ServOMap Results for OAEI 2013

Amal Kammoun[1] and Gayo Diallo[1]

ERIAS, INSERM U897, University of Bordeaux, France
first.last@isped.u-bordeaux2.fr

Abstract. We briefly present in this paper ServOMap, a large scale ontology matching system, and the performance it achieved during the OAEI 2013 campaign. This is the second participation in the OAEI campaign.

1 Presentation of the system

ServOMap [1] is a large scale ontology matching system designed on top of the ServO Ontology Server system [2], an idea originally developed in [3]. It is able to handle ontologies which contain several hundred of thousands entities. To deal with large ontologies, ServOMap relies on an indexing strategy for reducing the search space and computes an initial set of candidates based on the terminological description of entities of the input ontologies.

New components have been introduced since the 2012 version of the system. Among them:

- The use of a set of string distance metrics to complement the vectorial based similarity of the IR library we use[1],
- An improved contextual similarity computation thanks to the introduction of a Machine Learning strategy,
- The introduction of a general purpose background knowledge, WordNet [4], to deal with synonymy issues within entities' annotation,
- The use of a logical consistency check component.

In 2013, ServOMap participated in the entities matching track and does not implemented a specific adaptation for the **Interactive Matching** and **Multifarm** tracks.

1.1 State, purpose, general statement

ServOMap is designed with the purpose of facilitating interoperability between different applications which are based on heterogeneous knowledge organization systems (KOS). The heterogeneity of these KOS may have several causes including their language format and their level of formalism. Our system relies on Information Retrieval (IR) techniques and a dynamic description of entities of different KOS for computing the similarity between them. It is mainly designed for meeting the need of matching large scale ontologies. It has proven to be efficient for tackling such an issue during the 2012 OAEI campaign.

[1] http://lucene.apache.org/

1.2 Specific techniques used

ServOMap has a set of components highly configurable. The overall workflow is depicted on figure 1. It includes three steps briefly described in the following. Typically, the input of the process is two ontologies which can be described in OWL, RDF(S), SKOS or OBO. ServOMap provides a set of weighted correspondences [5] between the entities of these input ontologies.

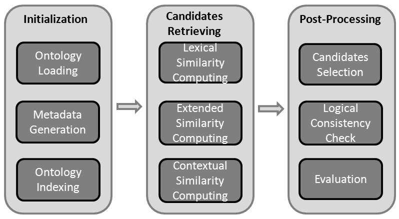

Fig. 1. ServOMap matching process.

Initialization Step. During the initialization step, the **Ontology Loading** component has in charge of processing the input ontologies. For each entity (concept, property, individual), a virtual document from the set of annotations is generated for indexing purpose. These annotations include the ID, labels, comments and, if the entity is a concept, information about it properties. For an individual, the values of domain and range are considered as well.

Metadata Generation. A set of metrics are computed. They include the size of input ontologies in term of concepts, properties and individuals, the list of languages denoting the annotations of entities (labels, comments), etc. Determining the size helps adapting latter the matching strategy. Indeed, besides detecting an instances matching case, we distinguish this year small (less than 500 concepts) from large ontologies. Detecting the set of languages allows using latter the appropriate list of stopwords.

Ontology Indexing. With ServOMap we consider an ontology as a corpus of semantic document to process. Therefore, the purpose of the indexing module is to build an inverted index for each input ontology from the virtual documents generated previously. The content of each virtual document is passed through a set of filters: stopwords removal, non alphanumeric characters removal, lowercasing and stemming labels, converting numbers to characters. In addition, labels denoting concepts are enriched by their permutation. This operation is applied to the first 4 words of each label. For instance, after enriching the term '*Bone Marrow Donation*' we obtain the set {*Bone Marrow Donation, Marrow Bone Donation, Marrow Donation Bone, Donation Marrow Bone, Donation Bone Marrow*}.

Further, two strategies are used for indexing, *exact* and *relaxed* indexing. Exact indexing allows high precise retrieving. In this case, before the indexing process, all words for each label are concatained by removing spaces between them. In addition, for optimization purpose, the possibility is offered to index each entity with information about its siblings, descendants and ancestors.

Candidates Retrieving. The objective is to compute a set of candidates mappings M $= \bigcup(M_{exact}$, $M_{relaxed}$, $M_{context}$, $M_{prop})$.

