We Divide, You Conquer:
From Large-scale Ontology Alignment to Manageable Subtasks with a Lexical Index and Neural Embeddings*

Ernesto Jiménez-Ruiz[1,2], Asan Agibetov[3], Matthias Samwald[3], Valerie Cross[4]

[1] The Alan Turing Institute, London, United Kingdom
[2] Department of Informatics, University of Oslo, Norway
[3] Section for Artificial Intelligence and Decision Support, Medical University of Vienna, Vienna, Austria
[4] Miami University, Oxford, OH 45056, United States

1 Introduction

An ontology *matching task* $\mathcal{MT}$ is composed of a pair of ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and possibly an associated *reference alignment* $\mathcal{M}^{RA}$. The objective of a matching task is to discover an overlapping of $\mathcal{O}_1$ and $\mathcal{O}_2$ in the form of an alignment $\mathcal{M}$. The *size or search space* of a matching task is typically bound to the size of the Cartesian product between the entities of the input ontologies. Large-scale ontology matching tasks still pose serious challenges to state-of-the-art ontology alignment systems [2]. In this paper we propose a novel method to effectively divide an input ontology matching task $\mathcal{MT}$ into several (independent) and more manageable (sub)tasks. This method relies on an efficient lexical index (as in LogMap [3]), a neural embedding model [4] and locality modules [5]. Unlike other state-of-the-art approaches, our method provides guarantees about the preservation of the coverage of the relevant ontology alignment.

2 Methods

The approach presented in this paper relies on an 'inverted' lexical index (we will refer to this index as Lexl), commonly used in information retrieval applications, and also used in ontology alignment systems like LogMap [3]. Lexl encodes the labels of all entities of the input ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, including their lexical variations (*e.g.*, preferred labels, synonyms), in the form of pairs *key-value* where the key is a set of words and the value is a set of entity identifiers[1] such that the set of words of the key appears in one of the entity labels. Table 1 shows a few example entries of Lexl for two ontologies.

2.1 Creating matching subtasks from Lexl

Deriving mappings from Lexl. Each entry in Lexl, after discarding entries pointing to only one ontology, is a source of candidate mappings. For instance the example in Table 1 suggests that there is a (potential) mapping between the entities $\mathcal{O}_1$:Serous_acinus and $\mathcal{O}_2$:Liver_acinus since they are associated to the same entry in Lexl $\{acinus\}$. The mappings derived from Lexl are not necessarily correct but will link lexically related

* An extended version of this paper is available in arXiv.org [1].
[1] The indexation module associates unique numerical identifiers to entity URIs.

Table 1: Inverted lexical index LexI (left) and entity index (right). Index values have been split into elements of $\mathcal{O}_1$ and $\mathcal{O}_2$. '-' indicates that the ontology does not contain entities for that index entry.

Index key	**Index value**	
	Entities $\mathcal{O}_1$	**Entities $\mathcal{O}_2$**
{ acinus }	7661,8171	118081
{ mesothelial, pleural }	19987	117237
{ hamate, lunate }	55518	-
{ feed, breast }	-	113578,111023

ID	**URI**
7661	$\mathcal{O}_1$:Serous_acinus
8171	$\mathcal{O}_1$:Hepatic_acinus
19987	$\mathcal{O}_1$:Mesothelial_cell_of_pleura
55518	$\mathcal{O}_1$:Lunate_facet_of_hamate
118081	$\mathcal{O}_2$:Liver_acinus
117237	$\mathcal{O}_2$:Pleural_Mesothelial_Cell
113578	$\mathcal{O}_2$:Breast_Feeding
111023	$\mathcal{O}_2$:Inability_To_Breast_Feed

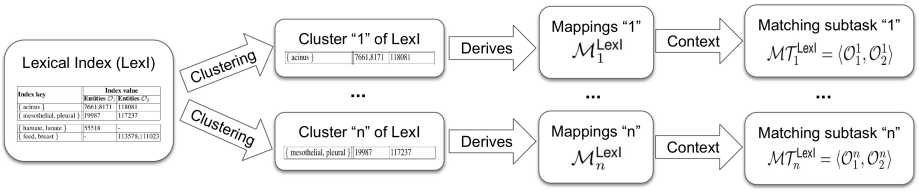


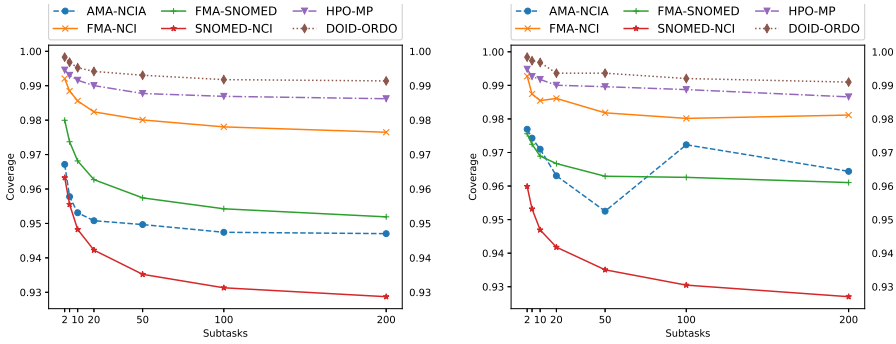
Fig. 1: Pipeline to extract matching subtasks from LexI.

