From Monolingual to Multilingual Ontologies: The Role of Cross-lingual Ontology Enrichment

Shimaa Ibrahim[1,2] ✉, Said Fathalla[1,3], Hamed Shariat Yazdi[1], Jens Lehmann[1,4], and Hajira Jabeen[1]

[1] Smart Data Analytics (SDA), University of Bonn, Bonn Germany
{ibrahim,fathalla,shariat,jens.lehmann,jabeen}@cs.uni-bonn.de
[2] Institute of Graduate Studies and Research, University of Alexandria, Alexandria, Egypt
[3] Faculty of Science, University of Alexandria, Alexandria, Egypt
[4] Enterprise Information Systems Department, Fraunhofer IAIS, Sankt Augustin, Germany

Abstract. While the multilingual data on the Semantic Web grows rapidly, the building of multilingual ontologies from monolingual ones is still cumbersome and hampered due to the lack of techniques for cross-lingual ontology enrichment. Cross-lingual ontology enrichment greatly facilitates the semantic interoperability between different ontologies in different natural languages. Achieving such enrichment by human labor is a time-consuming and error-prone task. Thus, in this paper, we propose a fully automated ontology enrichment approach using cross-lingual matching (OECM), which builds a multilingual ontology by enriching a monolingual ontology from another one in a different natural language. OECM selects the best translation among all available translations of ontology concepts based on their semantic similarity with the target ontology concepts. We present a use case of our approach for enriching English Scholarly Communication Ontologies using German and Arabic ontologies from the MultiFarm benchmark. We have compared our results with the results from the Ontology Alignment Evaluation Initiative (OAEI 2018). Our approach has higher precision and recall in comparison to five state-of-the-art approaches. Additionally, we recommend some linguistic corrections in the Arabic ontologies in Multifarm which have enhanced our cross-lingual matching results.

Keywords: Cross-lingual ontology enrichment · Cross-lingual matching · multilingual ontology · Ontology engineering · Knowledge management.

1 Introduction

The wide proliferation of multilingual data on the Semantic Web results in many ontologies scattered across the web in various natural languages. According to the Linked Open Vocabularies (LOV)[5], the majority of the ontologies in the

[5] https://lov.linkeddata.es/dataset/lov/vocabs

Semantic Web are in English, however, ontologies in other Indo-European languages also exist. For instance, out of a total 681 vocabularies found in LOV, 500 are in English, 54 in French, 39 in Spanish, and 33 in German. Few ontologies exist in non-Indo-European languages, such as 13 in Japanese and seven in Arabic. Monolingual ontologies with labels or local names presented in a certain language are not easily understandable to speakers of other languages. Therefore, in order to enhance semantic interoperability between monolingual ontologies, approaches for building multilingual ontologies from the existing monolingual ones should be developed [26]. Multilingual ontologies can be built by applying *cross-lingual ontology enrichment* techniques, which expand the target ontology with additional concepts and semantic relations extracted from external resources in other natural languages [23]. For example, suppose we have two ontologies; Scientific Events Ontology in English (SEO_{en}) and Conference in German ($Conference_{de}$). Both SEO_{en} and $Conference_{de}$ have complementary information, i.e. SEO_{en} has some information which does not exist in $Conference_{de}$ and vice versa. Let us consider a scenario where a user wants to get information from both SEO_{en} and $Conference_{de}$ to be used in an ontology-based application. This may not be possible without a cross-lingual ontology enrichment solution, which enrich the former by the complementary information in the latter. Manual ontology enrichment is a resource demanding and time-consuming task. Therefore, fully automated *cross-lingual* ontology enrichment approaches are highly desired [23]. Most of the existing work in ontology enrichment focus on enriching English ontologies from English sources only (monolingual enrichment) [23]. To the best of our knowledge, only our previous work [1,14] has addressed the cross-lingual ontology enrichment problem by proposing a semi-automated approach to enrich ontologies from multilingual text or from other ontologies in different natural languages.

