OAEI 2018 results of POMap++

Amir Laadhar[1], Faiza Ghozzi[2], Imen Megdiche[1], Franck Ravat[1], Olivier Teste[1], and Faiez Gargouri[2]

[1] University of Toulouse, IRIT (CNRS/UMR 5505) 118 Route de Narbonne 31062 Toulouse, France
{amir.laadhar,imen.megdiche,franck.ravat,olivier.teste}@irit.fr,
[2] University of Sfax, MIRACL Sakiet Ezzit 3021, Tunisie
{faiza.ghozzi,faiez.gargouri}@isims.usf.tn

Abstract. Ontology matching is the process of finding a set of correspondences between the entities of two or more ontologies representing a similar domain. POMap++ is an ontology matching system associating ontology partitioning to the machine learning techniques. This association delivers a local matching learning. POMap++ provides an automated local matching learning for the biomedical tracks. For the non-biomedical tracks we employ the version of POMap 2017. In this paper, we present POMap++ as well as the obtained results for the Ontology Alignment Evaluation Initiative of 2018.

Keywords: Semantic web, Ontology Matching, Ontology partitioning, Machine learning

1 Presentation of the system

Ontologies are the backbone of the semantic web. They enable sharing, reusing and accessing the knowledge resources [9]. Biomedical ontologies are domain-specific knowledge bases widely employed in biology and medicine. These ontologies have been separately developed by different experts using different terminologies and modeling techniques. The integration of these data sources requires ontology matching tools. Ontology matching is the identification process correspondences between the entities of different ontologies. The alignment process is quite challenging in terms of the complexity of the existing biomedical ontologies. POMap++ divide a biomedical ontology alignment to a set of sub-matching tasks called partitions. We align each sub-matching task using its local adequate settings. We automatically determine the local matching settings by generating a specific machine learning model for each sub-matching task. This automated tuning process of local matching parameters aims to improve the overall matching quality of a large ontology matching task. We employed POMap++ for the biomedical matching tasks and POMap [3] for the non-biomedical matching tasks. In the following section, we provide a detailed description of POMap++.

1.1 State, purpose, general statement

1.2 Specific techniques used

The workflow of POMap++ for our second participation in the OAEI comprises four main steps, as flagged by the figure 1: Input ontologies indexing and loading, input ontologies partitioning, local matching learning and output alignment generation. The first and the last step are the same as in the last version of POMap [3]. In the second step, we define the pair of similar partitions between the two input ontologies. In the third step, we apply machine learning techniques in order to align every identified pair of similar partitions. In the following, we detail each of the four steps.

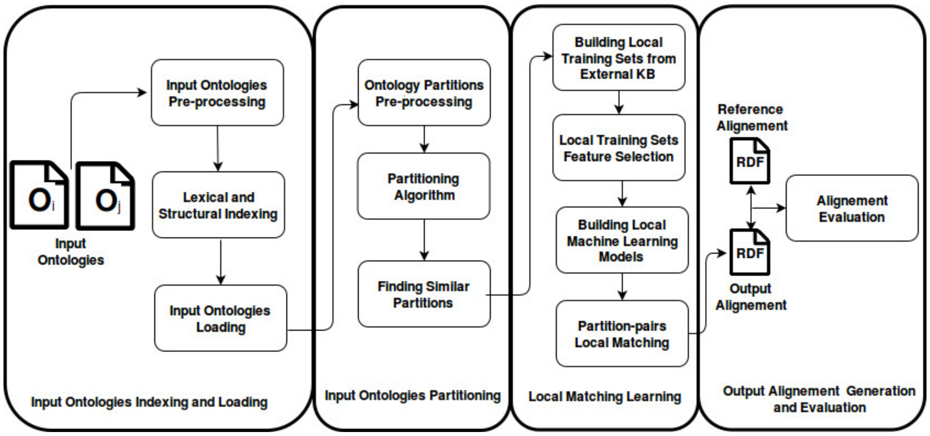
Input Ontologies Pre-processing
Lexical and Structural Indexing
Input Ontologies Loading
O_i
O_j
Input Ontologies
Input Ontologies Indexing and Loading
Ontology Partitions Pre-processing
Partitioning Algorithm
Finding Similar Partitions
Input Ontologies Partitioning
Building Local Training Sets from External KB
Local Training Sets Feature Selection
Building Local Machine Learning Models
Partition-pairs Local Matching
Local Matching Learning
Reference Alignement
RDF
Alignement Evaluation
RDF
Output Alignement
Output Alignement Generation and Evaluation

Fig. 1. The architecture of POMap++.

Step 1: Input ontologies indexing and loading

The first step of the ontology indexation and loading is the pre-processing task. We pre-process the annotations of the two input ontologies by applying the Porter stemming [8] as well as the stop word removal process. We also remove the special characters. These indexes are stored along with the structure of the input ontologies. The structural indexing is responsible for representing the relationships between entities. Then, during the third task, the indexed data structures are loaded into the next step of POMap++.

Step 2: Input ontologies partitioning

We divide an ontology into a set of partitions using the hierarchical agglomerative clustering [5] approach. This approach does not take as input the required number of partitions. The hierarchical agglomerative clustering algorithm receives as input structural similarity scores between all the entities of an input ontology. We compute the structural similarity between the entities of a single ontology according to the following Definition. The Definition 1 is inspired by Wu and Palmer [10] similarity measure.