Lexical Similarity Computing. Let's assume that after the initializing step we have two indexes I_1 and I_2 corresponding respectively to the input ontologies O_1 and O_2. The first step for candidates retrieving is to compute the initial set of candidates mappings constituted by only couple of concepts and denoted by M_{exact}. This set is obtained by performing an exact search, respectively over I_1 using O_2 as search component and over I_2 using O_1. To do so, a query which takes the form of a virtual document is generated for each concept and sent to the target index. The search is performed through the IR library which use the usual *tf.idf* score. We select the best K results having a score greater than a given threshold θ. The obtained couples are filtered out in order to keep only those satisfying *the lexical similarity condition*. This condition is checked as follows.

For each filtered couple (c_1, c_2), two lexical descriptions are generated. They are constituted respectively by ID and labels of c_1 and its direct ancestors (Γ_1), ID and labels of c_2 and its direct ancestors (Γ_2).

We compute a similarity Sim_{lex}=$f(\alpha \times ISub(\Gamma_1, \Gamma_2), \beta \times QGram(\Gamma_1, \Gamma_2), \gamma \times Lev(\Gamma_1, \Gamma_2))$, where I-Sub, QGram and Lev denote respectively the ISUB similarity measure [6], the QGram and Levenshtein distance. Coefficients α, β and γ are chosen empirically for OAEI 2013. All couples with Sim_{lex} greater than a threshold are selected. Finally, M_{exact} is the intersection of the two set of selected couples obtained after the search performed on the two indexes.

The same process is repeated in order to compute the set $M_{relaxed}$ from the concepts not yet selected with the exact search. A similar strategy for computing M_{exact} is used for computing the similarity between the properties of the input ontologies. This generates the M_{prop} set. Here, the description of a property includes its domain and range.

Extended Similarity Computing. In order to deal with synonym issue, from the set of concepts not selected after the previous phase, we use the WordNet dictionary for retrieving alternative labels for concepts to be mapped. The idea is to check whether a concept in the first ontology is denoted by synonym terms in the second one. All couples in this case are retrieved as possible candidates.

Contextual Similarity Computing. The idea is to acquire new candidates mappings, $M_{context}$, among those couples which have not been selected in the previous steps. To do so, we rely on the structure of the ontology by considering that the similarity of two entities depends on the similarity of the entities that surround them. In 2013, we have introduced a Machine Learning strategy which uses M_{exact} as basis for training set using the WEKA tool [7]. Indeed, according to our tests, candidates mappings from M_{exact} use to be highly accurate. Therefore, retrieving candidates using contextual similarity is transformed as a classification problem. Each new couple is to be classified as *correct* or *incorrect* according to candidates already in M_{exact}.

We use 5 similarity measures (Levenshtein, Monge-Elkan, QGram, Jackard and BlockDistance) to compute the features of the training set. For each couple (c_1, c_2) $\in M_{exact}$, we compute the 5 scores using the ID and labels associated to c_1 and c_2 and denote this entry as *correct*. We complete M_{exact} by randomly generating new couples assumed to be incorrect. To do so, for each couple (c_1, c_2) in M_{exact}, we compute the 5 scores for (c_1, *ancestor*(c_2)), (*ancestor*(c_1), c_2), (*descendant*(c_1), c_2) and (c_1, *descendant*(c_2)) and denote them as *incorrect*. The *ancestor* and *descendant* functions retrieve the super-concepts and sub-concepts of a given concept. We use the J48 decision tree algorithm of Weka for generating the classifier.

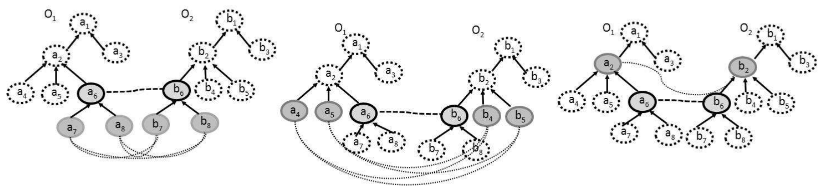

Fig. 2. Strategy for contextual based candidates generation. For each couple of M_{exact}, the similarity of the surrounding concepts are looked up.

We build the dataset to classify as follows. The exact set is used to learn new candidates couples according to the strategy depicted on figure 2 by assuming here for instance that (a_6, b_6) $\in M_{exact}$. For each couple of M_{exact}, the idea is to retrieve possible couples not already in M_{exact} among the sub-concepts ((a_7, b_7), (a_7, b_8), (a_8, b_8), (a_8, b_7) in figure 2), the super-concepts and the siblings. For each candidate couple (c_1, c_2), if the score

$$s = f(getScoreDesc(), getScoreAsc(), getScoreSib())$$

is greater than a fixed threshold, then we compute the 5 similarity scores for (c_1, c_2). The functions getScoreDesc(), getScoreAsc(), getScoreSib() compute respectively a score for (c_1, c_2) from its descendants, ancestors and siblings concepts. The obtained dataset is classified using the previously built classifier.