In this paper we address the following research question; *how can we automatically build multilingual ontologies from monolingual ones?* We propose a fully automated ontology enrichment approach in order to create multilingual ontologies from monolingual ones using cross-lingual matching. We extend our previous work [14] by: 1) using the semantic similarity to select the best translation of class labels, 2) enriching the target ontology by adding new classes in addition to all their related subclasses in the hierarchy, 3) using ontologies in non-Indo-European languages (e.g., Arabic), as the source of information, 4) building multilingual ontologies, and 5) developing a fully automated approach. OECM comprises six phases: 1) *translation*: translate class labels of the source ontology, 2) *pre-processing*: process class labels of the target and the translated source ontologies, 3) *terminological matching*: identify potential matches between class labels of the source and the target ontologies, 4) *triple retrieval*: retrieve the new information to be added to the target ontology, 5) *enrichment*: enrich the target ontology with new information extracted from the source ontology, and 6) *validation*: validate the enriched ontology. A noticeable feature of OECM is that we consider multiple translations for a class label. In addition, the use of semantic similarity has significantly improved the quality of the matching process.

We present a use case for enriching the Scientific Events Ontology (SEO) [9], a scholarly communication ontology for describing scientific events, from German and Arabic ontologies. We compare OECM to five state-of-the-art approaches for cross-lingual ontology matching task. OECM outperformed these approaches in terms of precision, recall, and F-measure. Furthermore, we evaluate the enriched ontology by comparing it against a Gold standard created by ontology experts. The implementation of OECM and the datasets used in the use case are publicly available[6].

The remainder of this paper is structured as follows: we present an overview of related work in section 2. Overview of the proposed approach is described in section 3. In order to illustrate possible applications of OECM, a use case is presented in section 4. Experiments and evaluation results are presented in section 5. Finally, we conclude with an outline of the future work in section 6.

2 Related Work

A recent review of the literature on multilingual Web of Data found that the potential of the Semantic Web for being multilingual can be accomplished by techniques to build multilingual ontologies from monolingual ones [12]. Multilingual enrichment approaches are used to build multilingual ontologies from different resources in different natural languages [6,24,5]. Espinoza et al. [6] has proposed an approach to generate multilingual ontologies by enriching the existing monolingual ontologies with multilingual information in order to translate these ontologies to a particular language and culture (ontology localization). In fact, ontology enrichment depends on matching the target ontology with external resources, in order to provide the target ontology with additional information extracted from the external resources.

All the literature have focused on the cross-lingual ontology matching techniques which are used for matching different natural languages of linguistic information in ontologies [12,26]. Meilicke et al. [20] created a benchmark dataset (MultiFarm) that results from the manual translations of a set of ontologies from the conference domain into eight natural languages. This dataset is widely used to evaluate the cross-lingual matching approaches [28,7,15,16]. Manual translation of ontologies can be infeasible when dealing with large and complex ontologies. Trojahn et al. [27] proposed a generic approach which relies on translating concepts of source ontologies using machine translation techniques into the language of the target ontology. In the translation step, they depend on getting one translation for each concept (one-to-one translation), then they apply monolingual matching approaches to match concepts between the source ontologies and the translated ones. Fu et al. [10,11] proposed an approach to match English and Chinese ontologies by considering the semantics of the target ontology, the mapping intent, the operating domain, the time and resource constraints and user feedback. Hertling and Paulheim [13] proposed an approach which utilizes

[6] `https://github.com/shmkhaled/OECM`

Wikipedias inter-language links for finding corresponding ontology elements. Lin and Krizhanovsky [18] proposed an approach which use Wiktionary[7] as a source of background knowledge to match English and French ontologies. Tigrine et al. [25] presented an approach, which relies on the multilingual semantic network BabelNet[8] as a source of background knowledge, to match several ontologies in different natural languages. In the context of OAEI 2018 campaign[9] for evaluating ontology matching technologies, AML [7], KEPLER [16], LogMap [15] and XMap [28] provide high-quality alignments. These systems use terminological and structural alignments in addition to using external lexicon, such as WordNet[10] and UMLS-lexicon[11] in order to get the set of synonyms for the ontology elements. In order to deal with multilingualism, AML and KEPLER rely on getting (one-to-one translation) using machine translation technologies, such as Microsoft translator, before starting the matching process. LogMap and XMap do not provide any information about the utilized translation methodology. Moreover, LogMap is an iterative process, that starts from initial mappings (almost exact lexical correspondences) to discover new mappings. It is mentioned in [15] that the main weakness of LogMap is that it can not find matching between ontologies which do not provide enough lexical information as it depends mainly on the initial mappings. A good literature of the state-of-the-art approaches in cross-lingual ontology matching is provided in [26].

