City Research Online

City St George's, University of London

Citation: Jimenez-Ruiz, E., Agibetov, A., Chen, J., Samwald, M. & Cross, V. (2020). Dividing the Ontology Alignment Task with Semantic Embeddings and Logic-based Modules. Paper presented at the ECAI 2020, 8-12 Jun 2020, Santiago, Chile. doi: 10.3233/FAIA200167

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/23742/

Link to published version: https://doi.org/10.3233/FAIA200167

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

Dividing the Ontology Alignment Task with Semantic Embeddings and Logic-based Modules

Ernesto Jiménez-Ruiz[1] and **Asan Agibetov**[2] and **Jiaoyan Chen**[3] and **Matthias Samwald**[2] and **Valerie Cross**[4]

Abstract. Large ontologies still pose serious challenges to state-of-the-art ontology alignment systems. In this paper we present an approach that combines a neural embedding model and logic-based modules to accurately divide an input ontology matching task into smaller and more tractable matching (sub)tasks. We have conducted a comprehensive evaluation using the datasets of the Ontology Alignment Evaluation Initiative. The results are encouraging and suggest that the proposed method is adequate in practice and can be integrated within the workflow of systems unable to cope with very large ontologies.

1 Introduction

The problem of (semi-)automatically computing an alignment between independently developed ontologies has been extensively studied in the last years. As a result, a number of sophisticated ontology alignment systems currently exist [44, 15].[5] The Ontology Alignment Evaluation Initiative[6] (OAEI) [3, 4] has played a key role in the benchmarking of these systems by facilitating their comparison on the same basis and the reproducibility of the results. The OAEI includes different tracks organised by different research groups. Each track contains one or more matching tasks involving small-size (*e.g.*, conference), medium-size (*e.g.*, anatomy), large (*e.g.*, phenotype) or very large (*e.g.*, largebio) ontologies. Some tracks only involve matching at the terminological level (*e.g.*, concepts and properties) while other tracks also expect an alignment at the assertional level (*i.e.*, instance data).

Large ontologies still pose serious challenges to ontology alignment systems. For example, several systems participating in the *largebio track* were unable to complete the largest tasks during the latest OAEI campaigns.[7] These systems typically use advanced alignment methods and are able to cope with small and medium size ontologies with competitive results, but fail to complete large tasks in a given time frame or with the available resources such as memory.

There have been several efforts in the literature to divide the ontology alignment task (*e.g.*, [20, 22]). These approaches, however, have not been successfully evaluated with very large ontologies, failing to scale or producing partitions of the ontologies leading to information loss [42]. In this paper we propose a novel method to accurately divide the matching task into several independent, smaller and manageable (sub)tasks, so as to scale systems that cannot cope with very large ontologies.[8] Unlike state-of-the-art approaches, our method: *(i)* preserves the coverage of the relevant ontology alignments while keeping manageable matching subtasks; *(ii)* provides a formal notion of matching subtask and semantic context; *(iii)* uses neural embeddings to compute an accurate division by learning semantic similarities between words and ontology entities according to the ontology alignment task at hand; *(iv)* computes self-contained (logical) modules to guarantee the inclusion of the (semantically) relevant information required by an alignment system; and *(v)* has been successfully evaluated with very large ontologies.

2 Preliminaries

A *mapping* (also called *match*) between entities[9] of two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ is typically represented as a 4-tuple $\langle e_1, e_2, r, c\rangle$ where e_1 and e_2 are entities of $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively; r is a semantic relation, typically one of $\{\sqsubseteq, \sqsupseteq, \equiv\}$; and c is a confidence value, usually, a real number within the interval $(0, 1]$. For simplicity, we refer to a mapping as a pair $\langle e_1, e_2\rangle$. An ontology *alignment* is a set of mappings $\mathcal{M}$ between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$.

An ontology *matching task* $\mathcal{MT}$ is composed of a pair of ontologies $\mathcal{O}_1$ (typically called source) and $\mathcal{O}_2$ (typically called target) 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: $|Sig(\mathcal{O}_1)| \times |Sig(\mathcal{O}_2)|$, where $Sig(\mathcal{O})$ denotes the signature (*i.e.*, entities) of $\mathcal{O}$ and $|\cdot|$ denotes the size of a set.

An ontology *matching system* is a program that, given as input a matching task $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2\rangle$, generates an ontology alignment $\mathcal{M}^S$.[10] The standard evaluation measures for an alignment $\mathcal{M}^S$ are *precision* (P), *recall* (R) and *f-measure* (F) computed against a reference alignment $\mathcal{M}^{RA}$ as follows:

$$P = \frac{|\mathcal{M}^S \cap \mathcal{M}^{RA}|}{|\mathcal{M}^S|},\ R = \frac{|\mathcal{M}^S \cap \mathcal{M}^{RA}|}{|\mathcal{M}^{RA}|},\ F = 2 \cdot \frac{P \cdot R}{P + R} \quad (1)$$

[1] City, University of London, UK, email: ernesto.jimenez-ruiz@city.ac.uk; and Department of Informatics, University of Oslo, Norway
[2] Section for Artificial Intelligence and Decision Support, Medical University of Vienna, Vienna, Austria
[3] Department of Computer Science, University of Oxford, UK
[4] Miami University, Oxford, OH 45056, United States
[5] Ontology matching surveys and approaches: http://ontologymatching.org/
[6] OAEI evaluation campaigns: http://oaei.ontologymatching.org/
[7] Largebio track: http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/
[8] A preliminary version of this work has been published in arXiv [25] and in the Ontology Matching workshop [26].
[9] In this work we accept any input ontology in the OWL 2 language [18]. We refer to (OWL 2) concepts, properties and individuals as entities.
[10] Typically automatic, although there are systems that also allow human interaction [32].

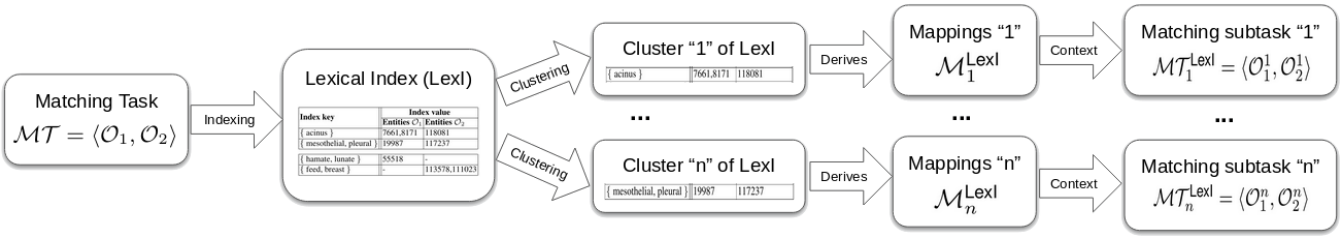


Figure 1: Pipeline to divide a given matching task $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$.

2.1 Problem definition and quality measures

We denote *division* of an ontology matching task $\mathcal{MT}$, composed by the ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, as the process of finding n matching subtasks $\mathcal{MT}_i = \langle \mathcal{O}_1^i, \mathcal{O}_2^i \rangle$ (with i=1,...,n), where $\mathcal{O}_1^i \subset \mathcal{O}_1$ and $\mathcal{O}_2^i \subset \mathcal{O}_2$.

