XMapGen and XMapSiG Results for OAEI 2013

Warith Eddine Djeddi and Mohamed Tarek Khadir

LabGED, Computer Science Department, University Badji Mokhtar, Annaba, Algeria
{djeddi, khadir}@labged.net

Abstract. The XMapGen and XMapSig systems are flexible and self-configuring matching tools using different strategies for combining multiple similarity measures into a single aggregated metric with the final aim of improving the ontology alignment quality of large scale ontologies. XMapGen and XMapSig are two variants of XMap++. The results obtained by the two ontology matching tools within the 9th edition of the Ontology Alignment Evaluation Initiative (OAEI 2013) campaign are therefore presented.

1 Presentation of the system

We present a fully automatic general purpose ontology alignment tools called XMapGen (eXtensible Mapping using Genetic) and XMapSig (eXtensible Mapping using Sigmoid), a new and lighter implementations of their ancestor XMap++ [1]. XMapGen and XMapSig include several matchers. These matchers calculate similarities between the terms from the different source ontologies. The matchers implement strategies based on linguistic matching, structure-based strategies and strategies that use auxiliary information in the thesaurus WordNet to enhance the alignment process. XMapGen uses Genetic Algorithm (GA) as a machine learning-based method to ascertain how to combine multiple similarity measures into a single aggregated metric with the final aim of improving the ontology alignment quality. XMapSig uses sigmoid function [4] for combining the corresponding weights for different semantic aspects, reflecting their different importance. This year, XMapGen and XMapSig participate in five tracks including Benchmark, Conference, Library, Anatomy and Large Biomedical Ontologies tracks.

1.1 State, purpose, general statement

XMapGen and XMapSig are a scalable ontology alignment tools capable of matching English language ontologies described in different OWL languages (i.e., OWL Lite, OWL DL, and OWL Full). The major principle of the matching strategy in XMapGen and XMapSig approaches is combining multiple similarity measures into a single similarity metric using weights determined by intelligent strategies in order to skip over the burden of manual selection. Despite the impressive strategy in adding GA, aligning medium-sized and large-scale ontologies is still very time consuming and computationally expensive. This inspires us to consider the use of a particular parallel matching on multiple cores or machines for dealing with the scalability issue on ontology matching.

1.2 Specific techniques used

In this section, the workflow of XMap++ and its main components is briefly described and shown in Fig.1. Both systems XMapGen and XMapSig calculate three different basic measures to create three similarity matrixes. String-based, semantic and structural methods are the three different categories of measuring similarities.

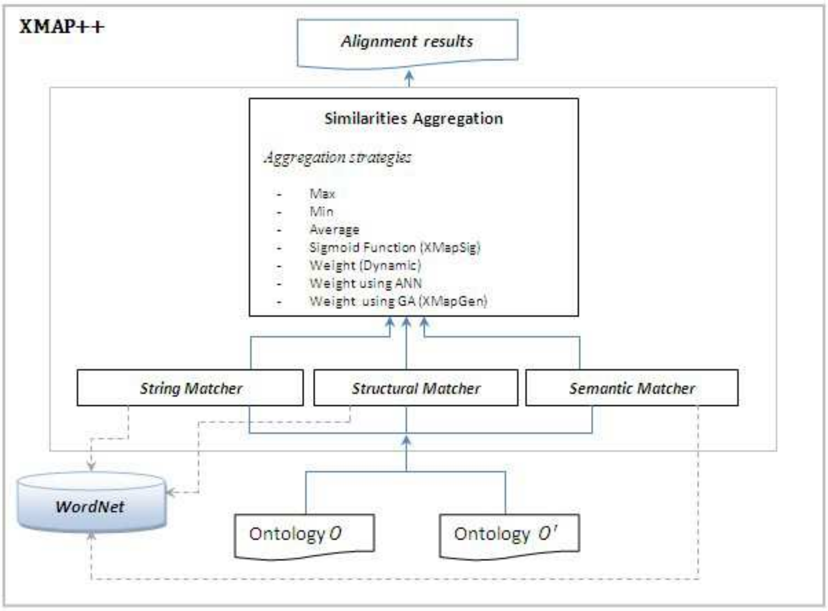

Fig. 1. Sketch of Architecture for XMAP++.

