Enhancing Entity Matching Through Systematic Association of Matchers to Linking Problem Types

Chloé Khadija Jradeh[1,2], Konstantin Todorov[3] and Cassia Trojahn[2,4]

[1]University of Toulouse Capitole, 2 Rue du Doyen Gabriel Marty, 31000 Toulouse
[2]IRIT, Maison de la Recherche, 5 Allées Antonio Machado, 31058 Toulouse
[3]Université de Montpellier, 163 rue Auguste Broussonnet 34090 Montpellier
[4]University of Jean Jaurès, Campus Mirail 5, 5 Allées Antonio Machado, 31058 Toulouse

1. Introduction

Entity matching is a critical task in integrating and linking entities across different Knowledge Graphs (KGs). Each entity matching task involves a pair of KGs, and the nature of these KGs, such as their size, schema, data quality, and domain, can categorize them into different Linking Problem Types (LPTs). Selecting the most appropriate matcher for different types of LPTs can substantially enhance the accuracy and effectiveness of entity matching. This research aims to empirically evaluate matchers for each LPT and develop a framework to systematically associate matchers with specific LPTs, enhancing both accuracy and efficiency in the entity matching process.

2. Methodology

In the following, we describe the systematic approach to associating matchers with LPTs. The methodology involves three primary steps: LPT Categorization, Matcher Evaluation, and Framework Development.

LPT Categorization

The process of identifying and categorizing different LPTs have been conducted in [1] using a clustering technique. The criteria for defining LPTs include data schema compatibility, data format, and data quality metrics. In Table 1, we have selected a couple of LPTs and the respective pairs of KGs entirely belonging to these LPTs. The LPT 1.1.1.2 arises from inconsistencies in the format of predicate values, such as using different data types (e.g., strings and integers) for the same attribute. While the LPT 5.7 involves large KGs, making the matching process non-scalable.

	LPT name	KGs Pairs
LPT 1.1.1.2	Predicate value format value type	MarvelCinematicUniverse-Marvel, Memory alpha-Memory beta, Memory alpha-stex, Starwars-swg, Starwars-swtor
LPT 5.7	Graph scalability Problem	MarvelCinematicUniverse-Marvel, Starwars-swg, Starwars-swtor

Table 1: Common LPTs for the some datasets pairs.

OM-2024: The 19th International Workshop on Ontology Matching collocated with the 23rd International Semantic Web Conference (ISWC 2024), November 11th, Baltimore, USA.
khadija.jradeh@irit.fr (C. K. Jradeh); konstantin.get@gmail.com (K. Todorov); Cassia.Trojahn@irit.fr (C. Trojahn)

Matcher Evaluation

To empirically evaluate the matchers, we take the pairs of KGs associated with each LPT and assess the performance of each matcher on these pairs, calculating the average precision (prec.), recall (rec.), and F-measure (fm.) across the pairs. This helps identifying the most effective matcher for a given LPT. Table 2 shows the performance of each matcher on each KG pair of Table 1 and sums-up their average performance.

We intentionally removed the KG pair “Starwars-swtor" KG pair from the evaluation process to use it as a test case. Note that the best matching results for the pair “Starwars-swtor" was achieved using BaselineAltLabel (fm. of 0.91).

Matcher	Pair	LPT 1.1.1.2			LPT 5.7		
		Prec.	**fm.**	**Rec.**	**Prec.**	**fm.**	**Rec.**
BaseLineLabel[1]	MarvelCinematicUniverse-Marvel	0.90	0.69	0.56	0.90	0.69	0.56
	Memory alpha-Memory beta	0.95	0.85	0.77	-	-	-
	Memory alpha-stex	0.98	0.91	0.84	-	-	-
	Starwars-swg	0.95	0.67	0.52	0.95	0.67	0.52
	Average	**0.945**	**0.78**	**0.672**	**0.925**	**0.68**	**0.54**
BaseLineAltLabel[1]	MarvelCinematicUniverse-Marvel	0.86	0.76	0.68	0.86	0.76	0.68
	Memory alpha-Memory beta	0.88	0.89	0.89	-	-	-
	Memory alpha-stex	0.88	0.90	0.93	-	-	-
	Starwars-swg	0.92	0.74	0.62	0.92	0.74	0.62
	Average	**0.885**	**0.823**	**0.78**	**0.89**	**0.75**	**0.65**
LogMap [2]	MarvelCinematicUniverse-Marvel	0.84	0.60	0.46	0.84	0.60	0.46
	Memory alpha-Memory beta	0.89	0.82	0.76	-	-	-
	Memory alpha-stex	0.88	0.82	0.77	-	-	-
	Starwars-swg	0.94	0.80	0.69	0.94	0.80	0.69
	Average	**0.887**	**0.76**	**0.67**	**0.89**	**0.70**	**0.575**
LSMatch [3]	MarvelCinematicUniverse-Marvel	0.63	0.50	0.41	0.63	0.50	0.41
	Memory alpha-Memory beta	0.59	0.66	0.75	-	-	-
	Memory alpha-stex	0.53	0.63	0.80	-	-	-
	Starwars-swg	0.76	0.36	0.23	0.76	0.36	0.23
	Average	**0.628**	**0.538**	**0.548**	**0.695**	**0.43**	**0.32**