Size of the division. The size of each matching subtask is smaller than the original task and thus reduces the search space. Let $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1, \ldots, \mathcal{MT}_n\}$ be the division of a matching task $\mathcal{MT}$ into n subtasks. The *size ratio* of the subtasks $\mathcal{MT}_i$ and $\mathcal{D}^n_{\mathcal{MT}}$ with respect to the original matching task size is computed as follows:

$$\mathsf{SizeRatio}(\mathcal{MT}_i, \mathcal{MT}) = \frac{|Sig(\mathcal{O}_1^i)| \times |Sig(\mathcal{O}_2^i)|}{|Sig(\mathcal{O}_1)| \times |Sig(\mathcal{O}_2)|} \quad (2)$$

$$\mathsf{SizeRatio}(\mathcal{D}^n_{\mathcal{MT}}, \mathcal{MT}) = \sum_{i=1}^{n} \mathsf{SizeRatio}(\mathcal{MT}_i, \mathcal{MT}) \quad (3)$$

The ratio $\mathsf{SizeRatio}(\mathcal{MT}_i, \mathcal{MT})$ is less than 1.0 while the aggregation $\sum_{i=1}^{n} \mathsf{SizeRatio}(\mathcal{MT}_i, \mathcal{MT})$, being n the number of matching subtasks, can be greater than 1.0 as matching subtasks depend on the division technique and may overlap.

Alignment coverage. The division of the matching task aims at preserving the target outcomes of the original matching task. The *coverage* is calculated with respect to a relevant alignment $\mathcal{M}$, possibly the reference alignment $\mathcal{M}^{RA}$ of the matching task if it exists, and indicates whether that alignment can still be (potentially) discovered with the matching subtasks. The formal notion of coverage is given in Definitions 1 and 2.

Definition 1 (Coverage of a matching task) *Let $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$ be a matching task and $\mathcal{M}$ an alignment. We say that a mapping $m = \langle e_1, e_2 \rangle \in \mathcal{M}$ is covered by the matching task if $e_1 \in Sig(\mathcal{O}_1)$ and $e_2 \in Sig(\mathcal{O}_2)$. The coverage of $\mathcal{MT}$ w.r.t. $\mathcal{M}$ (denoted as $\mathsf{Coverage}(\mathcal{MT}, \mathcal{M})$) represents the set of mappings $\mathcal{M}' \subseteq \mathcal{M}$ covered by $\mathcal{MT}$.*

Definition 2 (Coverage of the matching task division) *Let $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1, \ldots, \mathcal{MT}_n\}$ be the result of dividing a matching task $\mathcal{MT}$ and $\mathcal{M}$ an alignment. We say that a mapping $m \in \mathcal{M}$ is covered by $\mathcal{D}^n_{\mathcal{MT}}$ if m is at least covered by one of the matching subtask $\mathcal{MT}_i$ (with i=1,...,n) as in Definition 1. The coverage of $\mathcal{D}^n_{\mathcal{MT}}$ w.r.t. $\mathcal{M}$ (denoted as $\mathsf{Coverage}(\mathcal{D}^n_{\mathcal{MT}}, \mathcal{M})$) represents the set of mappings $\mathcal{M}' \subseteq \mathcal{M}$ covered by $\mathcal{D}^n_{\mathcal{MT}}$. The coverage is given as a ratio with respect to the (covered) alignment:*

$$\mathsf{CoverageRatio}(\mathcal{D}^n_{\mathcal{MT}}, \mathcal{M}) = \frac{|\mathsf{Coverage}(\mathcal{D}^n_{\mathcal{MT}}, \mathcal{M})|}{|\mathcal{M}|} \quad (4)$$

3 Methods

In this section we present our approach to compute a division $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1, \ldots, \mathcal{MT}_n\}$ given a matching task $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$ and the number of target subtasks n. We rely on locality ontology modules to extract self-contained modules of the input ontologies. The module extraction and task division is tailored to the ontology alignment task at hand by embedding the contextual semantics of a (combined) inverted index of the ontologies in the matching task.

Figure 1 shows an overview of our approach. *(i)* The ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ are indexed using the lexical index Lexl (see Section 3.2); *(ii)* Lexl is divided into clusters based on the semantic embeddings of its entries (see Section 3.4); *(iii)* entries in those clusters derive potential mapping sets (see Section 3.3); and *(iv)* the context of these mapping sets lead to matching subtasks (see Sections 3.1 and 3.3). Next, we elaborate on the methods behind these steps.

3.1 Locality modules and context

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. That is, a module contains the context (*i.e.*, sets of *semantically related* entities) of the input signature. In this paper we rely on bottom-locality modules [13, 29], which will be referred to as locality-modules or simply as modules. These modules include the ontology axioms required to describe the entities in the signature. Locality-modules compute self-contained ontologies and are tailored to tasks that require reusing a fragment of an ontology. Please refer to [13, 29] for further details.

Locality-modules play an key role in our approach as they provide the context for the entities in a given mapping or set of mappings as formally presented in Definition 3.

Definition 3 (Context of a mapping and an alignment) *Let $m = \langle e_1, e_2 \rangle$ be a mapping between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. We define the context of m (denoted as $\mathsf{Context}(m, \mathcal{O}_1, \mathcal{O}_2)$) as a pair of locality modules $\mathcal{O}_1' \subseteq \mathcal{O}_1$ and $\mathcal{O}_2' \subseteq \mathcal{O}_2$, where $\mathcal{O}_1'$ and $\mathcal{O}_2'$ include the semantically related entities to e_1 and e_2, respectively. Similarly, the* context *for an alignment $\mathcal{M}$ between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ is denoted as $\mathsf{Context}(\mathcal{M}, \mathcal{O}_1, \mathcal{O}_2) = \langle \mathcal{O}_1', \mathcal{O}_2' \rangle$, where $\mathcal{O}_1'$ and $\mathcal{O}_2'$ are modules including the semantically related entities for the entities $e_1 \in Sig(\mathcal{O}_1)$ and $e_2 \in Sig(\mathcal{O}_2)$ in each mapping $m = \langle e_1, e_2 \rangle \in \mathcal{M}$.*

Intuitively, as the context of an alignment (*i.e.*, $\mathsf{Context}(\mathcal{M}, \mathcal{O}_1, \mathcal{O}_2) = \langle \mathcal{O}_1', \mathcal{O}_2' \rangle$) semantically characterises the entities involved in that alignment, a matching task $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$ can be reduced to the task $\mathcal{MT}^{\mathcal{M}}_{\mathcal{O}_1\text{-}\mathcal{O}_2} = \langle \mathcal{O}_1', \mathcal{O}_2' \rangle$ without information loss in terms of finding $\mathcal{M}$ (*i.e.*, $\mathsf{Coverage}(\mathcal{MT}^{\mathcal{M}}_{\mathcal{O}_1\text{-}\mathcal{O}_2}, \mathcal{M}) = \mathcal{M}$). For example, in the small OAEI *largebio* tasks [3, 4] systems are given the context of the reference alignment as a (reduced) matching task (*e.g.*, $\mathcal{MT}^{RA}_{\text{fma-nci}} = \mathsf{Context}(\mathcal{M}^{RA}_{\text{fma-nci}}, \mathcal{O}_{\text{FMA}}, \mathcal{O}_{\text{NCI}}) = \langle \mathcal{O}'_{\text{FMA}}, \mathcal{O}'_{\text{NCI}} \rangle$), instead of the whole FMA and NCI ontologies.