Most of the literature have focused on enriching monolingual ontologies with multilingual information in order to translate or localize these ontologies. In addition, in the cross-lingual ontology matching task, there is a lack of exact one-to-one translation between terms across different natural languages which negatively affects the matching results. We address this limitations in our proposed approach by building multilingual ontologies, where a class label is presented by several natural languages, from monolingual ones. Such approach support the ontology matching process with multiple translations for a class label in order to enhance the matching results.

3 The Proposed Approach

Goal: Given two ontologies S and T, in two different natural languages L_s and L_t respectively, as RDF triples $\langle s, p, o\rangle \in \mathcal{C} \times \mathcal{R} \times (\mathcal{C} \cup \mathcal{L})$ where $\mathcal{C}$ is the set of ontology domain entities (i.e. classes), $\mathcal{R}$ is the set of relations, and $\mathcal{L}$ is the set of literals. We aim at finding the complementary information $\mathcal{T}_e = S - (S \cap T)$ from S in order to enrich T.

The proposed approach comprises six phases (Figure 1): translation, preprocessing, terminological matching, triple retrieval, enrichment, and validation. The input is the two ontologies in two different natural languages, i.e. the target

[7] https://www.wiktionary.org/
[8] https://babelnet.org/
[9] http://oaei.ontologymatching.org/2018/results/multifarm/index.html
[10] https://wordnet.princeton.edu/
[11] https://www.nlm.nih.gov/research/umls/

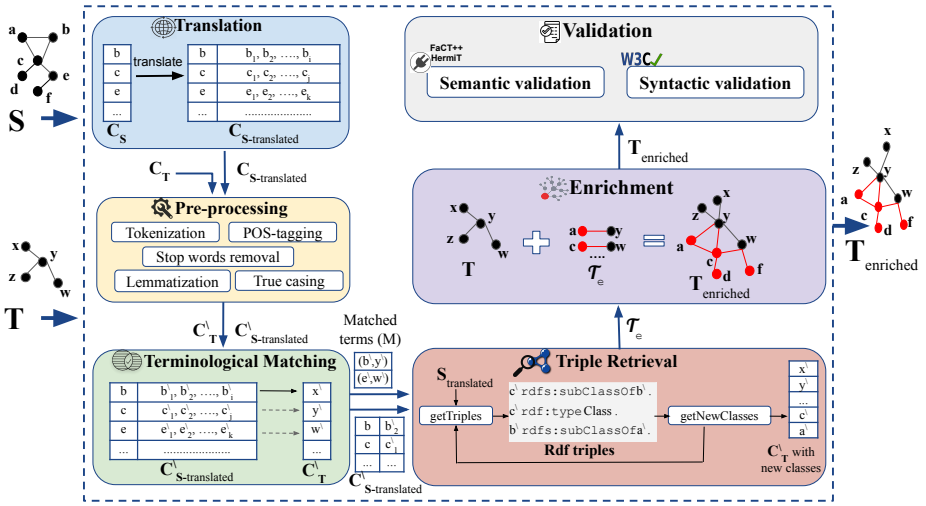

Fig. 1. The workflow of OECM.

ontology T and the source ontology S. The output is the multilingual enriched ontology $T_{enriched}$ in two different natural languages L_1 and L_2. In the following subsections, we describe each of these phases in details.

3.1 Translation

Let $\mathcal{C}_S$ and $\mathcal{C}_T$ be the set of classes in S and T respectively. Each class is represented by a label or a local name. The aim of this phase is to translate each class in $\mathcal{C}_S$ to the language of T (i.e. L_t). Google Translator[12] is used to translate classes of source ontologies. All available translations are considered for each class. Therefore, the output of the translation is $\mathcal{C}_{S-translated}$ which has each class, in S, associated with a list of all available translations. For example, the class `Thema` in German has a list of English translations (Subject and Topic), and the class label "مراجعة" in Arabic has a list of English translations such as "Review, Revision, Check". The best translation will be selected in the terminological matching phase (subsection 3.3).

