

A Framework for Recommending Ontology Matching Systems based on Application Requirements

Diego Pessoa

Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Recife, Brazil
`derp@cin.ufpe.br`

Abstract. Ontology matching is the process of generating correspondences between terms of different ontologies. Today, several methods for ontology matching have been proposed, which makes difficult the choice of the most suitable to use in a particular setting. In this paper, we propose a novel ontology matching framework that uses automatic matchers recommendation to generate alignments. The differential of this work is the employment of application requirements as means of acquiring knowledge about a particular matching task.

Keywords: Ontology Matching, Ontology Matchers Recommendation, Knowledge Acquisition

1 Problem Statement

Ontology matching is the task of finding relationships between entities expressed in different ontologies [7]. It usually outputs alignments containing a set of correspondences between ontology terms, which are generated by using a single similarity measure or by combining different ones [6].

In the last years, many Ontology Matching Systems (Ontology Matchers) have been proposed, as stated in [15]. There is a yearly event, organized by the Ontology Alignment Evaluation Initiative (OAEI, e.g., [1]), in which matchers are tested under different test cases. The OAEI results have demonstrated that the evaluated matchers have achieved different performances depending on the matching task. For example, in 2016 edition, the matcher ALIN reached a high F-measure (0.74) in the Conference test case, but it was unable to provide any results in the Large-bio test case [1]. As there is limited knowledge about which factors may impact on the matchers' performance, it becomes challenging to a user the choice of the most suitable ones for a particular matching task. This fact increases the need for an automatic approach to select, combine and tune matchers.

This work introduces a framework for the automatic recommendation of ontology matchers regarding an application-specific matching task. Existing frameworks typically consider matchers' parameters or reference alignments as input. In this case, the user is engaged mostly only in the later validation of correspondences. A differential of the proposed framework is that it allows the user to define a set of application requirements, which are formalized as RDF resources. The framework aims not only to reduce the search space of a matching task (by producing ontology segments), but also to recommend the most suitable matchers for the reduced setting generated according to the requirements.

2 Relevancy

The integration of ontologies (i.e., establishing a unified view of ontologies from heterogeneous sources) has several applications (e.g. data integration, search, and analysis). Ontology matching is an essential step in this process, as sources usually employ different terms to describe the same real-world concept, even in the case of sources from the same domain.

Especially in large-scale matching tasks (i.e., when dealing with several ontologies that may contain a lot of elements), it is hard to acquire good quality alignments. It is because of the necessity of more computational effort (given the high quantity of items to compare), as also a greater number of user validations. As there are several ontology matchers available, the configuration of a matching task may be a complicated and time-consuming task to the user. In this aspect, there is a lack of approaches that could automatically provide the generation of alignments, by using a set of recommended matchers for a particular matching task.

The basic idea of proposed framework is to solve this issue by allowing users to define a set of application requirements, enabling both the reduction of the amount of compared terms and the delivery of a set of recommended matchers. Consequently, it also will make easier the configuration of a particular matching task, rather than the need of having knowledge on matchers' characteristics.

3 Related Work

There are a few works in the literature addressing the ontology matchers recommendation problem. The work in [13] has identified (by applying questionnaires with domain experts) a set of features related to matchers (regarding input, output, approach, usage, cost, and documentation). For each feature, the user can define weights which are used by a multi-criteria decision method called Analytic Hierarchy Process (AHP) that determines the suitable matchers. However, they consider only a fixed set of matchers, in such a way that it would be necessary to apply new questionnaires to identify the features for novel approaches. As ontology matchers are in constant evolution, this could be a useless effort. Also, users may not have knowledge about matchers peculiarities, which makes challenging to choose the relevant ones according to their interests.

The approaches [12, 14] consider textual and structural-based characteristics of input ontologies to recommend matchers before the task execution. However, it would lead to dismissing matchers that may provide better results in practice, since they do not consider any result of alignments. The work in [2] deals with this issue by considering previous results. But, as it would be unfeasible to run all matchers on every possible scenario, they are executed only over random ontology samples (called ontology segments). However, generating random samples may lead to uncertain evaluations, since every execution may present different results.

In [11], three recommendation strategies based on the use of ontology segments are proposed. The first generates segment pairs based on the exact matching with a set of concepts. The second considers a whole set of validated mapping suggestions and the third only segment pairs of these validated set. However, there is no assurance that good performance on parts of the ontologies may lead to the same result on the whole ontologies. Furthermore, several measures can be used to define the matcher performance, which can result in different recommendations depending on the chosen metric.