Table 1: Inverted lexical index LexI. For readability, 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 entry.

#	Index key	Index value	
		Entities $\mathcal{O}_1$	**Entities $\mathcal{O}_2$**
1	{ disorder }	$\mathcal{O}_1$:Disorder_of_pregnancy, $\mathcal{O}_1$:Disorder_of_stomach	$\mathcal{O}_2$:Pregnancy_Disorder
2	{ disorder, pregnancy }	$\mathcal{O}_1$:Disorder_of_pregnancy	$\mathcal{O}_2$:Pregnancy_Disorder
3	{ carcinoma, basaloid }	$\mathcal{O}_1$:Basaloid_carcinoma	$\mathcal{O}_2$:Basaloid_Carcinoma, $\mathcal{O}_2$:Basaloid_Lung_Carcinoma
4	{ follicul, thyroid, carcinom }	$\mathcal{O}_1$:Follicular_thyroid_carcinoma	$\mathcal{O}_2$:Follicular_Thyroid_carcinoma
5	{ hamate, lunate }	$\mathcal{O}_1$:Lunate_facet_of_hamate	-

3.2 Indexing the ontology vocabulary

We rely on a semantic inverted index (we will refer to this index as LexI). This index maps sets of words to the entities where these words appear. LexI 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 *key-value* pairs where the key is a set of words and the value is a set of entities such that the set of words of the key appears in (one of) the entity labels. Similar indexes are commonly used in information retrieval applications [11], Entity Resolution systems [40], and also exploited in ontology alignment systems (*e.g.*, LogMap [27], ServOMap [14] and AML [16]) to reduce the search space and enhance the matching process. Table 1 shows a few example entries of LexI for two input ontologies.

LexI is created as follows. *(i)* Each label associated to an ontology entity is split into a set of words; for example, the label "Lunate facet of hamate" is split into the set {"lunate", "facet", "of", "hamate"}. *(ii)* Stop-words are removed from the set of words. *(iii)* Stemming techniques are applied to each word (*i.e.*, {"lunat", "facet", "hamat"}). *(iv)* Combinations of subsets of words also serve as keys in LexI; for example, {"lunat", "facet"}, {"hamat", "lunat"} and so on.[11] *(v)* Entities leading to the same (sub)set of words are associated to the same key in LexI, for example {"disorder"} is associated with three entities. Finally, *(vi)* entries in LexI pointing to entities of only one ontology or associated to a number of entities larger than α are not considered.[12] Note that a single entity label may lead to several entries in LexI, and each entry in LexI points to one or more entities.

3.3 Covering matching subtasks

Each entry (*i.e.*, a *key-value* pair) in LexI is a source of candidate mappings. For instance, the example in Table 1 suggests that there is a candidate mapping $m = \langle \mathcal{O}_1$:Disorder_of_stomach, $\mathcal{O}_2$:Pregnancy_disorder$\rangle$ since these entities are associated to the {*"disorder"*} entry in LexI. These mappings are not necessarily correct but will link lexically-related entities, that is, those entities sharing at least one word among their labels (*e.g.*, "disorder"). Given a subset of entries or rows of LexI (*i.e.*, $l \subseteq$ LexI), the function $\mathsf{Mappings}(l) = \mathcal{M}^l$ provides the set of mappings derived from l. We refer to the set of all (potential) mappings suggested by LexI (*i.e.*, $\mathsf{Mappings}(\mathsf{LexI})$) as $\mathcal{M}^{\mathsf{LexI}}$. $\mathcal{M}^{\mathsf{LexI}}$ represents a manageable subset of the Cartesian product between the entities of the input ontologies. For example, LexI suggest around 2×10^4 potential mappings for the matching task $\mathcal{MT}_{\text{fma-nci}} = \langle \mathcal{O}_{\text{FMA}}, \mathcal{O}_{\text{NCI}} \rangle$, while the Cartesian product between $\mathcal{O}_{\text{FMA}}$ and $\mathcal{O}_{\text{NCI}}$ involves more than 5×10^9 mappings.

Since standard ontology alignment systems rarely discover mappings outside $\mathcal{M}^{\mathsf{LexI}}$, the context of $\mathcal{M}^{\mathsf{LexI}}$ (recall Definition 3) can be seen as a reduced matching task $\mathcal{MT}^{\mathsf{LexI}} = \mathsf{Context}(\mathcal{M}^{\mathsf{LexI}}, \mathcal{O}_1, \mathcal{O}_2) = \langle \mathcal{O}_1^{\mathsf{LexI}}, \mathcal{O}_2^{\mathsf{LexI}} \rangle$ of the original task $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$. However, the modules $\mathcal{O}_1^{\mathsf{LexI}}$ and $\mathcal{O}_2^{\mathsf{LexI}}$, although smaller than $\mathcal{O}_1$ and $\mathcal{O}_2$, can still be challenging for many ontology matching systems. A solution is to divide or cluster the entries in LexI to lead to several tasks involving smaller ontology modules.

Definition 4 (Matching subtasks from LexI) *Let $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$ be a matching task,* LexI *the inverted index of the ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, and $\{l_1, \ldots, l_n\}$ a set of n clusters of entries in* LexI. *We denote the set of matching subtasks from* LexI *as $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$ where each cluster l_i leads to the matching subtask $\mathcal{MT}_i^{\mathsf{LexI}} = \langle \mathcal{O}_1^i, \mathcal{O}_2^i \rangle$, such that $\mathsf{Mappings}(l_i) = \mathcal{M}_i^{\mathsf{LexI}}$ is the set of mappings suggested by the* LexI *entries in l_i (i.e.,* key-value *pairs) and $\mathcal{O}_1^i$ and $\mathcal{O}_2^i$ represent the context of $\mathcal{M}_i^{\mathsf{LexI}}$ w.r.t. $\mathcal{O}_1$ and $\mathcal{O}_2$.*

Quality of the matching subtasks. The matching subtasks in Definition 4 rely on LexI and the notion of context, thus it is expected that the tasks in $\mathcal{D}^n_{\mathcal{MT}}$ will cover most of the mappings $\mathcal{M}^S$ that a matching system can compute, that is $\mathsf{CoverageRatio}(\mathcal{D}^n_{\mathcal{MT}}, \mathcal{M}^S)$ will be close to 1.0. Furthermore, the use of locality modules to compute the context guarantees the extraction of matching subtasks that are suitable to ontology alignment systems in terms of preservation of the logical properties of the given signature.

Intuitively each cluster of LexI will lead to a smaller matching task $\mathcal{MT}_i^{\mathsf{LexI}}$ (with respect to both $\mathcal{MT}^{\mathsf{LexI}}$ and $\mathcal{MT}$) in terms of search space. Hence $\mathsf{SizeRatio}(\mathcal{MT}_i^{\mathsf{LexI}}, \mathcal{MT})$ will be smaller than 1.0. The overall aggregation of ratios (cf. Equation 3) depends on the clustering strategy of the entries in LexI and it is also expected to be smaller than 1.0.

Reducing the search space in each matching subtask $\mathcal{MT}_i^{\mathsf{LexI}}$ has the potential of enabling the evaluation of systems that cannot cope with the original matching task $\mathcal{MT}$ in a given time-frame or with (limited) computational resources.

