# OAEI 2017 results of KEPLER

Marouen KACHROUDI[1], Gayo DIALLO[2], and Sadok BEN YAHIA[1]

[1] Université de Tunis El Manar, Faculté des Sciences de Tunis
Informatique Programmation Algorithmique et Heuristique
LIPAH-LR 1 ES14, 2092, Tunis, Tunisie
{marouen.kachroudi,sadok.benyahia}@fst.rnu.tn
[2] BPH Center - INSERM U1219, Team ERIAS & LaBRI UMR5800,
Univ. Bordeaux
gayo.diallo@u-bordeaux.fr

**Abstract.** This paper presents and discusses the results produced by the KEPLER system for the 2017 Ontology Alignment Evaluation Initiative (OAEI 2017). This method is based on the exploitation of three different strategy levels. The proposed alignment method KEPLER is enhanced by the integration of powerful treatments inherited from other related domains, such as Information Retrieval (IR) [1]. For scaling, the method is equipped with a partitioning module. For the management of multilingualism, the KEPLER method develops a well-defined strategy based on the use of a translator, and this provides very encouraging results.

## 1 Presentation of the system

Given the substantial growth of the Semantic Web users that creates and updates knowledge all over the world in a multitude of conceptualizations. This process has been accelerated due to a few initiatives which encourage all the active participants to make their data available to the public. These actors often publish their data sources in their own respective languages, in order to make this information interoperable and accessible to members of other linguistic communities [2]. As a solution, the ontology alignment process aims to provide semantic interoperable bridges between heterogeneous and distributed information systems. Indeed, the informative volume reachable via the Semantic Web stresses needs of techniques guaranteeing the share, reuse and interaction of all resources [3]. The explicitation of the associated concepts related to a particular domain of interest resorts to ontologies, considered as the kernel of the Semantic Web. In this register, KEPLER is an ontology alignment system dealing with the key challenges related to heterogeneous ontologies on the semantic Web, and it uses several hybrid alignment strategies. KEPLER is designed to discover alignments for both normal size and large scale ontologies. In addition, the proposed alignment approach has the ability to treat multilingual ontologies as well as monolingual ones.

### 1.1 State, purpose, general statement

The proposed method, KEPLER, exploits besides the classic techniques, an external resource, *i.e.*, a translator to deal with multilingualism. The method KEPLER implements an alignment strategy which aims at exploiting all the wealth of the used ontologies.

### 1.2 Specific techniques used

The main idea of the KEPLER method is to exploit the expressiveness of the OWL language to infer the similarity between entities of two given ontologies. Entities are described using OWL primitives with their semantics. We can then consider ontology as a semantic graph where entities are nodes connected by links which are OWL primitives. These links have specified semantic primitives. Indeed, if two ontologies in the same domain are similar, their semantic graphs are also the same.

**Parsing and pretreatment** This module allows to extract the ontological entities initially represented by a primitive form of lists. In other words, at the parsing stage, we seek primarily to transform an OWL ontology in a well defined structure that preserves and highlight all the information contained in this ontology. Furthermore, in the resulting informative format, has a considerable impact on the results of the similarity computation thereafter. Thus, we get couples formed by the name of the entity and its associated label. In the next step we add an element to such couples to process these entities regardless of their native language.

**Partitioning** This module aims at splitting ontologies into smaller parts to support the alignment task [4]. Consequently, partitioning a set $\mathcal{B}(\mathcal{C})$ is to find subsets $\mathcal{B}_1$, $\mathcal{B}_2$,..., $\mathcal{B}_n$, encompassing semantically close elements bound by a relevant set of relationships, *i.e.*, $\mathcal{O} = \bigcup\{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_n\}$, where $\mathcal{B}_i$ is an ontological block, and $n$ is the resulting number of extracted blocks. Hence, we can define an ontological portion as a reduced ontology that could be extracted from another larger one by splitting up the latter according to its both constituants : structures and semantics. One way to obtain such a partitioning, can be to maximize the relationships inside a block and minimize the relationship between the blocks themselves. The partitioning quality result can be evaluated using different criteria:

- *The size of the generated blocks*: that must have a reasonable size, *i.e.*, a number of elements that can be handled by an alignment tool;
- *The number of the generated blocks*: this number should be as small as possible to limit the number of blocks pairs to be aligned;
- *The compactness degree of a block*: a block is said to be substantially compact if relations (lexical and structural ones) are stronger inside the block and low outside.