Post-Processing Step . This step involves enriching the set of candidates mapping (mainly incorporating those couples having all their sub-concepts mapped), the selection of the final candidates from the set M and performing inconsistency check. We have implemented a new filtering algorithm for selecting the best candidates based on their scores and we perform consistency check as already implemented in the 2012 version (disjoints concepts, criss-cross). Further, we use the repair facility of the LogMap system [8] to perform logical inconsistency check. Finally, we have implemented an evaluator for computing the usual Precision/Recall/F-measure for the generated final mappings if a reference alignment is provided.

1.3 Adaptations made for the evaluation

ServOMap is configured to adapt its strategy to the size of the input ontologies. Therefore, as mentioned earlier, two categories are considered: input ontology with size less than 500 concepts and ontology with size greater than 500 concepts. For large ontologies, our tests showed that exact search is sufficient for generating concepts mappings of OAEI test cases, while for small one relaxed and extended search is needed.

Further, according to the performance achieved by our system in OAEI 2012 [9], the focus of this year was more to improve the recall than optimizing the computation time. From technical point of view, the previous version of ServOMap was based on the following third party components: the JENA framework for processing ontologies and the Apache Luncene API as IR library. We have moved from JENA framework to the OWLAPI library for ontology processing, in particular for handling in an efficient manner complex domain and range axioms and taking into account wider formats of input ontologies. In addition, a more recent version of the IR library is used for the actual version. However, in order to have a compatible SEALS client, we have downgraded the version of the Apache Lucene API used for the evaluation. This leaded to a less robust system for the 2013 campaign as some components have not been fully adapted.

1.4 Link to the system and parameters file

The wrapped SEALS client for ServOMap version used for the OAEI 2013 edition is available at http://lesim.isped.u-bordeaux2.fr/ServOMap. The instructions for testing the tool is described in the tutorial dedicated to the SEALS client[2].

1.5 Link to the set of provided alignments

The results obtained by ServOMap during OAEI 2013 are available at http://lesim.isped.u-bordeaux2.fr/ServOMap/oaei2013.zip/.

2 Results

We present in this section the results obtained by running the ServOMap system with the SEALS client. As the uploaded version does not implement multilingual and interactive matching features, the results of the corresponding tracks are not described here.

2.1 Benchmark

In the OAEI 2013 campaign, the Benchmark track includes only the bibliography test case in a blind mode. The experiments are performed on a Debian Linux virtual machine configured with four processors and 8GB of RAM. ServOMap finished the task in about 7mn. Because of some issues in processing tests set from #261-4 to #266, the results of ServOMap has been affected and decreased compared to 2012.

[2] http://oaei.ontologymatching.org/2013/seals-eval.html

Test set	H-Precision	H-Recall	H-F-score
biblioc	0.63	0.22	0.33

Table 1. ServOMap results on the Benchmark track

2.2 Anatomy

The Anatomy track consists of finding an alignment between the Adult Mouse Anatomy (2,744 classes) and a part of the NCI Thesaurus (3,304 classes). The evaluation is performed on a server with 3.46 GHz (6 cores) and 8GB RAM. Table 2 shows the results and runtime of ServOMap.

Test set	Precision	Recall	F-score	Runtime (s)
Anatomy	0.961	0.618	0.752	43

Table 2. ServoMap results on the Anatomy track

2.3 Conference

The conference track contains 16 ontologies from the same domain (conference organization). These ontologies are in English and each ontology must be matched against each other. The match quality was evaluated against an original (ra1) as well as entailed reference alignment (ra2). ServoMap increased its performance in term of F-measure by 0.07. The table 3 shows the results obtained on this track.

Test set	Precision	Recall	F-score
Conference (ra1)	0.73	0.55	0.63
Conference (ra2)	0.69	0.5	0.58

Table 3. ServOMap results on the Conference track

2.4 Library

The library track is about matching two thesauri, the STW and the TheSoz thesaurus. They provide a vocabulary for economic respectively social science subjects and are used by libraries for indexation and retrieval. Thanks to the use of a new API for processing ontologies, ServOMap was able to handle directly the two thesauri of the library track without any adaptation. ServOMap performed the task in a longer time (4 compared to 2012 edition of OAEI, however by increasing the F-measure.