3.4 Semantic embeddings

We use a *semantic embedding* approach to identify, given n, a set of clusters of entries $\{l_1, \ldots, l_n\}$ from LexI. As in Definition 4, these clusters lead to the set of matching subtasks $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$. The *semantic embeddings* aim at representing into the same (vector) space the features about the relationships among words and ontology entities that occur in LexI. Hence, words and entities that belong to similar semantic contexts will typically have similar vector representations.

Embedding model. Our approach currently relies on the StarSpace toolkit[13] and its neural embedding model [49] to learn *embeddings* for the words and ontology entities in LexI. We adopt the

[11] In order to avoid a combinatorial blow-up, the number of computed subsets of words is limited.

[12] In the experiments we used $\alpha = 60$.

[13] StarSpace: https://github.com/facebookresearch/StarSpace

Table 2: Matching tasks. AMA: Adult Mouse Anatomy. DOID: Human Disease Ontology. FMA: Foundational Model of Anatomy. HPO: Human Phenotype Ontology. MP: Mammalian Phenotype. NCI: National Cancer Institute Thesaurus. NCIA: Anatomy fragment of NCI. ORDO: Orphanet Rare Disease Ontology. SNOMED CT: Systematized Nomenclature of Medicine – Clinical Terms. Phenotype ontologies downloaded from BioPortal. For all tracks we use the consensus with vote=3 as system mappings $\mathcal{M}^S$. The Phenotype track does not have a gold standard so a consensus alignment with vote=2 is used as reference.

OAEI track	Source of $\mathcal{M}^{RA}$	Source of $\mathcal{M}^S$	Task	Ontology	Version	Size (classes)
Anatomy	Manually created [10]	Consensus (vote=3)	AMA-NCIA	AMA	v.2007	2,744
				NCIA	v.2007	3,304
Largebio	UMLS-Metathesaurus [28]	Consensus (vote=3)	FMA-NCI	FMA	v.2.0	78,989
			FMA-SNOMED	NCI	v.08.05d	66,724
			SNOMED-NCI	SNOMED CT	v.2009	306,591
Phenotype	Consensus alignment (vote=2) [21]	Consensus (vote=3)	HPO-MP	HPO	v.2016	11,786
				MP	v.2016	11,721
			DOID-ORDO	DOID	v.2016	9,248
				ORDO	v.2016	12,936

TagSpace [48] training setting of StarSpace. Applied to our setting, StarSpace learns associations between a set of words (*i.e.*, keys in LexI) and a set of relevant ontology entities (*i.e.*, values in LexI). The StarSpace model is trained by assigning a d-dimensional vector to each of the relevant features (*e.g.*, the individual words and the ontology entities in LexI). Ultimately, the look-up matrix (the matrix of embeddings - latent vectors) is learned by minimising the loss function in Equation 5.

$$\sum_{\substack{(w,e)\in E^+,\\ e^-\in E^-}} L^{batch}(sim(\boldsymbol{v}_w,\boldsymbol{v}_e), sim(\boldsymbol{v}_w,\boldsymbol{v}_{e_1^-}),\ldots,sim(\boldsymbol{v}_w,\boldsymbol{v}_{e_j^-})) \quad (5)$$

In this loss function we compare positive samples with negative samples. Hence we need to indicate the generator of positive pairs $(w,e) \in E^+$ (in our setting those are *word-entity* pairs from LexI) and the generator of negative entries $e^- \in E^-$ (in our case we sample from the list of entities in the values of LexI). StarSpace follows the strategy by Mikolov et al. [36] and selects a random subset of j negative examples for each batch update. Note that we tailor the generators to the alignment task by sampling from LexI. The similarity function sim operates on d-dimensional vectors (*e.g.*, $\boldsymbol{v}_w$, $\boldsymbol{v}_e$ and $\boldsymbol{v}_{e^-}$), in our case we use the standard dot product in Euclidean space.

Clustering strategy. The semantic embedding of each entry $\varepsilon = (K,V) \in$ LexI is calculated by concatenating *(i)* the mean vector representation of the vectors associated to each word in the key K, with *(ii)* the mean vector of the vectors of the ontology entities in the value V, as in Equation 6, where $\oplus$ represents the concatenation of two vectors, $\boldsymbol{v}_w$ and $\boldsymbol{v}_e$ represents d-dimensional vector embeddings learnt by StarSpace, and $\boldsymbol{v}_\varepsilon$ is a $(2*d)$-dimension vector.

$$\boldsymbol{v}_\varepsilon = \frac{1}{|K|}\sum_{w\in K}\boldsymbol{v}_w \oplus \frac{1}{|V|}\sum_{e\in V}\boldsymbol{v}_e \quad (6)$$

Based on the embeddings $\boldsymbol{v}_\varepsilon$ we then perform standard clustering with the K-means algorithm to obtain the clusters of LexI entries $\{l_1,\ldots,l_n\}$. For example, following our approach, in the example of Table 1 entries in rows 1 and 2 (respectively 3 and 4) would belong to the same cluster.

Suitability of the embedding model. Although we could have followed other embedding strategies, we advocated to learn new entity embeddings with StarSpace for the following reasons: *(i)* ontologies, particularly in the biomedical domain, may bring specialised vocabulary that is not fully covered by precomputed word embeddings; *(ii)* to embed not only words but also concepts of both ontologies; and *(iii)* to obtain embeddings tailored to the ontology alignment task (*i.e.*, to learn similarities among words and concepts dependant on the task). StarSpace provides the required functionalities to embed the semantics of LexI and identify accurate clusters. Precise clusters will lead to smaller matching tasks, and thus, to a reduced global size of the computed division of the matching task $\mathcal{D}^n_{\mathcal{MT}}$ (cf. Equation 3).

4 Evaluation

In this section we provide empirical evidence to support the suitability of the proposed method to divide the ontology alignment task. We rely on the datasets of the Ontology Alignment Evaluation Initiative (OAEI) [3, 4], more specifically, on the matching tasks provided in the *anatomy*, *largebio* and *phenotype* tracks. Table 2 provides an overview of these OAEI tasks and the related ontologies and mapping sets.

