity selection
Missing values not found in Wikidata
DBpedia lookup
Get the entities for missing values

Fig. 2. Pipeline for CEA task

2.2 Method used for the CTA tasl

After annotating the cell values, we search the different types of each of these entities in the same column using the Wikidata SPARQL Endpoint [11]. The focus is to find the most suitable class that represents the entities in the column. For this task, we have submitted the most frequent/voted type for a column.

3 Results

3.1 Result for CEA

In Round 1 of SemTab, we focused on the CTA and CEA tasks and submitted the results for them to the challenge. We did not participate in the CPA task because our motivation was to improve CEA and CTA results. In Round 1, the CEA result is satisfactory with above 90% accuracy. LexMa holds the 8th position in the challenge (see Table 1). Our focus in the next rounds was to improve the performance in the CEA. LexMa achieved similar results and relative positions (see Table 1).

2T is the 'Tough Tables' dataset [6, 10] which was used in Round 4 together with a synthetic dataset [9] as in previous rounds. Figure 5 summarizes the performance in terms of F1-score, recall and precision for different types of tables within the 2T dataset [6]. The 2T dataset brings additional complexity to the challenge, but LexMa, unlike in the other rounds, outperformed 5 participating systems (see Table 1).

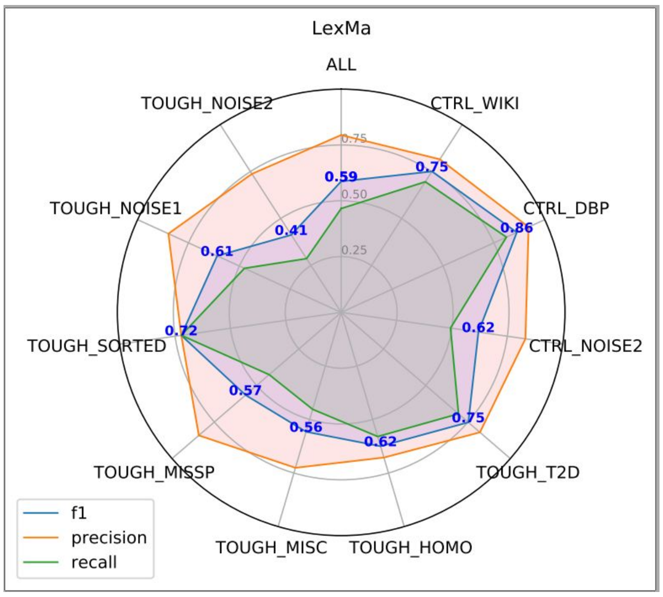


Fig. 5. Performance in Round4 2T dataset.

Table 1. Results for cell entity annotation. Official results: http://www.cs.ox.ac.uk/isg/challenges/sem-tab/2020/results.html

Task Name	F1-Score	Precision	Leaderboard position
CEA Round 1	0.909	0.913	8 out 10 competing systems
CEA Round 2	0.915	0.927	9 out 10 competing systems
CEA Round 3	0.863	0.907	9 out 9 competing systems
CEA Round 4	0.845	0.911	7 out 9 competing systems
CEA Round 4 (2T)	0.587	0.795	4 out 9 competing systems

3.2 Result for CTA

For this task, initially, LexMa got a 40.4% F1-score but after removing duplicate values and applying a ranking method, we improved our CTA result to 63.8%.

Table 2. Results for column type annotation. Official results: http://www.cs.ox.ac.uk/isg/challenges/sem-tab/2020/results.html

Task Name	Approximate F1-Score	Approximate Precision	Leaderboard position
CTA Round 1	0.638	0.734	14 out 15 competing systems

3.3 Result links

Our GitHub repository contains all the final submitted results. (https://github.com/shaliniktyagi/TabularData_to_Knowledge_graph). The code for completing the challenge is also available in GitHub repository, together with instructions about how to run the codes.

4 Discussion

Overall, this study has developed a simple approach but better results in 2T than five systems, which suggests that LexMa provides a flexible annotation system for the automatic table annotation. While there are a number of methods available, we took a rather simple but efficient approach with the use of existing technologies. Our main effort was in the pre-processing, lexical matching and parallel computing part of the challenge. In pre-processing several ideas were tried but the most effective were the selective special characters removal, duplicate words removal, white space removal and extra punctuation removal. This pre-processing improved the KG look-up efficiency and resulted in quite a good accuracy against the ground truth. We highly recommend an appropriate data conditioning upfront for the automated table annotation.

