# RE-miner for data linking results for OAEI 2020*

Armita Khajeh Nassiri[1 [0000−0002−5734−0351]], Nathalie Pernelle[1,2 [0000−0003−1487−393X]], Fatiha Saïs[1 [0000−0002−6995−2785]], and Gianluca Quercini[1 [0000−0001−9195−1618]]

[1] LRI, CNRS 8623, Paris Saclay University, Orsay F-91405, France
[2] LIPN, CNRS (UMR 7030), University Sorbonne Paris Nord, France
firstname.lastname@lri.fr

**Abstract.** This paper presents the RE-miner results for data linking in the ontology alignment contest OAEI 2020, Spimbench track. RE-miner discovers all minimal and diverse referring expressions of all instances of a given source knowledge graph. In a second step, it exploits these referring expressions to find the possible links to a target knowledge graph. This is the first participation of RE-miner in the OAEI campaign and produces the best result in terms of F-measure on the Spimbench dataset.



## 1 Presentation of the system

As the Web of Data continues to grow, more and more knowledge graphs (KGs) that cover a wide range of topics are emerging in the Linked Open Data (LOD) Cloud. As knowledge graphs are usually built independently from one another, inevitably, the same Internationalized Resource Identifier (IRI) is not necessarily reused for a given individual. Thus, it is essential to have systems capable of data linking, i.e., to produce a set of mapping between the individuals of two knowledge graphs representing the same real-world object. RE-miner for data linking is one such system that, given a subset of class and property mappings between the source and target knowledge graphs, identifies possible sameAs links between the instances of the two KGs.

### 1.1 State, purpose, general statement

RE-miner for data linking consists of 2 main steps. The algorithm has been thoroughly presented in [4]. Here, we will miss out on the details and present the major steps taken in this campaign. First, discovering referring expressions for all instances of the source knowledge graph. A referring expression (RE) is a description that identifies an instance unambiguously in a class of a knowledge graph—instantiating the keys of a class yields numerous REs itself. However, many more referring expressions can potentially be found. To reduce the search space, RE-miner focuses on non-key properties. Both keys and maximal non-keys are obtained using SAKey [5]. Second, all the REs

discovered on a class of source knowledge graph are taken into account to link to instances of a target KG. The idea behind using REs for linking is that if an instance $x$ in the target knowledge graph satisfies a description that uniquely identifies the instance $u$ in the source knowledge graph, it is probable that the two instances are the same. Using different referring expressions, an instance $u$ might be linked to different target KG instances. A voting strategy is employed to choose the most confident link whenever possible.

### 1.2 Specific techniques used

This system focuses on the instance matching problem between the instances of a given class of the source dataset, on which the REs have been discovered, and a target dataset having a non-empty set of mapped properties to the source. In other words, this approach assumes the schemas to have previously been aligned.

**Create the source dataset.** We first create the dataset on the source KG for the given class $C$, for which we aim to find the alignments. The dataset is created by keeping all instances that are of type $C$, and all sub-classes of $C$ if the graph's schema is not saturated. For instance, in the Spimbench track, the instances of *Creative Works* class are to be linked. The dataset, contains all instanced belonging to this class and its 3 sub-classes namely *NewsItem*, *BlogPost*, and *Programme*.

**Referring Expressions**. We discover all minimal and diverse referring expressions of depth 1 on the source knowledge graph [4]. These REs do not contain the existential quantifier and are conjunctions of atoms (e.g., $album(x) \wedge createdBy(x, Beatles) \wedge releasedOn(x, \text{“}1966-05-2\text{”})$ holds as a referring expression when $x$ is instantiated with Yellow Submarine). We enrich this set, with the set of referring expressions that are obtained through instantiating each set of key properties for class $C$ obtained using SAKey. Being a referring expression, each of these descriptions, holds only for one instance in the class $C$ of the source KG.

**Linking and Voting Strategy.** These REs are then used to find possible links in the target dataset. For finding the possible candidate links, mapped properties and strict equality are used between the atoms of a RE and triples of the target knowledge graph. Moreover, first consider an instance $u$ of type $C$ in the source dataset and imagine that $k$ different referring expressions $\{RE_1(u), ..., RE_k(u)\}$ have been associated to it. Each of these REs can be linked to zero, one, or more instances of the target, using the bottom-up approach explained in [4]. We consider the properties mapped if they are strictly equal in source and target.
The confidence of each RE is inverse proportional to the number of links it suggests. However, if the unique name assumption (UNA) is fulfilled, only one sameAs link can be found between $u$ and an instance $x$ belonging to the target KG. Thus we propose a voting strategy that assigns a weight to each distinct link. The weight is the sum of the confidence degree of the REs proposing that link. Moreover, the weights are normalized such that they have a value between 0 and 1. Finally, the instance $x$ in the target knowledge graph being linked to $u$ with the highest weight is selected. For the Spimbench

dataset, we have set a very strict criterion. We only match two instances if and only if the link with the highest weight has a weight equal to one. This way, we imply that we only link two instances if we are really sure about it.