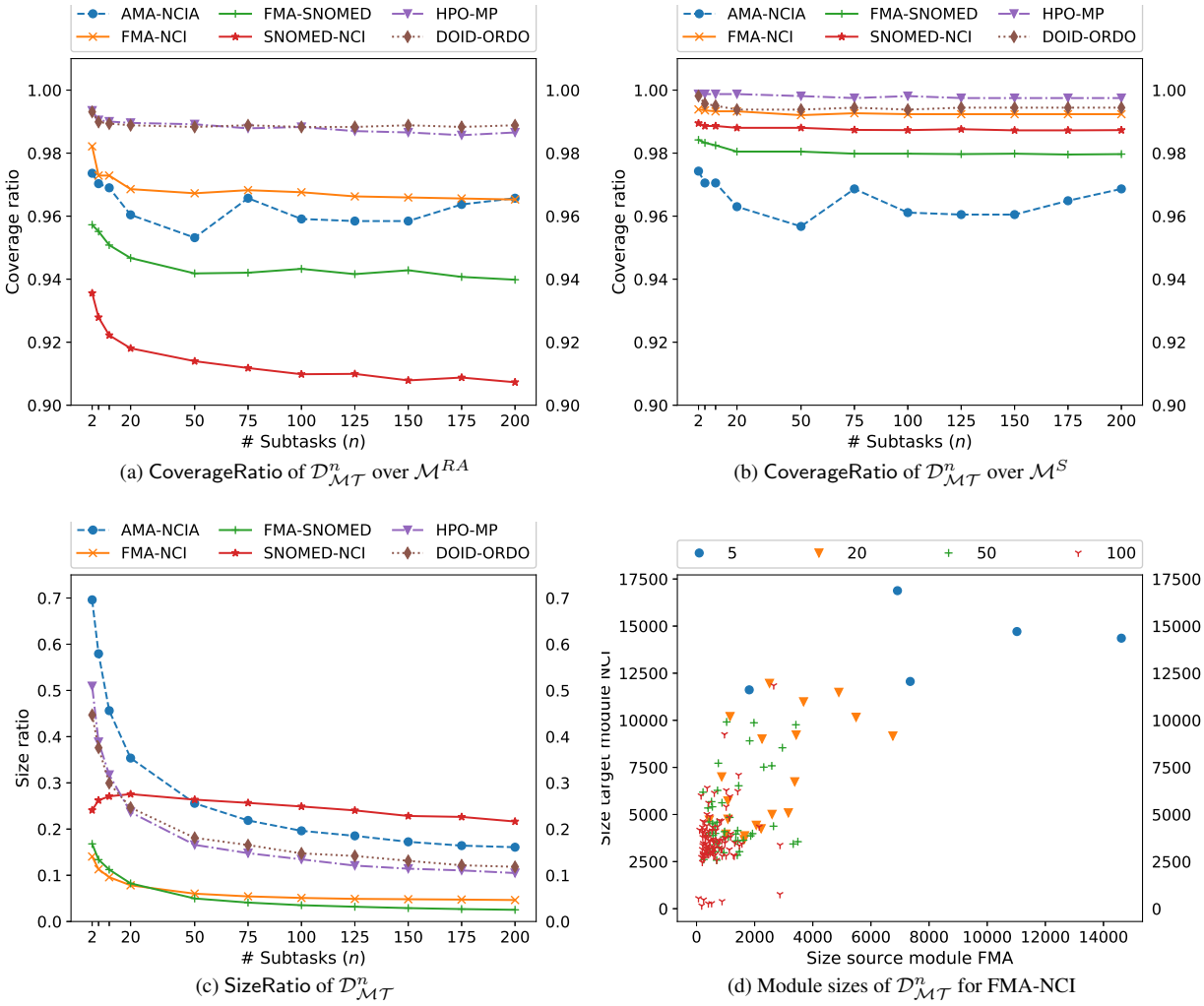
The methods have been implemented in Java[14] and Python[15] (neural embedding strategy), tested on a Ubuntu Laptop with an Intel Core i9-8950HK CPU@2.90GHz and allocating up to $25Gb$ of RAM. Datasets, matching subtasks, computed mappings and other supporting resources are available in the *Zenodo* repository [24]. For all of our experiments we used the following StarSpace hyperparameters: `-trainMode 0 -similarity dot --epoch 100 --dim 64`.

4.1 Adequacy of the division approach

We have evaluated the adequacy of our division strategy in terms of coverage (as in Equation 4) and size (as in Equation 3) of the resulting division $\mathcal{D}^n_{\mathcal{MT}}$ for each of the matching task in Table 2.

Coverage ratio. Figures 2a and 2b shows the coverage of the different divisions $\mathcal{D}^n_{\mathcal{MT}}$ with respect to the reference alignment and system computed mappings, respectively. As system mappings we have used the consensus alignment with vote=3, that is, mappings that have been voted by at least 3 systems in the last OAEI campaigns. The overall coverage results are encouraging: *(i)* the divisions $\mathcal{D}^n_{\mathcal{MT}}$ cover over 94% of the reference alignments for all tasks, with the exception of the SNOMED-NCI case where coverage ranges from 0.94

[14] Java codes: https://github.com/ernestojimenezruiz/logmap-matcher
[15] Python codes: https://github.com/plumdeq/neuro-onto-part



(a) CoverageRatio of $\mathcal{D}^n_{\mathcal{MT}}$ over $\mathcal{M}^{RA}$

(b) CoverageRatio of $\mathcal{D}^n_{\mathcal{MT}}$ over $\mathcal{M}^{S}$

(c) SizeRatio of $\mathcal{D}^n_{\mathcal{MT}}$

(d) Module sizes of $\mathcal{D}^n_{\mathcal{MT}}$ for FMA-NCI

Figure 2: Quality measures of $\mathcal{D}^n_{\mathcal{MT}}$ with respect to the number of matching subtasks n.

to 0.90; *(ii)* when considering system mappings, the coverage for all divisions is over 0.98 with the exception of AMA-NCIA, where it ranges from 0.956 to 0.974; *(iii)* increasing the number of divisions n tends to *slightly* decrease the coverage in some of the test cases, this is an expected behaviour as the computed divisions include different semantic contexts (*i.e.*, locality modules) and some relevant entities may fall out the division; finally *(iv)* as shown in [42], the results in terms of coverage of state-of-the-art partitioning methods (*e.g.*, [22, 20]) are very low for the OAEI *largebio* track (0.76, 0.59 and 0.67 as the best results for FMA-NCI, FMA-SNOMED and SNOMED-NCI, respectively), thus, making the obtained results even more valuable.

Size ratio. The results in terms of the size (*i.e.*, search space) of the selected divisions $\mathcal{D}^n_{\mathcal{MT}}$ are presented in Figure 2c. The search space is improved with respect to the original $\mathcal{MT}$ for all the cases, getting as low as 5% of the original matching task size for the FMA-NCI and FMA-SNOMED cases. The gain in the reduction of the search space gets relatively stable after a given division size; this result is expected since the context provided by locality modules ensures modules with the necessary semantically related entities. The scatter plot in Figure 2d visualise the size of the source modules against the size of the target modules for the FMA-NCI matching subtasks with divisions of size $n \in \{5, 20, 50, 100\}$. For instance, the (blue) circles represent points $\left(|Sig(\mathcal{O}^i_1)|, |Sig(\mathcal{O}^i_2)|\right)$ being $\mathcal{O}^i_1$ and $\mathcal{O}^i_2$ the source and target modules (with i=1,...,5) in the matching subtasks of $\mathcal{D}^5_{\mathcal{MT}}$. It can be noted that, on average, the size of source and target modules decreases as the size of the division increases. For example, the largest task in $\mathcal{D}^{20}_{\mathcal{MT}}$ is represented in point $(6754, 9168)$, while the largest task in $\mathcal{D}^{100}_{\mathcal{MT}}$ is represented in point $(2657, 11842)$.

Computation times. The time to compute the divisions of the matching task is tied to the number of locality modules to extract, which can be computed in polynomial time relative to the size of the input ontology [13]. The creation of LexI does not add an important overhead, while the training of the neural embedding model ranges from $21s$ in AMA-NCI to $224s$ in SNOMED-NCI. Overall, for example, the required time to compute the division with 100 matching subtasks ranges from $23s$ (AMA-NCIA) to approx. $600s$ (SNOMED-NCI).