To the best of our knowledge, there are no other work that addresses ontology matchers recommendation by employing application requirements as means of acquiring knowledge about the priorities for a particular ontology matching task. The assumption of using requirements is allowing the user to define the relevant terms and the quality metrics to be considered in the matching. We intend to provide both a way to reduce the search space, through the generation of ontology segments related to terms that meet the requirements; and a form to evaluate the matchers more accurately, by considering the metrics that are more significant to the user.

4 Research Questions & Hypotheses

The following research questions (Q) and related hypothesis (H) investigate how the use of application requirements will reduce the search space of an ontology matching task and consequently improve the matchers' recommendation:

- **Q1:** Would the use of application requirements provide the generation of better ontology segments? Will these segments reduce the search scope of a matching task without loss of quality?
- **H1:** Employing application requirements will enable to generate better ontology segments, reducing more efficiently the search space of a matching task, compared to state of the art techniques.
- **Q2:** How can the use of application requirements improve ontology matchers' recommendations? Is it possible to formalize the application needs regarding an ontology matching task?
- **H2:** Application requirements will allow the users to specify which terms (data requirements) and metrics (quality requirements) should be considered in an ontology matching task. The generation of ontology segments based on the most relevant terms and the use of a set of preferred metrics

when evaluating matchers will provide better recommendations, compared to state of the art. RDF resources will be used to formalize the application requirements.

5 Proposed Approach

The proposed framework aims to support ontology matching users to perform matching tasks using a set of matchers suitable for a particular application. Figure 1 presents the respective framework components and workflow. We introduce a brief example to illustrate the definition of application requirements and detail the framework workflow in what follows.

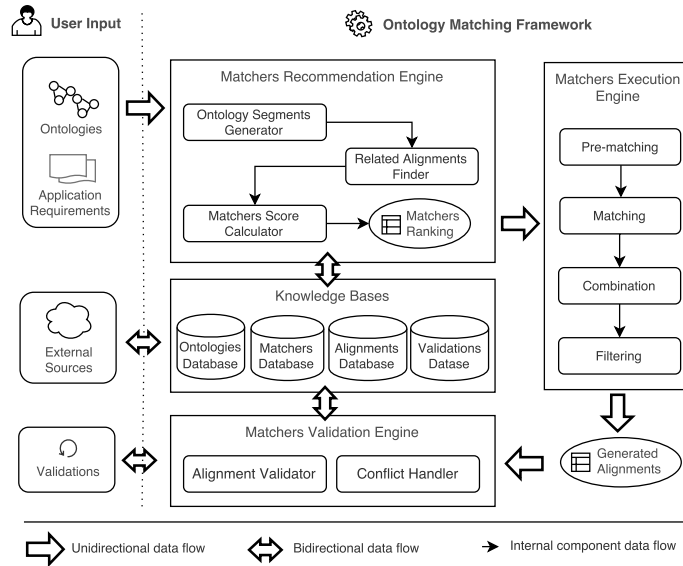


Fig. 1. Proposed Ontology Matching Framework.

To start an ontology matching task, the user should provide a pair of ontologies to match and a set of application requirements. These requirements are represented in the form of RDF statements, given the following two categories: i) Data Requirements and ii) Quality Requirements. The first are statements describing characteristics of the more relevant terms to be considered in the matching ontologies. These statements are used to generate the ontologies segments. The second ones, are statements that assign weights to quality metrics (e.g. precision, recall, execution time). These weights are used in the evaluation of alignments generated by matchers, resulting in recommendation scores.

To illustrate the definition of requirements, we introduce an ontology matching scenario. Suppose an application to integrate open data for understanding

motivations behind people’s migration from one country to another in the last years. Assuming that a large number of data sources may provide diverse data about countries and cities, this would be a typical case when application requirements can be used to specify the scope of a particular ontology matching setting. Table 1 shows the definition of two data requirements (DR1 and DR2), stating the preference for ontology classes that match with the terms *Weather* and *GDP*, in which the latter should have *GDP per capita* as a subclass. As Quality Requirement (QR), we illustrate the definition of the weights 0.7 and 0.3 for precision and execution time respectively, assuming that the application intends to reduce the number of generated correspondences and the execution time, given a large number of sources that may be considered.

Table 1. Examples of Application Requirements as RDF statements

Subject	Predicate	Object
#DR1	dataRequirement:hasMatchingClass	"weather"
#DR2	dataRequirement:hasMatchingClass	"GDP"
#DR2	dataRequirement:hasMatchingSubClass	"GDP per capita"
#QR1	qualityRequirement:hasPrecisionWeight	0.7
#QR2	qualityRequirement:hasExecutionTimeWeight	0.3

5.1 Framework Workflow

The framework workflow starts by receiving a set of ontologies and application requirements from the user. Then, the following steps are performed: i) ontology segment generation; ii) related alignments finding; iii) matchers score calculation, iv) matchers execution and iv) alignments validation. Furthermore, as a support for these steps, the framework also uses an already established knowledge base containing ontologies, matchers, alignments and validations acquired from reliable sources (e.g. OAEI).