**Translation** : An originality of our system, is to solve the heterogeneity problem mainly due to multilingualism, given the importance of this area of research [5, 6]. This challenge brings us to choose between two alternatives, either we consider the translation path to one of the languages according to the two input ontologies, or we consider the translation path to a chosen pivot language. At this stage, we must have a foreseeable vision for the rest of our approach. Specifically, at the semantic alignment stage we use an external resource such as WordNet[3]. The latter is a lexical database for the English language. Therefore, the choice is governed by the use of WordNet, and we will prepare a translation of the two ontologies to the pivot language, which is English. To perform the translation phase we chose Bing Microsoft[4] tool.

[3] https://wordnet.princeton.edu/

[4] https://www.bing.com/translator

**Indexation** : Whether on the Internet, with many search engine or local access, we need to find documents or simply sites. Such research is valuable to browse each file and the analysis thereafter. However, the full itinerary of all documents with the terms of a given query is expensive since there are too many documents and prohibitive response times. To enable faster searching, the idea that was previously used in some works [1] is to execute the analysis in advance and store it in an optimized format for the search. Indexing is one of the novelties of our approach. It consists in reducing the search space through the use of effective search strategy on the built indexes. In fact, we no longer need the sequential scan because with the index structure, we can directly know what document contains a particular word. To ensure this indexing phase we use the Lucene[5] tool. Lucene is a Java API that allows developers to customize and deploy their own indexing and search engine. Lucene uses a suitable technology for all applications that require text search. Indeed, at the end of the indexing process, we get four different indexes to everyone of the two input ontologies depending on the type of the detected entities (*i.e.*, concepts, data types, relationships, and instances). More precisely, we get a first index file $\mathcal{I}_1$ that corresponds to concepts index, a second one $\mathcal{I}_2$ dedicated to relationships, a third one $\mathcal{I}_3$ where we find a *datatypes* file index, and the last is $\mathcal{I}_4$ a file indexing instances. The documents at the indexes represent semantic information about any ontological entity. These semantic information is enriched by means of the external resource (*i.e.*, WordNet). Indeed, for each entity, the method keeps the entity name, the label, the translated label in English and its synonyms in English. So with Lucene, we created a set of indexes for the two ontologies, a search query is set up to return all the mapping candidates.

**Candidate Mappings Identification** : With Lucene, `TermQuery` is the most basic query type to search through an index. It can be built using one term. In our case, `TermQuery`'s role is to find the entities in common between the indexes. Indeed, once the indexes are set up, the querying step of the latter is activated. Thus, the query implementation satisfies the terminology search and semantic aspects at once as we are querying documents that contain a given ontological entity and its synonyms obtained via WordNet. It is worthy to mention that indexes querying is done in both senses. In other words, if we have two indexes $\mathcal{I}_1$ and $\mathcal{I'}_2$ respectively belonging to $\mathcal{O}_1$ and $\mathcal{O}_2$, querying is outlined as follows : $(\mathcal{I}_1 \rightarrow Query \rightarrow \mathcal{I'}_2)$ and $(\mathcal{I'}_2 \rightarrow Query \rightarrow \mathcal{I}_1)$. Indeed, this junction greatly increases the alignment method performance. The result of this process is a set of documents sorted by relevance according to the Lucene score assigned to each returned document. Thus, for each query, our system keep the first five documents returned and considers them as candidate mappings for the next phase.

**Filtering and Recovery** : The filtering module consists of two complementary submodules, each one is responsible of a specific task in order to refine the set of primarily aligned candidates. Indeed, once the list of candidates is ready, the alignment method uses the first filter. Indeed, we should note that indexes querying may includes a set of redundant mappings. Doing so, this filter eliminates the redundancy. Indeed, it goes through the list of candidates and for each candidate, it checks if there are duplicates. If this is the case, it removes the redundant element(s). At the end of filtering phase, we have a candidates list without redundancy, however, there is always the concern of *false*

[5] https://lucene.apache.org/

*positives*, indeed, there was the need to establish a second filter. Once the redundant candidates are deleted, the system uses the second filter that eliminates *false positives*. This filter is applied to what we call *partially* redundant entities. An entity is considered as *partially* redundant if it belongs to two different mappings (*i.e.*, being given three ontological entities $e_1$, $e_2$ and $e_3$. If on the one hand, $e_1$ is aligned to $e_2$, and secondly, $e_1$ is aligned to $e_3$, this last alignment is qualified as doubtful. We note that our method generates $(1:1)$ alignments. To overcome this challenge, the alignment method compares the topology of the two suspicious entities ($e_3$ neighbors with $e_1$ neighbors, $e_2$ neighbors with $e_1$ neighbors ) with respect to the redundant entity $e_1$, and retains the couple having the highest topological proximity value. All candidates are subject of this filter, and as output we have the final alignment file.