4.2 Evaluation of OAEI systems

In this section we show that the division of the alignment task enables systems that, given some computational constraints, were unable to complete an OAEI task. We have selected the following five systems from the latest OAEI campaigns, which include novel alignment techniques but failed to scale to very large matching tasks: MAMBA (v.2015) [35], FCA-Map (v.2016) [52], SANOM (v.2017) [37],

Table 3: Evaluation of systems that failed to complete OAEI tasks in the 2015-2018 campaigns. Times reported in seconds (s).

Tool	Task	Matching subtasks	Performance measures			Computation times (s)		
			P	R	F	Min	Max	Total
MAMBA (v.2015)	AMA-NCIA	5	0.870	**0.624**	0.727	73	785	1,981
		10	0.885	0.623	0.731	41	379	1,608
		50	**0.897**	0.623	**0.735**	8	154	**1,377**
FCA-Map (v.2016)	FMA-NCI	20	**0.656**	0.874	**0.749**	39	340	**2,934**
		50	0.625	**0.875**	0.729	19	222	3,213
	FMA-SNOMED	50	**0.599**	0.251	**0.354**	6	280	3,455
		100	0.569	**0.253**	0.350	5	191	**3,028**
	SNOMED-NCI	150	**0.704**	0.629	**0.664**	5	547	**16,822**
		200	0.696	**0.630**	0.661	5	395	16,874
SANOM (v.2017)	FMA-NCI	20	**0.475**	0.720	**0.572**	40	1,467	9,374
		50	0.466	**0.726**	0.568	15	728	**7,069**
	FMA-SNOMED	100	**0.145**	**0.210**	**0.172**	3	1,044	13,073
		150	0.143	0.209	0.170	3	799	**10,814**
POMap++ (v.2018)	FMA-NCI	20	0.697	0.732	0.714	24	850	5,448
		50	**0.701**	**0.748**	**0.724**	11	388	**4,041**
	FMA-SNOMED	50	0.520	0.209	0.298	4	439	5,879
		100	**0.522**	**0.209**	**0.298**	3	327	**4,408**
ALOD2vec (v.2018)	FMA-NCI	20	0.697	0.813	0.751	115	2,141	13,592
		50	**0.698**	**0.813**	**0.751**	48	933	**12,162**
	FMA-SNOMED	100	0.702	0.183	0.29	9	858	12,688
		150	**0.708**	**0.183**	**0.291**	7	581	**10,449**

ALOD2vec (v.2018) [43] and POMap++ (v.2018) [30]. MAMBA failed to complete the anatomy track, while FCA-Map, SANOM, ALOD2vec and POMap++ could not complete the largest tasks in the largebio track. MAMBA and SANOM threw an out-of-memory exception with $25Gb$, whereas FCA-Map, ALOD2vec and POMap++ did not complete the tasks within a 6 hours time-frame. We have used the SEALS infrastructure to conduct the evaluation [3, 4].

Table 3 shows the obtained results in terms of precision, recall, f-measure, and computation times (time for the easiest and the hardest task, and total time for all tasks) over different divisions $\mathcal{D}^n_{\mathcal{MT}}$ computed using our strategy. For example, FCA-Map was run over divisions with 20 and 50 matching subtasks (*i.e.*, $n \in \{20, 50\}$) in the FMA-NCI case. Note that for each matching subtask a system generates a partial alignment $\mathcal{M}^S_i$, the final alignment for the (original) matching task is computed as the union of all partial alignments ($\mathcal{M}^S = \bigcup_{i=1}^{n} \mathcal{M}^S_i$). The results are encouraging and can be summarised as follows:

i) We enabled several systems to produce results even for the largest OAEI test case (*e.g.*, FCA-Map with SNOMED-NCI).

ii) The computation times are also very good falling under the 6 hours time frame, specially given that the (independent) matching subtasks have been run sequentially without parallelization.

iii) The size of the divisions, with the exception of FCA-Map, is beneficial in terms of total computation time.

iv) The increase of number of matching subtasks is positive or neutral for MAMBA, POMap++ and ALOD2vec in terms of f-measure, while it is slightly reduced for FCA-Map and SANOM.

v) Global f-measure results are lower than top OAEI systems; nevertheless, since the above systems could not be evaluated without the divisions, these results are obtained without any fine-tuning of their parameters.

vi) The computation times of the hardest tasks, as n increases, is also reduced. This has a positive impact in the monitoring of alignment systems as the hardest task is completed in a reasonable time.

5 Related work

Partitioning and blocking. Partitioning and modularization techniques have been extensively used within the Semantic Web to improve the efficiency when solving the task at hand (*e.g.*, visualization [45, 1], reuse [29], debugging [47], classification [7]). Partitioning or blocking has also been widely used to reduce the complexity of the ontology alignment task [16]. In the literature there are two major categories of partitioning techniques, namely: *independent* and *dependent*. Independent techniques typically use only the structure of the ontologies and are not concerned about the ontology alignment task when performing the partitioning. Whereas dependent partitioning methods rely on both the structure of the ontology and the ontology alignment task at hand. Although our approach does not compute (non-overlapping) partitions of the ontologies, it can be considered a dependent technique.

Prominent examples of ontology alignment systems including partitioning techniques are Falcon-AO [22], GOMMA [19], COMA++ [5] and TaxoMap [20]. Falcon-AO, GOMMA and COMA++ perform independent partitioning where the clusters of the source and target ontologies are independently extracted. Then pairs of similar clusters (*i.e.*, matching subtasks) are aligned using standard techniques. TaxoMap [20] implements a dependent technique where the partitioning is combined with the matching process. TaxoMap proposes two methods, namely: PAP (partition, anchor, partition) and APP (anchor, partition, partition). The main difference of these methods is the order of extraction of (preliminary) anchors to discover pairs of partitions to be matched (*i.e.*, matching subtasks). SeeCOnt [2] presents a seeding-based clustering technique to discover independent clusters in the input ontologies. Their approach has been evaluated with the Falcon-AO system by replacing its native PBM (Partition-based Block Matching) module [23]. Laadhar et al. [30] have recently integrated within the system POMap++ a hierarchical agglomerative clustering algorithm to divide an ontology into a set of partitions.

The above approaches, although presented interesting ideas, did not provide guarantees about the size and coverage of the discovered partitions or divisions. Furthermore, they have not been successfully evaluated on very large ontologies. On the one hand, as reported by Pereira et al. [42] the results in terms of coverage of the PBM method of Falcon-OA, and the PAP and APP methods of TaxoMap are very low for the OAEI largebio track. On the other hand, as discussed in Section 4, POMap++ fails to scale with the largest largebio tasks.

Note that the recent work in [31] has borrowed from our workshop paper [26] the quality measures presented in Section 2.1. They obtain competitive coverage results for medium size ontologies; however, their approach, as in POMap++, does not scale for large ontologies.

Blocking techniques are also extensively used in Entity Resolution (see [40] for a survey). Although related, the problem of blocking in ontologies is different as the logically related axioms for a seed signature play an important role when computing the blocks.

Our dependent approach, unlike traditional partitioning and blocking methods, computes overlapping self-contained modules (*i.e.*, locality modules [13]). Locality modules guarantee the extraction of all semantically related entities for a given signature. This capability enhances the coverage results and enables the inclusion of the (semantically) relevant information required by an alignment system. It is worth mentioning that the need of self-contained and covering modules, although not thoroughly studied, was also highlighted in a preliminary work by Paulheim [41].