Table 2
The matchers performance on the KGs pairs belonging to LPTs 1.1.1.2 and 5.7 of Table 1. The results are sourced from the KG track of the 2023 OAEI campaign (see https://oaei.ontologymatching.org/2023/results/knowledgegraph/index.html).

Framework Development

The framework will utilize Algorithm 1 to systematically select the optimal matcher for each pair of KGs associated with specific LPTs. This process involves comparing the average performance scores of various matchers across the LPTs linked to the input KG pair and selecting the matcher with the highest score.

Example Execution of Algorithm 2 Consider the KG pair “Starwars-swtor" with LPTs 1.1.1.2 and 5.7. For these LPTs, the overall performance of each matcher is computed as:

1. **BaseLineLabel** Average Precision = 0.935, Average F-measure = 0.73, Average Recall = 0.606.
2. **BaseLineAltLabel** Average Precision = 0.8875, Average F-measure = 0.7865, Average Recall = 0.715.
3. **LogMap** Average Precision = 0.8885, Average F-measure = 0.73, Average Recall = 0.6225.

[1]These matchers utilizes respectively `rdfs:label` and `skos:altLabel` for matching entities.

Algorithm 1 Select Best Matcher for a Pair of Knowledge Graphs

```
Input: Pair of KGs (KG1,KG2), Set of LPTs LPT_set, Average Performance Scores average_performance[Matcher][LPT]
Output: Best Matcher Best_Matcher for the given KG pair
Initialize Best_Matcher to None
Initialize Best_Score to 0
for each matcher Matcher_i in average_performance do
    Initialize total_score to 0
    for each LPT LPT_j in LPT_set do
        Add average_performance[Matcher_i][LPT_j] to total_score
    end for
    Calculate average_score as total_score divided by the number of LPTs in LPT_set
    if average_score > Best_Score then
        Set Best_Matcher to Matcher_i
        Set Best_Score to average_score
    end if
end for
return Best_Matcher
```

4. **LSMatch** Average Precision = 0.6615, Average F-measure = 0.484, Average Recall = 0.434.

To determine the best matcher, we compare the average F-measure scores. In this case, **BaseLineAltLabel** has the highest average performance. This outcome aligns with the initial performance results showing the algorithm's utility in selecting the best matcher.

3. Conclusion and Future Work

This research introduces a framework that systematically aligns specific LPTs with the most appropriate entity matching algorithms. Future work will include expanding the matcher evaluation phase to incorporate new pairs of KGs associated with additional LPTs and exploring the integration of other advanced matchers.

Acknowledgment

This work is partially supported by the French National Research Agency ANR DACE-DL project, grant number ANR-21-CE23-0019.

References

[1] R. Conde Salazar, C. Jonquet, and D. Symeonidou, *Classification of Linking Problem Types for linking semantic data*, in *SEMANTICS 2023 - 9th International Conference on Semantic Systems*, Leipzig, Germany, Sep. 2023, IOS Press, https://hal.science/hal-04206689, DOI: 10.3233/SSW230014.

[2] Jiménez-Ruiz, E., & Cuenca Grau, B. (2011). LogMap: Logic-Based and Scalable Ontology Matching. In L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. Noy, & E. Blomqvist (Eds.), *The Semantic Web – ISWC 2011* (pp. 273–288). Springer Berlin Heidelberg. ISBN 978-3-642-25073-6.

[3] Sharma, A., Jain, S., & Patel, A. (2024). Large Scale Ontology Matching System (LSMatch). *Recent Advances in Computer Science and Communications*, **17**(2), 20–30. ISSN: 2666-2566.