entities. We refer to the set of all mappings suggested by LexI as $\mathcal{M}^{\mathsf{LexI}}$. Note that $\mathcal{M}^{\mathsf{LexI}}$ represents a manageable subset of the Cartesian product between the entities of the input ontologies. Mappings outside $\mathcal{M}^{\mathsf{LexI}}$ will rarely be discovered by standard matching systems as they typically rely on lexical similarity measures [2].

Context as matching task. Logic-based module extraction techniques compute ontology fragments that capture the meaning of an input signature (*e.g.*, set of entities) with respect to a given ontology. In this paper we rely on bottom-locality modules [5], which will be referred to as locality-modules or simply as modules. Locality modules play an important role in ontology alignment tasks. For example they provide the scope or context (*i.e.*, sets of *semantically related* entities [5]) for the entities in a given mapping or set of mappings. The context of the mappings $\mathcal{M}^{\mathsf{LexI}}$ derived from LexI leads to two ontology modules $\mathcal{O}_1^{\mathsf{LexI}}$ and $\mathcal{O}_2^{\mathsf{LexI}}$ from $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. $\mathcal{MT}^{\mathsf{LexI}} = \langle \mathcal{O}_1^{\mathsf{LexI}}, \mathcal{O}_2^{\mathsf{LexI}} \rangle$ is the (single) matching subtask derived from LexI.

The whole set of entries in LexI, however, may lead to a very large number of candidate mappings $\mathcal{M}^{\mathsf{LexI}}$ and, as a consequence, to large modules $\mathcal{O}_1^{\mathsf{LexI}}$ and $\mathcal{O}_2^{\mathsf{LexI}}$. These modules, although smaller than $\mathcal{O}_1$ and $\mathcal{O}_2$, can still be challenging for many ontology matching systems. A solution is to divide the entries in LexI into more than one cluster. Figure 1 shows an overview of the pipeline where LexI is split into n clusters and these clusters lead to n matching subtasks $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$.

Clustering strategies. We have implemented two clustering strategies which we refer to as: *naive* and *neural embedding*. The naive strategy implements a very simple algorithm that randomly splits the entries in LexI into a given number of clusters of the same size, while the neural embedding strategy aims at identifying more accurate clusters and relies on the StarSpace toolkit and its neural embedding model [4]. Applied to the lexical index LexI, the neural embedding model learns vector representations for the individual words in the index keys, and for the individual entity identifiers in the



(a) Naive strategy (b) Neural embedding strategy

Fig. 2: Coverage ratio with respect to the number of matching subtasks in $\mathcal{D}^n_{\mathcal{MT}}$.

index values. Since an index key is a set of words (see Table 1), we use the *mean vector* representation of the vectors associated to each word. Based on these *aggregated* neural embeddings we then perform standard clustering with the K-means algorithm.

3 Evaluation

In this section we provide empirical evidence of the suitability of the presented approach to divide an ontology matching task.[2] We rely on the datasets of the Ontology Alignment Evaluation Initiative (OAEI), more specifically, on the matching tasks provided in the *anatomy* (AMA-NCIA), *largebio* (FMA-NCI, FMA-SNOMED, SNOMED-NCI) and *phenotype* (HPO-MP, DOID-ORDO) tracks.

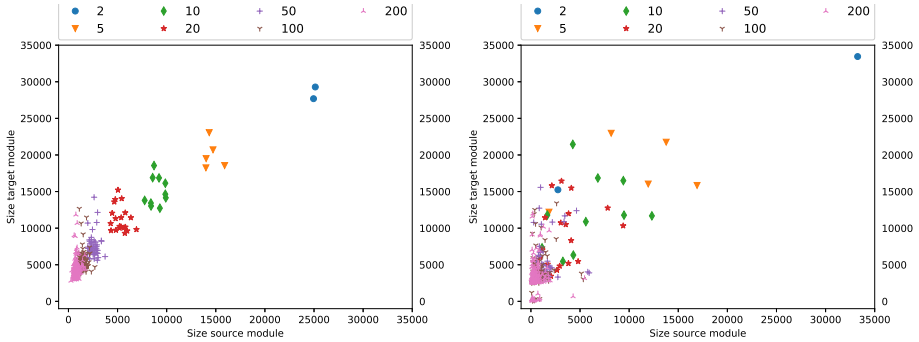
Adequacy of clustering strategies. We have evaluated the clustering strategies in terms of the coverage with respect to the OAEI reference alignments[3] and the size of the ontologies in the matching subtasks. We have compared the two strategies for different number of matching subtasks $n \in \{2, 5, 10, 20, 50, 100, 200\}$. Figure 2 shows the coverage of the different divisions $\mathcal{D}^n_{\mathcal{MT}}$ for the naive (left) and neural embedding (right) strategies. The results are very good and, in the worst case, approx. 93% of the available reference mappings in SNOMED-NCI are *covered* by the matching subtasks in $\mathcal{D}^{200}_{\mathcal{MT}}$.