Embedding and clustering. Recently, machine learning techniques such as semantic embedding [12] have been investigated for ontology alignment. They often first learn vector representations of the entities and then predict the alignment [9, 51, 46]. However, most of them focus on alignment of ontology individuals (*i.e.*, ABox) without considering the ontology concepts and axioms at the terminological level (*i.e.*, TBox). Nkisi-Orji et al. [39] predicts the alignment between ontology concepts with Random Forest, but incorporates the embeddings of words alone, without any other semantic components like in our work. Furthermore, these approaches focus on predicting the alignment, while our work aims at boosting an existing alignment system. Our framework could potentially be adopted in systems like in [39] if facing scalability problems for large ontologies.

Another piece of related work is the clustering of semantic components, using the canopy clustering algorithm [33] where objects are grouped into canopies and each object can be a member of multiple canopies. For example, Wu et al. [50] first extracted canopies (i.e., mentions) from a knowledge base, and then grouped the entities accordingly so as to finding out the entities with the same semantics (*i.e.*, canonicalization). As we focus on a totally different task – ontology alignment, the context that can be used, such as the embeddings for the words and ontology entities in LexI, is different from these works, which leads to a different clustering method.

6 Conclusions and future work

We have developed a novel framework to split the ontology alignment task into several matching subtasks based on a semantic inverted index, locality modules, and a neural embedding model. We have performed a comprehensive evaluation which suggests that the obtained divisions are suitable in practice in terms of both coverage and size. The division of the matching task allowed us to obtain results for five systems which failed to complete these tasks in the past. We have focused on systems failing to complete a task, but a suitable adoption and integration of the presented framework within the pipeline of any ontology alignment system has the potential to improve the results in terms of computation times.

Opportunities. Reducing the ontology matching task into smaller and more manageable tasks may also bring opportunities to enhance *(i)* user interaction [32], *(ii)* reasoning and repair [34], *(iii)* benchmarking and monitoring [3, 4], and *(iv)* parallelization. The computed independent matching subtasks can potentially be run in parallel in evaluation platforms like the HOBBIT [38]. The current evaluation was conducted sequentially as *(i)* the SEALS instance only allows running one task at a time, and *(ii)* the evaluated systems were not designed to run several tasks in parallel; for instance, we managed to run MAMBA outside SEALS, but it relies on a MySQL database and raised a concurrent access exception.

Impact on the f-measure. As shown in Section 4.2, the impact of the number of divisions on the f-measure depends on the evaluated systems. In the near future we aim at conducting an extensive evaluation of our framework over OAEI systems able to deal with the largest tasks in order to obtain more insights about the impact on the f-measure. In [25] we reported a preliminary evaluation where YAM-Bio [6] and AML [17] kept similar f-measure values, while LogMap [27] had a reduction in f-measure, as the number of divisions increased.

Number of subdivisions. Currently our strategy requires the size of the number of matching subtasks or divisions as input. The (required) matching subtasks may be known before hand if, for example, the matching tasks are to be run in parallel in a number of available CPUs. For the cases where the resources are limited or where a matching system is known to cope with small ontologies, we plan to design an algorithm to estimate the number of divisions so that the size of the matching subtasks in the computed divisions is appropriate to the system and resource constraints.

Dealing with a limited or large lexicon. The construction of LexI shares a limitation with state-of-the-art systems when the input ontologies are lexically disparate or in different languages. In such cases, LexI can be enriched with general-purpose lexicons (*e.g.*, WordNet), more specialised background knowledge (*e.g.*, UMLS Metathesaurus) or with translated labels using online services (*e.g.*, Google). On the other hand, a large lexicon may also have an important impact in the computation times. Our conducted evaluation shows, however, that we can cope with very large ontologies with a rich lexicon (*e.g.*, NCI Thesaurus).

Notion of context. Locality-based modules are typically much smaller than the whole ontology and they have led to very good results in terms of size and coverage. We plan, however, to study different notions of *context* of an alignment (*e.g.*, the tailored modules proposed in [8]) to further improve the results in terms of size while keeping the same level of coverage.

ACKNOWLEDGEMENTS

This work was supported by the SIRIUS Centre for Scalable Data Access (Norges forskningsråd), the AIDA project (Alan Turing Institute), Samsung Research UK, Siemens AG, and the EPSRC projects AnaLOG, OASIS and UK FIRES. We would also like to thank the anonymous reviewers that helped us improve this work.

REFERENCES

[1] Asan Agibetov, Giuseppe Patanè, and Michela Spagnuolo, 'Grontocrawler: Graph-Based Ontology Exploration', in *STAG*, (2015).

[2] Alsayed Algergawy, Samira Babalou, Mohammad J. Kargar, and S. Hashem Davarpanah, 'SeeCOnt: A New Seeding-Based Clustering Approach for Ontology Matching', in *ADBIS*, (2015).

[3] Alsayed Algergawy et al., 'Results of the Ontology Alignment Evaluation Initiative 2018', in *13th Int'l Workshop on Ontology Matching*, (2018).

[4] Alsayed Algergawy et al., 'Results of the Ontology Alignment Evaluation Initiative 2019', in *Int'l Workshop on Ontology Matching*, (2019).

[5] Alsayed Algergawy, Sabine Massmann, and Erhard Rahm, 'A clustering-based approach for large-scale ontology matching', in *ADBIS*, pp. 415–428, (2011).

[6] Amina Annane, Zohra Bellahsene, Faiçal Azouaou, and Clément Jonquet, 'YAM-BIO: results for OAEI 2017', in *12th Int'l Workshop on Ontology Matching*, (2017).

[7] Ana Armas Romero, Bernardo Cuenca Grau, and Ian Horrocks, 'MORe: Modular Combination of OWL Reasoners for Ontology Classification', in *Int'l Sem. Web Conf.*, (2012).

[8] Ana Armas Romero, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks, 'Module Extraction in Expressive Ontology Languages via Datalog Reasoning', *J. Artif. Intell. Res.*, **55**, (2016).

[9] Michael Azmy, Peng Shi, Jimmy Lin, and Ihab F Ilyas, 'Matching entities across different knowledge graphs with graph embeddings', *CoRR*, **abs/1903.06607**, (2019).

[10] Olivier Bodenreider, Terry F. Hayamizu, Martin Ringwald, Sherri de Coronado, and Songmao Zhang, 'Of mice and men: Aligning mouse and human anatomies', in *AMIA Symposium*, (2005).

[11] Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack, *Information Retrieval - Implementing and Evaluating Search Engines*, MIT Press, 2010.

[12] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang, 'A comprehensive survey of graph embedding: Problems, techniques, and applications', *IEEE Trans. on Know. and Data Eng.*, **30**(9), (2018).

[13] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler, 'Modular reuse of ontologies: Theory and practice', *J. Artif. Intell. Res.*, **31**, (2008).

[14] Gayo Diallo, 'An effective method of large scale ontology matching', *J. Biomedical Semantics*, **5**, 44, (2014).

[15] Jérôme Euzenat and Pavel Shvaiko, *Ontology Matching, Second Edition*, Springer, 2013.