Ontology Segment Generation. The first step is the generation of ontology segments following the data requirements provided, which will result in a reduced subset of ontologies containing only the most relevant terms for the user. The segment generation is made by traversing the ontologies structure searching for terms that meet the requirements. Ontology segments are automatically generated based on these terms and their correspondent elements (e.g. classes, subclasses, superclasses), depending on the data requirements.

Related Alignments Finding. The second step is seeking for related alignments (in the alignment database), i.e., the ones between ontologies that share similar characteristics with the generated segments. To define this similarity, we assign some values to ontologies (or segments) regarding the following matchers types: i) syntactic, ii) lexical; iii) structural and iv) instance-based.

Matcher Score Calculation. The third step is to calculate a score for the available matchers considering the results of alignments evaluations, following the weights for each metric defined in the quality requirements. The list of matchers ordered by score will compound the matchers ranking for the current setting.

Matchers Execution. Once the ranking of recommended matchers was established, the user can apply some criteria (e.g. minimum score threshold or maximum matchers quantity) to select matchers and then perform their execution, regarding the following steps: pre-matching, matching, combination, and filtering.

Alignment Validation. In the final step, the user can provide some feedback about alignments provided by the framework. For this, it is possible to walk through the alignments' correspondences and annotate them with positive or negative statements. This information is also stored in the Validations Database and may impact on subsequent interactions.

Knowledge Bases. The knowledge bases store information about ontologies, matchers, alignments and validations, serving as a baseline to the mentioned steps. Regardless of what is the current step in the workflow, the user can provide data to the knowledge base aiming to improve the obtained results. The Ontologies Database includes basic descriptions, such as URI and format (e.g. RDF, OWL). The Matchers Database stores some metadata about existing matchers, such as name, version, main features and service endpoint. In the Alignments Database, following as a standard the Alignment API [4], we store alignments and, if a gold-standard is available, it also stores a summary of metrics (e.g. precision, recall, f-measure) and matching information (e.g. correspondences found, expected and true positives) about the alignment generation. Finally, the Validation Database stores a set of statements containing negative or positive user annotations on correspondences.

6 Evaluation Plan

To evaluate the proposed framework, we perform experiments with real matchers on public datasets. The initial tests targeted the datasets provided by OAEI tracks (e.g. conference, anatomy). For this, we first have prepared the knowledge bases, by adding some metadata about reference ontologies, existing matchers (preferably OAEI participants) and alignments evaluations (when reference alignments are available). As further experiments, we also plan to test the framework in other domains, such as the integration of ontologies from open data repositories. Our hypothesis will be validated if the experiments demonstrate that the use of application requirements enables the reduction of a matching task and consequently the recommendation of the best matching systems.

7 Preliminary Results

To obtain some preliminary results, we have developed a prototype of the proposed framework. For initializing the Ontology Database, we have imported some ontologies from the OAEI Conference and Anatomy datasets (Cmt, Conference, ConfOf, Edas, Ekaw, Iasted, Sigkdd, Human, and Mouse). To fill the Matchers Database, we have considered the ones that usually are participants in OAEI campaigns and that have a publicly available source code. To standardize the access (input/output), we have implemented a wrapper for each matcher. In the initial experiment, we considered the matchers COMA [5], YAM [3], AML [9], LogMap [10] and FCAMap [8]. Figure 2 illustrates the comparison of obtained matching results considering the whole ontologies and the segments generated by the prototype. As result of this preliminary experiment, we observed that in the majority of cases, the prototype was able to improve quality metrics and in all the cases it was able to reduce the execution time.

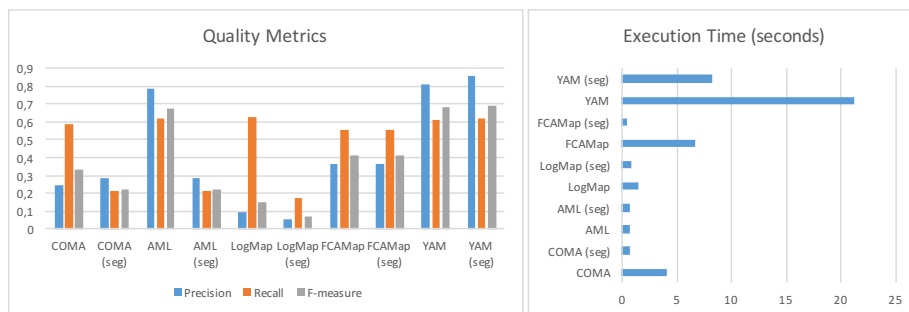


Fig. 2. Preliminary results.