In XMap++ approach, a generic workflow for a given ontology matching scenario is as follows:

1. Matching inputs are two ontologies, source O and target $O^{'}$ parsed by an Ontology Parser component;
2. The ***String Matcher*** based on linguistic matching compares the textual descriptions of the concepts associated with the nodes (labels, names) of each ontology;
3. The ***Linguistic matcher*** jointly aims at identifying words in the input strings, relaying on WordNet [5] which provide additional information towards unveiling mappings in cases where features such as labels are missing or in cases where names are replaced by random strings. These matching techniques may provide incorrect match candidates, structural matching is used to correct such match candidates based on their structural context. In order to deal with lexical ambiguity, we introduce the notion of *scope* belonging to a concept which represents the context where it is placed. In our approach, the similarity between two entities of different ontologies is evaluated not only by investigating the semantics of the entities

names, but also taking into account the local context, through which the effective meaning is described. In particular, the neighborhood of a term (immediate parent and children in the *is-a* hierarchy). Increasing the radius means enlarging the scope (i.e. this area) and, consequently, the set of neighbour concepts that intervene in the description of the context. The value of linguistic methods is added to the linguistic matcher or the structure matcher in order to enhance the semantic ambiguity during the comparison process of entity names;

4. The ***structural matcher*** aligns nodes based on their adjacency relationships. The relationships (e.g., *subClassOf* and *is-a*) that are frequently used in the ontology serve, at one hand, as the foundation of the structural matching. On the other hand, the structural rules are used to extract the ontological context of each node, up to a certain depth (radius). This context includes some of its neighbours, where each of them is associated a weight representing the importance it has when evaluating the contextual node. The XMap++ algorithm values the semantic relation between two concepts while taking in consideration the types of cardinality constraints (e.g. *OWLAllValuesFrom*, *OWLSomeValuesFrom*, *OWLMinCardinality*, *OWL-Cardinality*, *OWLMaxCardinality*, *Same_as* or *Kind_of*) and values between their properties (e.g. *OWLMaxCardinality* $>=$*1*). Alignment suggestions are then determined by combining and filtering the results generated by one or more matchers;

5. The three matchers perform similarity computation in which each entity of the source ontology is compared with all the entities of the target ontology, thus producing three similarity matrices, which contain a value for each pair of entities. After that, an aggregation operator is used to combine multiple similarity matrices computed by different matchers to a single aggregated $n * m$ similarity matrix, where n is the number of element in the source ontology and m is the number of elements in the target ontology. We refer to [1] for more detail about the pruning and splitting techniques on data matrices for two couple of entities;

6. XMap++ uses three types of aggregation operator; these strategies are *aggregation*, *selection* and *combination*. The *aggregation* reduces the similarity cube to a matrix, by aggregating all matcher's results matrices into one. This *aggregation* is defined by five strategies: *Max*, *Min*, *Average*, *sigmoid function* and *Weighted*. The *Max* strategy is an optimistic one, selecting the highest similarity value calculated by any matcher. Contrary, the *Min* strategy selects the lowest value. *Average* evens out the matcher results, calculating the average. The *sigmoid* method combines multiple results using a sigmoid methods, which is essentially a smoothed threshold function [4]. In order to satisfy a different importance of matcher results, *Weighted* computes the weighted sum of the results, according to user defined weights or automatic defined weights using a dynamic strategy [3], using an Artificial Neural Network (ANN) (Djeddi and Khadir, 2013) or using Genetic Algorithm (GA);

7. Finally, these values are filtered using a selection according to a defined threshold and the desired cardinality. In our algorithm, we adopt the *1-1* cardinality to find the optimal solution in polynomial time.