**Alignment Generation** : The result of the alignment process provides a set of mappings, which are serialized in the RDF format.

## 2 Results

In this section, we present the results obtained by KEPLER in the OAEI $2017$.

### 2.1 Anatomy

This track consists of two real world ontologies to be matched, the source ontology describing the Adult Mouse Anatomy (with $2744$ classes) and the target ontology is the NCI Thesaurus describing the Human Anatomy (with $3304$ classes). For this track, KEPLER succeeded to extract $74\%$ of correct mappings with a precision about $95\%$.

### 2.2 Conference

The conference track consists of $15$ ontologies from the conference organization domain and each ontology must be matched against every other ontology. The dataset describes the domain of organizing conferences from different perspectives. Precision values varies between $76\%$ and $58\%$. Recall values varies between $48\%$ and $68\%$. The metrics are obtained according to several scenarios of evaluation.

### 2.3 Multifarm

This dataset is composed of a subset of the Conference track, translated in nine different languages (*i.e.*, Chinese, Czech, Dutch, French, German, Portuguese, Russian, Spanish and Arabic). With a special focus on multilingualism, it is possible to evaluate and compare the performance of alignment approaches through these test cases. The main goal of the MultiFarm track is to evaluate the ability of the alignment systems to deal with multilingual ontologies. It serves the purpose of evaluating the strength and weakness of a given system across languages. KEPLER uses a special technique to determine the equivalence between ontology entities described in different natural languages. We chose to use the English language as a pivot language. Indeed, the use of a pivot language ensures greater consistency of obtained translations since it starts

from the same text. In the *different ontologies* case, the method is placed fourth with a recall value of $0.31\%$, whereas in the *same ontologies* case, the method occupies the first place with a recall value of $0.52\%$.

### 2.4 Large Biomedical Ontologies and Phenotype

In the scalability register, this track consists of finding alignments between the Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). These ontologies are semantically rich and contain tens of thousands of classes. The Large BioMed Track consists of three matching problems, *i.e.*, (1) FMA-NCI matching problem, (2) FMA-SNOMED matching problem and (3) SNOMED-NCI matching problem. KEPLER handles large ontologies in two phases: the first phase consists on partitioning the ontologies into a set of blocks and the second phase selects two suitable blocks giving the highest value of similarity to be aligned. KEPLER treated (*Task 1: FMA-NCI small fragments*)`[P :0.96% R :0.83%]` and (*Task 3: FMA-SNOMED small fragments*) `[P :0.82% R :0.55%]`. In the Phenotype track, our method succeeds in processing only the DOID-ORDO sub-case by identifying 1824 matches.

## 3 Conclusion

In this paper, we briefly described the KEPLER method with comments of the results obtained according to the OAEI 2017 tracks, corresponding to the SEALS platform evaluation modalities. Several observations regarding these results were highlighted, in particular the impact of the elimination of any ontological resource on the similarity values.

## References


1. Diallo, G.: An effective method of large scale ontology matching. Journal of Biomedical Semantics **5** (2014) 44 (Electronic Edition)
2. Berners-Lee, T.: Designing the web for an open society. In: Proceedings of the 20th International Conference on World Wide Web (WWW2011), Hyderabad, India (2011) 3–4
3. Suchanek, F.M., Varde, A.S., Nayak, R., Senellart, P.: The hidden web, xml and semantic web: A scientific data management perspective. Computing Research Repository (2011) 534–537
4. Kachroudi, M., Zghal, S., Ben Yahia, S.: Ontopart: at the cross-roads of ontology partitioning and scalable ontology alignment systems. International Journal of Metadata, Semantics and Ontologies **8**(3) (2013) 215–225
5. Diallo, G.: Efficient building of local repository of distributed ontologies. In: Proceedings of the Seventh International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2011, Dijon, France, November 28 - December 1, 2011. (2011) 159–166
6. Dramé, K., Diallo, G., Delva, F., Dartigues, J., Mouillet, E., Salamon, R., Mougin, F.: Reuse of termino-ontological resources and text corpora for building a multilingual domain ontology: An application to alzheimer's disease. Journal of Biomedical Informatics **48** (2014) 171–182