In lexical matching, using cosine similarity resulted in incremental accuracy against the ground truth. The lexical patterns could be analyzed further, and some pair-based analysis can be done. We have also tried a string length-based constraint but that did not lead to a significant improvement.

For the SemTab datasets running a job locally was not possible, in fact not only running the actual flow for look-up of the entities in the KG but to perform the data wrangling and text formatting was not very efficient while running on the local machine. A parallel processing using the Google CoLab [7] platform was a very efficient approach and reduced the turnaround time of the project significantly.

The SemTab challenge brings in a unique opportunity to learn and grow the programming skills. The pre-conditioning of the dataset and the format text editing was a rigorous task and took a multi-platform approach to achieve. All in all, the study and the entire challenge created a wide pool of research work which will be beneficial to the academic community at large.

5 Conclusion and Future Work

In this study, the aim was to annotate tabular data with the Wikidata Knowledge graph. Two tasks of the table annotation were accomplished in the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching such as CEA and CTA which has been discussed in detail above. Different techniques were used to improve the result on both tasks in Round 1 but in Rounds 2-4, the prime objective was to improve the performance of the CEA task by using different methods. In Round 4 (2T dataset), LexMa produced very promising results in comparison to other systems.

The SemTab challenge gives an engaging platform to systematically evaluate systems and lead to system improvements. Text processing and applying lexical matching with cosine similarity helped to improve a bit with 91.5% for the CEA task whereas in Round 2, the dataset had more noise in comparison to Round 1. Rounds 3 and 4 also brought additional noise and challenges. In conclusion, lexical matching techniques were able to improve performance for the CEA task to match a cell to a KG entity. Including DBpedia KG did not add a significant value in terms of overall improvement of the results; however, did improve the look up part.

In the future, we aim at improving column type annotation and cell entity annotation by using different techniques such as (pre-trained) word embedding. These techniques use a neural network model to learn word correlations within the text. The system ColNet [4], based on convolution neural networks, produced state-of-the-art results for the column type annotation. In the near future we also aim at analysing the use of CNNs to increase the accuracy of LexMa for the CEA and CTA tasks.

References

1. Malyshev, S., Krötzsch,M., González,L., Gonsior,J., and Bielefeldt,A Getting the Most out of Wikidata: Semantic Technology Usage in Wikipedia's Knowledge Graph. Wikimedia Foundation, San Francisco, U.S.A (2018)
2. Suchanek, F.H., Kasneci, G. and Weikum,G. (2007) YAGO: A core of semantic knowledge unifying WordNet and Wikipedia. In Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007
3. Cafarella,M.J., Halevy,A., Wang, Z. D., Wu.E and Zang,Y. WebTables: Exploring the Power of Tables on the Web, VLDB '08 Auckland, New Zealand.
4. Chen, J., Jiménez-Ruiz, E., Horrocks, I., Sutton, C.A. ColNet: Embedding the Semantics of Web Tables for Column Type Prediction. In: AAAI. pp. 29–36. (2019)
5. Pahi ,K., Thapa,P and Shakya,S. A Comparison of Semantic Similarity Methods for Maximum Human Interpretability, University Pulchowk Campus : Nepal (2019).
6. Vincenzo Cutrona, Federico Bianchi, Ernesto Jiménez-Ruiz and Matteo Palmonari, Tough Tables: Carefully Evaluating Entity Linking for Tabular Data. International Semantic Web Conference (ISWC). (2020)
7. Parallelism In Python, https://colab.research.google.com/drive/1nMDtWcVZCT9q1VWen5rXL8ZHVlxn2KnL, (Accessed on 21/10/2020)
8. Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen and Kavitha Srinivas. SemTab 2019: Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems. Extended Semantic Web Conference (ESWC). 2020
9. Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, Ernesto Jiménez-Ruiz, and Kavitha Srinivas. SemTab 2020: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching Data Sets (Version 2020) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.4282879
10. Vincenzo Cutrona, Federico Bianchi, Ernesto Jiménez-Ruiz and Matteo Palmonari. Tough Tables: Carefully Benchmarking Semantic Table Annotators [Data set]. Zenodo. https://doi.org/10.5281/zenodo.3840646
11. Ernesto Jiménez-Ruiz. Tabular Data Semantics for Python. https://github.com/ernestojimenezruiz/tabular-data-semantics-py