1.3 Adaptations made for the evaluation

Several technical adaptations were required for integrating the system into the Seals platform, such as:

- Updating some libraries (e.g., Alignment API) or changing the way some parameters are communicated.
- To deal with large ontologies, XMapGen and XMapSig conducted specific experiments to see whether a matching system can exploit a multi-core architecture [6] to speed up the matching process. We adapted parallel matching to the use of threading to distribute the jobs of two matchers (Classes matcher and Properties matcher) on all available CPU cores on only one machine.
- There are two factors that directly impact to the systems' performance. The first ones relates to matching by machine learning model. The training data and selected similarity metrics as learning attributes are important. A simple solution for this issue is proposed by selecting the most appropriate similarity metrics and training data according to their correlation with expert's assessment. The second issue relates to the threshold used as a filter in the selection module. Different tests require different thresholds.
- In XMap++, the aim of the *Structural Matcher* is to correct such match candidates based on their structural context. The structural approach matches the nodes based on their adjacency relationships. XMapGen and XMapSig exploit only the superclass-subclass relationships (subsumption relationships) that are frequently used in ontologies when the total number of entities is bigger than 1500 entities in each ontology. *We restrict the contextual similarity computing; only the value of the semantic relation between two concepts without taking in consideration the types of cardinality constraints and values between their properties*, because if the ontologies became larger, the efficiency of the automatic alignment methods decreases considerably, in term of execution time, and memory size.

2 Results

Evaluation results of XMapGen and XMapSig in the OAEI 2013 campaign are here evaluated and discussed. We participated in five tracks: *Benchmarks*, *Conference*, *Library*, *Anatomy* and *Large Biomedical* Ontologies. Detailed results and descriptions about the used computation environments are provided on the OAEI 2013 result page.

2.1 Benchmark

In this track, there are multiple match tasks per sub-track where one source ontology is compared with a number of systematically modified target ontologies. According to Table 1, it is shown that approximately 4% is the percentage improvement of XMapSig versus XMapGen. The recall low values are explained by the fact that ontological entities with scrambled labels, lexical similarity becomes ineffective. For two algorithms, structural similarity stems from lexical similarity hence scrambling the labels makes the alignment more challenging. This trend of reduction in precision, recall and f-measure which can be observed throughout the test cases from 248 till 266.

Table 1. Results for Benchmark track.

System	biblioc		
	P	R	F
XMapSig	0.70	0.50	0.58
XMapGen	0.66	0.46	0.54

2.2 Anatomy

The Anatomy track consists of finding an alignment between the Adult Mouse Anatomy (2744 classes) and a part of the NCI Thesaurus (3304 classes) describing the human anatomy. XMapSig achieves a good F-Measure value of ≈75% in an acceptable amount of time (393 sec.) (see Table 2). In a separate configuration using genetic algorithm (XMapGen) we could increase the recall to ≈2% but the precision is decreased of ≈5%, due to running the structural matcher in a lightweight version (restriction of the contextual similarity). XMapGen needs around 403 minutes to compute the alignment. We plan to use bio medical lexical databases like Unified Medical Language System (UMLS) for improving the recall.

Table 2. Results for Anatomy track.

System	Precision	F-Measure	Recall	Time(s)
XMapSig	0.856	0.753	0.673	393
XMapGen	0.808	0.747	0.695	403

2.3 Conference

The Conference track uses a collection of 16 ontologies from the domain of academic conferences. Most ontologies were equipped with OWL DL axioms of various kinds; this opens a useful way to test our semantic matchers. The match quality was evaluated against an original (ra1) as well as entailed reference alignment (ra2). As the Table 3 shows, for both evaluations we achieved F-Measure values better than the Baseline1 results (57% for ra1 and 53% for ra2) when using XMapGen1_4. Also with XMapSig1_3 we achieved F-Measure values better than the Baseline1 results (58% for ra1 and 53% for ra2).