The scatter plots in Figure 3 visualize, for the FMA-NCI case, the size of the source modules against the size of the target modules for the matching subtasks in each division $\mathcal{D}^n_{\mathcal{MT}}$. For instance, the (orange) triangles represent points $\big(|Sig(\mathcal{O}^i_1)|, |Sig(\mathcal{O}^i_2)|\big)$ being $\mathcal{O}^i_1$ and $\mathcal{O}^i_2$ the ontologies (with i=1,...,5) in the matching subtasks of $\mathcal{D}^5_{\mathcal{MT}}$. The naive strategy leads to rather balanced and similar tasks for each division $\mathcal{D}^n_{\mathcal{MT}}$, while the neural embedding strategy has more variability in the size of the tasks within a given division $\mathcal{D}^n_{\mathcal{MT}}$. Nonetheless, on average, the size of the matching subtasks in the neural embedding strategy are significantly smaller than the ones in the naive strategy.

Evaluation of OAEI systems. We have selected the following four systems from the latest OAEI campaigns: Mamba, FCA-Map, KEPLER, and POMap. These systems were unable to complete, given some computational constraints, some OAEI tasks. Table 2 shows the obtained results with different divisions $\mathcal{D}^n_{\mathcal{MT}}$ computed by the naive and

[2] Extended evaluation material in [1] and `https://doi.org/10.5281/zenodo.1214149`

[3] A mapping is *covered* if it can (potentially) be discovered in one or more matching subtasks.



(a) Naive strategy (b) Neural embedding strategy

Fig. 3: Source and target module sizes in the computed subtasks for FMA-NCI.

Table 2: Evaluation of systems that failed to complete OAEI tasks in 2015-2017.

Tool	Task	Year	Matching subtasks	Naive strategy				Neural embedding strategy			
				P	R	F	t (h)	P	R	F	t (h)
Mamba	Anatomy	2015	20	0.88	0.63	0.73	2.3	0.89	0.62	0.73	1.0
			50	0.88	0.62	0.73	2.4	0.89	0.62	0.73	1.0
FCA-Map	FMA-NCI	2016	20	0.56	0.90	0.72	4.4	0.62	0.90	0.73	3.1
			50	0.58	0.90	0.70	4.1	0.60	0.90	0.72	3.0
KEPLER	FMA-NCI	2017	20	0.45	0.82	0.58	8.9	0.48	0.80	0.60	4.3
			50	0.42	0.83	0.56	6.9	0.46	0.80	0.59	3.8
POMap	FMA-NCI	2017	20	0.54	0.83	0.66	11.9	0.56	0.79	0.66	5.7
			50	0.55	0.83	0.66	8.8	0.57	0.79	0.66	4.1

neural embedding strategies, for the OAEI tasks that the selected systems failed to compute results. For example, Mamba was able to complete the OAEI 2015 Anatomy track with divisions $\mathcal{D}^{20}_{\mathcal{MT}}$ and $\mathcal{D}^{50}_{\mathcal{MT}}$ involving 20 and 50 matching subtasks, respectively. The subtasks generated by the neural embedding strategy lead to much lower times.

The results are encouraging and suggest that the proposed method to divide an ontology matching task *(i)* leads to a very limited information loss (*i.e.*, high coverage), and *(ii)* enables new systems to complete large-scale OAEI tasks.

References

1. Jimenez-Ruiz, E., Agibetov, A., Samwald, M., Cross, V.: Breaking-down the Ontology Alignment Task with a Lexical Index and Neural Embeddings. arXiv (2018) Available from: `https://arxiv.org/abs/1805.12402`.
2. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Trans. Knowl. Data Eng. **25**(1) (2013) 158–176
3. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-Based and Scalable Ontology Matching. In: International Semantic Web Conference. (2011) 273–288
4. Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., Weston, J.: StarSpace: Embed All The Things! arXiv (2017)
5. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. Artif. Intell. Res. **31** (2008) 273–318