8 Reflections

In this work, we present a framework for matchers recommendation based on application requirements. Even though the preliminary results indicate that the proposed approach is promising, we now are focused on performing further experiments to obtain more extensive results. We still have some work on design and implementation of the framework, but the main structure was implemented in the initial prototype, which will support the execution of new experiments.

We expect that the proposed framework can bring as main contributions: i) the ease of preparation of a matching task, by using requirements instead of matcher-specific parameters; and ii) the generation of alignments with better quality, by reducing the matching search space and by using the best recommended matchers to generate alignments. Another benefit of this framework would be the reduction of execution time, since the matching will not be performed on entire ontologies, but only on the more relevant segments to the user.

Acknowledgments. I am grateful to my advisor Dr. Ana Carolina Salgado and my co-advisor Dr. Bernadette Farias Lóscio for their support and the opportunity for the realization of this work.

References

1. Achichi, M., Cheatham, M., Dragisic, Z., Euzenat, J., Faria, D., Ferrara, A., Flouris, G., Fundulaki, I., Harrow, I., Ivanova, V., Jiménez-Ruiz, E., Kuss, E., Lambrix, P., Leopold, H., 0001, H.L., Meilicke, C., Montanelli, S., Pesquita, C., Saveta, T., Shvaiko, P., Splendiani, A., Stuckenschmidt, H., Todorov, K., dos Santos, C.T., Zamazal, O.: Results of the Ontology Alignment Evaluation Initiative 2016. OM@ISWC (2016)
2. Anam, S., Kim, Y.S., Kang, B.H., Liu, Q.: Adapting a knowledge-based schema matching system for ontology mapping. In: Proceedings of the Australasian Computer Science Week Multiconference. pp. 27:1–27:10. ACSW '16, ACM, New York, NY, USA (2016), <http://doi.acm.org/10.1145/2843043.2843048>
3. Bellahsene, Z., Ngo, D.H., Bellahsene, Z.: YAM++ : (not) Yet Another Matcher for Ontology Matching Task. *Bases de Données Avancées* p. 5 (2012)
4. David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C.: The Alignment API 4.0. *Semantic Web ()* 2(1), 3–10 (Jan 2011)
5. Do, H.H., Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches. *VLDB* pp. 610–621 (2002)
6. Elshwimy, F.A., Algergawy, A., Sarhan, A., Sallam, E.A.: Aggregation of similarity measures in schema matching based on generalized mean. 2014 IEEE 30th International Conference on Data Engineering Workshops (ICDEW) pp. 74–79 (2014)
7. Euzenat, J., Shvaiko, P.: *Ontology Matching*. Springer Publishing Company, Incorporated, 2nd edn. (2013)
8. Fan, L., Xiao, T.: An automatic method for ontology mapping. In: Apolloni, B., Howlett, R.J., Jain, L.C. (eds.) *KES (3)*. Lecture Notes in Computer Science, vol. 4694, pp. 661–669. Springer (2007)
9. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: *The AgreementMakerLight Ontology Matching System*. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
10. Jiménez-Ruiz, E., Grau, B.C.: LogMap: Logic-Based and Scalable Ontology Matching. In: *The Semantic Web – ISWC 2011*, pp. 273–288. Springer, Berlin, Heidelberg, Berlin, Heidelberg (Oct 2011)
11. Lambrix, P., Kaliyaperumal, R.: A Session-based Ontology Alignment Approach enabling User Involvement. *Semantic Web Journal* (2016)
12. Li, J., Tang, J., Li, Y., Luo, Q.: Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Trans. on Knowl. and Data Eng.* 21(8), 1218–1232 (Aug 2009), <http://dx.doi.org/10.1109/TKDE.2008.202>
13. Mochol, M., Jentzsch, A., Euzenat, J.: Applying an analytic method for matching approach selection. In: *OM'06: Proceedings of the 1st International Conference on Ontology Matching - Volume 225*. pp. 37–48. Free University of Berlin, CEUR-WS.org (Nov 2006)
14. Pirró, G., Talia, D.: UFOME: An ontology mapping system with strategy prediction capabilities. *Data & Knowledge Engineering* 69(5), 444–471 (May 2010)
15. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. *Knowledge and Data Engineering, IEEE Transactions on* 25(1), 158–176 (2013)