Table 3. Results for Conference track.

System	RA1 Reference			RA2 Reference		
	P	R	F	P	R	F
XMapSig1_3	0.72	0.48	0.58	0.68	0.44	0.53
XMapGen1_4	0.68	0.49	0.57	0.64	0.45	0.53

2.4 Library

The library track involves the matching of the STW thesaurus (6,575 classes) and the Soz thesaurus (8,376 classes). Both of these thesauri provide vocabulary for economic and social sciences. Table 4 summarizes the results obtained by XMapGen and XMapSig. The mapping quality achieved by XMapSig on the library track is not as positive as on the other tracks. XMapSig attains a precision of 0.79 and a recall of 0.31. Possible reasons may be the absence of domain and range definitions (in fact, of properties in general), as for anatomy, and the presence of multi-lingual labels. As XMapSig does not respect languages, this may lead to false positives. XMapSig requires $\approx$ 48 min and 34 sec. It is mainly due to the fact that our approach uses the notion of context with a value of radius not fixed as an input parameter at the starting of the matching task. So the algorithm looks at all the depth for the compared ontologies which involve the creation of a matrix with $M > 1.3$ billion pairs.

XMapGen could perform worse in terms of precision (0.031) and obtained higher for recall than XMapSig (0.37). The low of precision is due to a problem in the training of the genetic algorithm. We fixed this problem with an improved version delivered after deadline (precision and recall performance was different). In this paper we decided to present (see Table 4), only the results generated with the official version of our tool (before the deadline of the contest), and not the one generated with an improved version (fixing the training problem of GA) submitted after the deadline.

Table 4. Results for Library track.

System	Precision	Recall	F-Measure	Time(s)
XMapSig	0.799	0.318	0.455	2914
XMapGen	0.031	0.371	0.057	3008

2.5 Large biomedical ontologies

This data set consists of several large scale ontologies, containing up to tens of thousands of concepts. Our two systems were only capable to match the small task for FMA-NCI and FMA-SNOMED. The large ones are not finished in time due to the high computational complexity. We found the NCI thesaurus very time consuming for context based mapping as its concepts have many siblings. Among the varying evaluation methods, XMapGen and XMapSig produced fairly consistent alignments when matching the FMA and NCI ontologies, all resulting in f-measures of approximately 0.60 (See Table 5). However, the results of the completed tasks indicate that our system is already capable of producing alignments of high quality in this domain, thus improving its efficiency, for instance by applying the complete functionalities of XMap++ , should result in an overall satisfying performance during the next evaluation. As not expected from our two systems, they could perform the alignment in less than 3 hours 25 min of Small FMA-SNOMED fragments with high precision and low recall (See Table 6).

Table 5. Results for the Large BioMed track: FMA-NCI tasks

Task 1: Small FMA and NCI fragments					
System	Size	Precision	Recall	F-Measure	Time(s)
XMapSig	1564	0.864	0.461	0.602	1477
XMapGen	1687	0.833	0.479	0.608	1504

Table 6. Results for the Large BioMed track: FMA-SNOMED tasks

Task 3: Small FMA and SNOMED fragments					
System	Size	Precision	Recall	F-Measure	Time(s)
XMapSig	1581	0.760	0.134	0.228	11720
XMapGen	1827	0.694	0.142	0.236	12127