Definition 1 (Structural similarity between entities). *We compute the structural similarity between all the entities in one ontology according to the Equation 1. For a given two entities $e_{i,x}$ and $e_{i,y}$ of an ontology O_i, lca is their lowest common ancestor. Dist($e_{i,x}$,lca) represents the shortest distance between $e_{i,x}$ and lca in terms of number of edges. Dist($e_{i,y}$,lca) denote the distance between $e_{i,y}$ and lca. Dist(r_i,lca) is the distance between the root r_i and lca.*

$$StrcSim(e_{i,x}, e_{i,y}) = \frac{Dist(r_i, lca) \times 2}{Dist(e_{i,x}, lca) + Dist(e_{i,y}, lca) + Dist(r_i, lca) \times 2} \tag{1}$$

Step 3: Local Matching learning

Due to the high complexity of biomedical ontologies, no single syntactic similarity measure can effectively all the syntactic heterogeneity of a matching task. Therefore, for each local matching task, we construct its specific machine learning model. The training set of every local learning model is not based on any reference alignments. We automatically construct a supervised training set for each local matching task of the set of local matchings. These training sets serve as the input for each local machine learning model. After identifying the partitions for each ontology, we find the set of similar partitions between the two input ontologies using a set of anchors. The existing works retrieve labeled data either from the reference alignment or by creating it manually. However, the reference alignment commonly does not exist. We derive each local training set by cross-searching the entities of a local matching with the existing biomedical knowledge bases like Uberon. Since we are dealing with biomedical ontologies, anchors are extracted by cross-searching the input ontologies with the available external biomedical knowledge bases (KB) such as the Unified Medical Language System (UMLS) Metathesaurus [1], Medical Subject Headings (MeSH) [4], Uberon [6] and BioPortal [7]. For instance, UMLS integrates more than 160 biomedical ontologies. In our case, we cross-search the two input ontologies with the Uberon ontology to derive the set anchors. We employ the-state-of-the art syntactic similarity measures[3] as features. The labeled data of the training set is usually hard to acquire. We apply the wrapper feature selection [2] method over the resulted local training sets. This technique selects the subset of the most effective and suitable features for each local training set. Therefore, each local matching task has its specific similarity measures. Then, we build a local machine learning model for each local matching task. The entities of each local matching task are classified using their specific machine learning model. This local learning model aligns the input entities based on the adequate matching parameters.

Step 4: Output alignment generation

[3] https://git.io/fNvqt

The generated correspondences for every local matching task $lm_{ij,q}$ are unified to generate the final alignment file for the whole ontology matching task. The alignment file is compared to the reference alignment to evaluate the overall result accuracy.

2 Results

2.1 Anatomy

The Anatomy track consists of finding the alignments between the Adult Mouse Anatomy and the NCI Thesaurus describing the human anatomy. The evaluation was run on a server coupled with 3.46 GHz (6 cores) and 8GB of RAM. Table 1 draws the performance of POMap++ compared to the five top matching systems. Our matching system achieved the third best result for this dataset with an F-measure of 89.7%, which is very close to the top results. The remaining challenge is to speed up the execution time by applying more optimizations. We also target the improvement of precision value for our next participation in the OAEI.

2.2 Disease and Phenotype

This track is based on a real use case in order to find alignments between disease and phenotype ontologies. Specifically, the selected ontologies are the Human Phenotype Ontology (HPO), the Mammalian Phenotype Ontology (MP), the Human Disease Ontology (DOID) and the Orphanet and Rare Diseases Ontology(ORDO). The evaluation was run on an Ubuntu Laptop with an Intel Core i9-8950UK CPU @ 2.90GHz x 12 coupled with 25Gb RAM. POMap++ succeeded to complete tow tasks HP-MP and DOID-ORDO. POMap produced 1502 mappings in the HP-MP task associated with 214 unique mappings. Among the eight matching systems, POMap++ achieved the fifth highest F-measure with an F-Measure of 69.9%. In the DOID-ORDO task, POMap generated 2563 mappings with 174 unique ones. For this task, POMap++ obtained an F-Measure of 84.5% being the third best result for this track.

2.3 LargeBio

This tracks aims to find the alignment between three large ontologies: Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). Among six matching tasks between these three ontologies, POMap++ succeeded to perform the matching between FMA-NCI (small fragments) and FMA-SNOMED (small fragments) with an F-Measure respectively of 88.9% and 40.4%. For the other tasks of the large biomedical track, POMap++ exceeded the defined timeout.

3 Conclusion

The obtained results of POMap++ are promising especially for disease and phenotype as well as the anatomy track in which we ranked as the third top performing matching system. However, we did not opt to perform the local matching using structural-level features. Consequently, we are planning to add structural-level feature to the local matching process.

References

1. Bodenreider, O.: The unified medical language system (umls): integrating biomedical terminology. vol. 32, pp. D267–D270. Oxford University Press (2004)
2. Kohavi, R., John, G.H.: Wrappers for feature subset selection. vol. 97, pp. 273–324. Elsevier (1997)
3. Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F.: Pomap results for oaei 2017. In: 12th International Workshop on Ontology Matching collocated with the 16th International Semantic Web Conference (OM@ ISWC'17). pp. pp–1 (2017)
4. Lipscomb, C.E.: Medical subject headings (mesh). vol. 88, p. 265. Medical Library Association (2000)
5. Müllner, D.: Modern hierarchical, agglomerative clustering algorithms (2011)
6. Mungall, C.J., Torniai, C., Gkoutos, G.V., Lewis, S.E., Haendel, M.A.: Uberon, an integrative multi-species anatomy ontology. vol. 13, p. R5. BioMed Central (2012)
7. Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A., Chute, C.G., et al.: Bioportal: ontologies and integrated data resources at the click of a mouse. vol. 37, pp. W170–W173. Oxford University Press (2009)
8. Porter, M.F.: Snowball: A language for stemming algorithms (2001)
9. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Transactions on knowledge and data engineering 25(1), 158–176 (2013)
10. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics. pp. 133–138. Association for Computational Linguistics (1994)