[16] Daniel Faria, Catia Pesquita, Isabela Mott, Catarina Martins, Francisco M. Couto, and Isabel F. Cruz, 'Tackling the challenges of matching biomedical ontologies', *J. Biomedical Semantics*, **9**(1), (2018).

[17] Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F. Cruz, and Francisco M. Couto, 'The AgreementMakerLight Ontology Matching System', in *OTM-ODBASE Conference*, (2013).

[18] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter F. Patel-Schneider, and Ulrike Sattler, 'OWL 2: The next step for OWL', *J. Web Semantics*, **6**(4), (2008).

[19] Anika Groß, Michael Hartung, Toralf Kirsten, and Erhard Rahm, 'On matching large life science ontologies in parallel', in *Data Integration in the Life Sciences (DILS)*, (2010).

[20] Fayçal Hamdi, Brigitte Safar, Chantal Reynaud, and Haïfa Zargayouna, 'Alignment-based partitioning of large-scale ontologies', in *Advances in Knowledge Discovery and Management*, (2009).

[21] Ian Harrow, Ernesto Jiménez-Ruiz, et al., 'Matching disease and phenotype ontologies in the ontology alignment evaluation initiative', *J. Biomedical Semantics*, **8**(1), (2017).

[22] Wei Hu, Yuzhong Qu, and Gong Cheng, 'Matching large ontologies: A divide-and-conquer approach', *Data Knowl. Eng.*, **67**, 140–160, (2008).

[23] Wei Hu, Yuanyuan Zhao, and Yuzhong Qu, 'Partition-Based Block Matching of Large Class Hierarchies', in *Asian Sem. Web Conf.*, (2006).

[24] Ernesto Jiménez-Ruiz, Asan Agibetov, Jiaoyan Chen, Matthias Samwald, and Valerie Cross. Dividing the Ontology Alignment Task [Data set], 2019. Zenodo. https://doi.org/10.5281/zenodo.3547888.

[25] Ernesto Jiménez-Ruiz, Asan Agibetov, Matthias Samwald, and Valerie Cross, 'Breaking-down the ontology alignment task with a lexical index and neural embeddings', *CoRR*, **abs/1805.12402**, (2018).

[26] Ernesto Jiménez-Ruiz, Asan Agibetov, Matthias Samwald, and Valerie Cross, 'We divide, you conquer: from large-scale ontology alignment to manageable subtasks with a lexical index and neural embeddings', in *13th Int'l Workshop on Ontology Matching*, (2018).

[27] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau, 'LogMap: Logic-Based and Scalable Ontology Matching', in *Int'l Sem. Web Conf.*, (2011).

[28] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga Llavori, 'Logic-based assessment of the compatibility of UMLS ontology sources', *J. Biomedical Semantics*, **2**, (2011).

[29] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ulrike Sattler, Thomas Schneider, and Rafael Berlanga, 'Safe and Economic Re-Use of Ontologies: A Logic-Based Methodology and Tool Support', in *European Sem. Web Conf.*, (2008).

[30] Amir Laadhar, Faïza Ghozzi, Imen Megdiche, Franck Ravat, Olivier Teste, and Faïez Gargouri, 'OAEI 2018 results of POMap++', in *13th Int'l Workshop on Ontology Matching*, (2018).

[31] Amir Laadhar, Faïza Ghozzi, Imen Megdiche, Franck Ravat, Olivier Teste, and Faïez Gargouri, 'Partitioning and local matching learning of large biomedical ontologies', in *34th ACM/SIGAPP Symposium on Applied Computing SAC*, (2019).

[32] Huanyu Li, Zlatan Dragisic, Daniel Faria, Valentina Ivanova, Ernesto Jiménez-Ruiz, Patrick Lambrix, and Catia Pesquita, 'User validation in ontology alignment: functional assessment and impact', *The Knowledge Engineering Review*, **34**, (2019).

[33] Andrew McCallum, Kamal Nigam, and Lyle H Ungar, 'Efficient clustering of high-dimensional data sets with application to reference matching', in *6th ACM SIGKDD*, (2000).

[34] Christian Meilicke, *Alignment incoherence in ontology matching*, Ph.D. dissertation, University of Mannheim, 2011.

[35] Christian Meilicke, 'MAMBA - results for the OAEI 2015', in *10th Int'l Workshop on Ontology Matching*, (2015).

[36] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, 'Distributed representations of words and phrases and their compositionality', *arXiv*, (oct 2013).

[37] Majid Mohammadi, Amir Ahooye Atashin, Wout Hofman, and Yao-Hua Tan, 'SANOM results for OAEI 2017', in *12th Int'l Workshop on Ontology Matching*, (2017).

[38] Axel-Cyrille Ngonga Ngomo, Alejandra García-Rojas, and Irini Fundulaki, 'HOBBIT: holistic benchmarking of big linked data', *ERCIM News*, **2016**(105), (2016).

[39] Ikechukwu Nkisi-Orji, Nirmalie Wiratunga, Stewart Massie, Kit-Ying Hui, and Rachel Heaven, 'Ontology alignment based on word embedding and random forest classification', in *ECML/PKDD*, (2018).

[40] George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, and Themis Palpanas, 'A Survey of Blocking and Filtering Techniques for Entity Resolution', *CoRR*, **abs/1905.06167**, (2019).

[41] Heiko Paulheim, 'On Applying Matching Tools to Large-scale Ontologies', in *3rd Int'l Workshop on Ontology Matching*, (2008).

[42] Sunny Pereira, Valerie Cross, and Ernesto Jiménez-Ruiz, 'On partitioning for ontology alignment', in *Int'l Sem. Web Conf. (Poster)*, (2017).

[43] Jan Portisch and Heiko Paulheim, 'ALOD2Vec matcher', in *13th Int'l Workshop on Ontology Matching*, (2018).

[44] Pavel Shvaiko and Jérôme Euzenat, 'Ontology matching: State of the art and future challenges', *IEEE Trans. Knowl. Data Eng.*, **25**(1), (2013).

[45] Heiner Stuckenschmidt and Anne Schlicht, 'Structure-based partitioning of large ontologies', in *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, Springer, (2009).

[46] Zequn Sun, Jiacheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu, 'TransEdge: Translating Relation-Contextualized Embeddings for Knowledge Graphs', in *Int'l Sem. Web Conf. (ISWC)*, (2019).

[47] Boontawee Suntisrivaraporn, Guilin Qi, Qiu Ji, and Peter Haase, 'A modularization-based approach to finding all justifications for OWL DL entailments', in *Asian Sem. Web Conf.*, (2008).

[48] Jason Weston, Sumit Chopra, and Keith Adams, '#tagspace: Semantic embeddings from hashtags', in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (2014).

[49] Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston, 'StarSpace: Embed All The Things!', *arXiv*, (2017).

[50] Tien-Hsuan Wu, Zhiyong Wu, Ben Kao, and Pengcheng Yin, 'Towards practical open knowledge base canonicalization', in *27th ACM Int'l Conf. on Inf. and Knowledge Management*, (2018).

[51] Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu, 'Multi-view knowledge graph embedding for entity alignment', in *28th Int'l Joint Conf. on Art. Intell. (IJCAI)*, (2019).

[52] Mengyi Zhao and Songmao Zhang, 'FCA-Map results for OAEI 2016', in *11th Int'l Workshop on Ontology Matching*, (2016).