**MELT.** Matching EvaLuation Toolkit (MELT) is a framework optimized for OAEI campaigns, facilitating submissions to the SEALS and HOBBIT evaluation platforms [2]. The Spimbench track, on which we evaluate our performance, is available on the HOBBIT, Holistic Benchmarking of Big Linked Data, platform[1]. We used MELT to wrap it as a HOBBIT package, and as our implementation is in Python, we used MELT's External Matching. Thankfully, MELT has eased the submission process; however, we assume that it causes some run-time overhead.

## 2 Results

### 2.1 Spimbench track

Spimbench is an instance matching track and the only track we have done evaluations on, in this first year of participation. It consists of two datasets of different sizes: the SANDBOX dataset with about 380 instances and 10000 triplets, and the MAINBOX dataset with about 1800 instances and 50000 triplets. We have compared our results with AML [1], Lily [7], FTRL_IM [6], and LogMap [3] in Table 2.1. All these systems had participated in the past year(s) of the competition.

**Table 1.** Comparison of Performance in Spimbench track. The time performance is reported in ms.

| | | Precision | Recall | F-measure | Time |
|---|---|---|---|---|---|
| SANDBOX | AML | 0.8348 | 0.8963 | 0.8645 | 6446 |
| | Lily | 0.9835 | **1.0** | 0.9917 | 2050 |
| | FTRL-IM | 0.8542 | **1.0** | 0.9214 | **1525** |
| | LogMap | 0.9382 | 0.7625 | 0.8413 | 7483 |
| | RE-miner | **1.0** | 0.9966 | **0.9983** | 7284 |
| MAINBOX | AML | 0.8385 | 0.8835 | 0.8604 | 38772 |
| | Lily | 0.9908 | **1.0** | 0.9953 | 3899 |
| | FTRL-IM | 0.8558 | 0.9980 | 0.9214 | **2247** |
| | LogMap | 0.8801 | 0.7094 | 0.7856 | 26782 |
| | RE-miner | **0.9986** | 0.9966 | **0.9976** | 33966 |

The same strategy explained in Section 1.1 is used on both datasets for RE-miner. In total, for the Sandbox dataset, 6920 REs are created. Whereas for the Mainbox dataset, there are a total of 39892 REs among which 14085 are from key instantiation. We can observe that we outperform the other systems in terms of Precision, and F-measure on both datasets, showing a slight better performance than Lily. However, we come

[1] `http://project-hobbit.eu/`

short when comparing the time-performance. This is mainly due to the fact that our system must first compute the keys and non-keys of a given class using a Java-based application, and then find the REs. Indeed more optimization can be done to decrease the run-time.

## 3 General Comments

RE-miner for data linking has shown satisfactory results in the Spimbench instance matching track. Although the source and target KGs shared almost the same ontology, there were still some properties that would not be mapped together using strict similarity. However, this did not hamper the performance of our system. This is because of the fact that RE-miner usually discovers not just one but many more REs for each instance. This will allow the system to choose the target instance most of the REs pointing to agree on. Moreover, for this dataset, we have been fastidious, only outputting links we really deem correct. As future work, we aim to do modifications, allowing us to participate in more tracks for the next years and focus more on enhancing our system's run-time.

## 4 Conclusion

In this paper, we briefly presented the main components of our instance matching system RE-miner for data linking. The evaluation of results on the Spimbench track was presented, and we showed a better Precision and F-measure than other systems taking part in the campaign this year. However, in terms of run-time, more improvement and optimization are to be done.

## References

1. Faria, D., Pesquita, C., Tervo, T., Couto, F.M., Cruz, I.F.: AML and AMLC results for OAEI 2019. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019. CEUR Workshop Proceedings, vol. 2536, pp. 101–106. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2536/oaei19_paper3.pdf
2. Hertling, S., Portisch, J., Paulheim, H.: MELT - matching evaluation toolkit. In: Semantic Systems. The Power of AI and Knowledge Graphs - 15th International Conference, SEMANTiCS 2019, Karlsruhe, Germany, September 9-12, 2019, Proceedings. pp. 231–245 (2019)
3. Jiménez-Ruiz, E.: Logmap family participation in the OAEI 2019. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019. CEUR Workshop Proceedings, vol. 2536, pp. 160–163. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2536/oaei19_paper11.pdf
4. Khajeh Nassiri, A., Pernelle, N., Saïs, F., Quercini, G.: Generating referring expressions from rdf knowledge graphs for data linking. In: Pan, J.Z., Tamma, V., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (eds.) The Semantic Web – ISWC 2020. pp. 311–329. Springer International Publishing, Cham (2020)

5. Symeonidou, D., Armant, V., Pernelle, N., Saïs, F.: Sakey: Scalable almost key discovery in rdf data. In: International Semantic Web Conference. pp. 33–49. Springer (2014)
6. Wang, X., Jiang, Y., Luo, Y., Fan, H., Jiang, H., Zhu, H., Liu, Q.: FTRLIM results for OAEI 2019. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019. CEUR Workshop Proceedings, vol. 2536, pp. 146–152. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2536/oaei19_paper9.pdf
7. Wu, J., Pan, Z., Zhang, C., Wang, P.: Lily results for OAEI 2019. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019. CEUR Workshop Proceedings, vol. 2536, pp. 153–159. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2536/oaei19_paper10.pdf