Test set	Precision	Recall	F-score	Runtime (s)
Library	0.699	0.783	0.739	648

Table 4. ServoMap results on the Library track

2.5 Large biomedical ontologies

The Large BioMed track consists of finding alignments between the Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). There are 6 sub tasks corresponding to different sizes of input ontologies (small fragment and whole ontology for FMA and NCI and small and large fragments for SNOMED CT). The results obtained by ServOMap are depicted on Table 5.

Test set	Precision	Recall	F-score	Runtime (s)
Small FMA-NCI	0.951	0.815	0.877	141
Whole FMA-NCI	0.727	0.803	0.763	2,690
Small FMA-SNOMED	0.955	0.622	0.753	391
Whole FMA- Large SNOMED	0.861	0.620	0.721	4,059
Small SNOMED-NCI	0.933	0.642	0.761	1,699
Whole NCI- Large SNOMED	0.822	0.637	0.718	6,320

Table 5. ServOMap results on Large BioMed Track

3 General comments

This is the second time that we participate in the OAEI campaign. While we participated with two configurations of our system to the 2012 edition of the campaign, respectively with ServOMap-lt and ServOMap, this year a unique version has been submitted. Several changes have been introduced. We moved from JENA to OWLAPI for processing ontologies and a more recent version of the Apache Lucene API that is used as IR tool. This last change introduced some issues on having a wrapped tool compatible with the Seals client. Therefore, the uploaded version of ServOMap uses a downgraded version of Lucene to be able to run correctly with the client. This resulted of a degraded performance and less robust system compared to that obtained with the actual version of our tool. Further, the uploaded version has not been optimized in term of computation time. This affected particularly the runtime for the Large BioMed Track.

3.1 Comments on the results

The evaluated ServOMap version for OAEI 2013 shows a significant improvement for the conference and library track. We have increased our recall in several tasks without loosing enough in term of precision. Overall, We notice that, the introduction of string similarity measures and inconsistency repair facility affected the computation time. However, ServOMap confirmed its ability to cope with very large dataset but also shows that it relies heavily on the terminological richness of the input ontologies.

4 Conclusion

We have briefly described the ServOMap ontology matching system and presented the results achieved during the 2013 edition of the OAEI campaign. Several components, including Machine Learning based contextual similarity computing, have been added to the previous version. In the vein of the last year participation, the performance achieved by ServOMap are still very interesting and places it among the best system for large scale Ontology matching. Future work will include improving the strategy of contextual similarity computing and focusing on a more efficient semantic filtering component of candidate mappings. Further, we will investigate interactive and multilingual matching issues.

5 Acknowledgments

This work has been partly supported by the EU FP7 ARITMO project. We also thank the organizers of OAEI with providing test dataset and the evaluation infrastructure.

References

1. M. Ba and G. Diallo. Large-scale biomedical ontology matching with servomap. *IRBM*, 34(1):56 – 59, 2013. Digital Technologies for Healthcare.
2. Gayo Diallo. Efficient building of local repository of distributed ontologies. In *IEEE Proceedings of the SITIS'2011 International Conference.*, pages 159–166, November 2011.
3. Gayo Diallo. *An ontology-based architecture for structured and unstructured data management*. PhD thesis, Université Joseph Fourier - Grenoble 1, December 2006. Original Title: Une Architecture á base d'Ontologies pour la Gestion Unifiées des Données Structurées et non Structurées.
4. George A. Miller. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 38:39–41, 1995.
5. Pavel Shvaiko and Jérôme Euzenat. Ontology matching: State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1):158–176, 2013.
6. Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias. A string metric for ontology alignment. In Y. Gil, editor, *Proceedings of the International Semantic Web Conference (ISWC 05)*, volume 3729 of *LNCS*, pages 624–637. Springer-Verlag, 2005.
7. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.
8. Ernesto Jimenez Ruiz, Bernardo Cuenca Grau, Yujiao Zhou, and Ian Horrocks. Large-scale interactive ontology matching: Algorithms and implementation. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, pages 444–449. IOS Press, 2012.
9. José-Luis Aguirre, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Willem Robert van Hage, Laura Hollink, Christian Meilicke, Andriy Nikolov, Dominique Ritze, and François Scharffe et al. Results of the ontology alignment evaluation initiative 2012. In Pavel Shvaiko, Jérôme Euzenat, Anastasios Kementsietsidis, Ming Mao, Natasha Fridman Noy, and Heiner Stuckenschmidt, editors, *OM*, volume 946 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.