3 General comments

3.1 Comments on the results and future improvements

As previously stated, the aim of this development experience was not to deliver a tool to compete with others in terms of precision and recall. Instead, we aimed at the development of a new and stable version of XMap++ using new and state-of-the-art technologies and alignment methods. Additionally, to tackle the large ontology matching problem we improved the runtime of the algorithm using a divide-and-conquer approach that can partition the execution of the matchers into small threads was improved and joins their results after each similarity calculation. A direct comparison between the XMapGen and XMapSig shows that the addition of GA does not has a negative effect on the algorithm but, on the contrary, leads to slightly better results, especially in terms of recall. In most track, XMapSig supplies high precision than XMapGen. Whereas using Genetic Algorithm (XMapGen) performs quite high in terms of recall than using sigmoid function (XMapSig). The reason is behind the using of the sigmoid function and the weight for linguistic matcher. Therefore, for a high value of the linguistic weight, some important properties of classes founded by XMapSig may be omitted, as the weight of linguistic matcher is high and the algorithm focuses more on the linguistic level (names of classes) than the structural level (properties and their restrictions). This problem can be resolved by using a sigmoidal function, which increases proportionally the important similarity of the structural matcher, to be considered in the final calculation of two classes similarities. Finally we participated in the OAEI 2013 with two variants with the aim to analyze the strength and the weakness of each strategy (Sigmoid and Genetic) at the goal for combing them in one ontology alignment task.

3.2 Discussions on the way to improve the proposed system

Some probable approaches to improving our tools are listed as follows:

1. Adopt more flexible strategies in defining the way for automatic threshold rather than manually tuning. Developing dynamic strategies for setting the correct threshold value for each compared ontologies and not one for all;

2. Take comments and Instance information of ontology into account, especially when the name of concept is meaningless;
3. Matching larger ontologies still takes significantly longer time when parsing ontologies with Alignment API. We plan to solve this problem using an ontology parser which permits to load multiple ontologies in parallel via threading;
4. Usage of background knowledge based on the UMLS Meta-thesaurus to have high recall when aligning ontologies from the biomedical science domain.

3.3 Comments on the OAEI 2013 procedure

As a first participation, we found the OAEI procedure very convenient and the organizers very supportive. The use of Seals allows objective assessments. The OAEI test cases are various and this leads to comparison on different levels of difficulty, which is very interesting. We found that SEALS platform is a very valuable tool to compare the performance of our system with the others.

4 Conclusion

Our system participated to the campaign with two versions (XMapGen and XMapSig) of our approach, corresponding to different strategies of weights aggregation. Generally, according to our results in OAEI 2013, our two systems delivered fair results comparatively to other participants. The preliminary results were quite good to encourage us to continue seeking better solutions. It seems that both systems XMapGen and XMapSig can efficiently match semantically rich ontologies containing tens (and even hundreds) of thousands of classes. We confirm that the addition of Genetic Algorithm (GA) keeps the performance and, furthermore, eliminates the necessity of tuning the weights manually. Moreover, the learning framework is very flexible: many combinations of matchers and parameters may be used in the future, various types of training models (Resilient propagation, Levenberg marquardt, Backpropagation, Anneal, Radial or Manhatan method, etc.) and new metrics.

References

1. Djeddi, W., Khadir, M.T.: Ontology alignment using artificial neural network for large-scale ontologies. In the International Journal of Metadata, Semantics and Ontologies (IJMSO), Vol.8, No.1, pp.75-92 (2013)
2. Djeddi, W., Khadir, M.T.: Introducing artificial neural network in ontologies alignment process. In the Journal Control and Cybernetics, Vol. 41, No. 4, pp.743-759 (2012)
3. Djeddi, W., Khadir, M.T. : A dynamic multistrategy ontology alignment framework based on semantic relationships using WordNet. In Proc of the 3rd International Conference on Computer Science and its Applications (CIIA11), 1315 December, Saida, Algeria, pp.149154 (2012)
4. Djeddi, W., Khadir, M.T.: XMAP: a novel structural approach for alignment of OWL-full ontologies. In Proc. of the International Conference on Machine and Web Intelligence (ICMWI), pp.347-352 (2010)
5. Fellbaum, C. : WordNet: An Electronic Lexical Database, MIT Press, Cambridge, MA (1998)
6. Gross, A., Hartung, M., Kirsten, T. and Rahm, E. : On matching large life science ontologies in parallel. In , in Lambrix, P. and Kemp, G.J.L. (Eds), DILS, Springer, pp.35-49 (2010)