3.2 Pre-processing

The aim of this phase is to process classes of $\mathcal{C}_T$ and lists of translations in $\mathcal{C}_{S-translated}$ by employing a variety of natural language processing (NLP) techniques, such as tokenization, POS-tagging (part-of-speech tagging), and lemmatization, to make it ready for the next phases. In order to enhance the similarity

[12] `https://translate.google.com/`

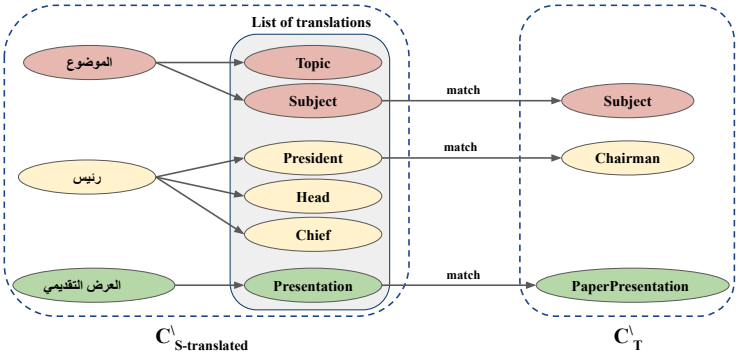

Fig. 2. Illustration of a terminological matching between list of translations, in English, for every concept in $\mathcal{C}'_{S-translated}$, in Arabic, and $\mathcal{C}'_T$ in English

results between $\mathcal{C}_T$ and $\mathcal{C}_{S-translated}$, stop words are removed and normalization methods and regular expressions are used to remove punctuation, symbols, additional white spaces, and to normalize the structure of strings. Furthermore, our pre-processing is capable of recognizing classes such as camel cases "ReviewArticle" and adds a space between lower-case and upper-case letters "Review Article" (i.e. true casing technique). The output of this phase is $\mathcal{C}'_T$, which has pre-processed translations of classes in T, and $\mathcal{C}'_{S-translated}$, which has pre-processed translations for each class in S.

3.3 Terminological Matching

The aim of this phase is to identify potential matches between class labels of S and T. We perform a pairwise lexical and/or semantic similarity between the list of translations of each class in $\mathcal{C}'_{S-translated}$ and $\mathcal{C}'_T$ to select the best translation for each class in S that matches the corresponding class in T (see algorithm 1). Jaccard similarity [22] is used to filter the identical concepts instead of using semantic similarity from the beginning because there is no need for extra computations to compute semantic similarity between two identical classes. The reason behind choosing the Jaccard similarity is that according to the experiments conducted for the ontology alignment task for the MultiFarm benchmark in [2], Jaccard similarity has achieved the best score in terms of precision. For non-identical concepts, we compute the semantic similarity using the path length measure, based on WordNet[10], which returns the shortest path between two words in WordNet hierarchy [3]. If two words are semantically equivalent, i.e., belonging to the same WordNet synset, the path distance is 1.00. We use a specific threshold θ in order to get the set of matched terms (matched classes) M. We obtained the best value of $\theta = 0.9$ which has the best matching results after running the experiments for ten times. If no match is found, we consider this class as a new class that can be added to T and we consider its list of translations as synonyms for that class. Generally, class labels have more

Algorithm 1: Terminological Matching

```
   Data: C'_{S-translated}, C'_T, θ similarity threshold
   Result: M matched terms, C'_{S-translated}
 1 foreach c_s ∈ C'_{S-translated}, t ∈ listOfTranslations, c_t ∈ C'_T do
 2 |   similarityScore ← getSimilarity(t,c_t)
 3 |   if similarityScore ≥ θ then
 4 |   |   M ::= (t, c_t)
 5 |   |   C'_{S-translated} = update(C'_{S-translated},M)
 6 Function getSimilarity(sentence1, sentence2):double
 7 |   similarity ← getJaccardSimilarity(sentence1, sentence2)
 8 |   if similarity ≠ 1 then
 9 |   |   similarity ← (sentenceSimilarity(sentence1,sentence2)
10 |   |   + sentenceSimilarity(sentence2,sentence1))/2
11 |   return similarity
12 Function sentenceSimilarity(sentence1, sentence2):double
13 |   simScore ← 0.0
14 |   count ← 0.0
15 |   foreach w_i ∈ sentence1.split(" ") do
16 |   |   foreach w_j ∈ sentence2.split(" ") do
17 |   |   |   pathSim ::= getPathSimilarity(w_i,w_j)
18 |   |   simScore+ = pathSim.max
19 |   |   count+ = 1
20 |   simScore ← simScore/count
21 |   return simScoure
```

than one word, for example "InvitedSpeaker", therefore, the semantic similarity between sentences presented in [21] is adapted as described in algorithm 1 - line 9. Given two sentences *sentence*1 and *sentence*2, the semantic similarity of each sentence with respect to the other is defined by: for each word $w_i \in sentence1$, the word w_j in *sentence*2 that has the highest path similarity with w_i is determined. The word similarities are then summed up and normalized with the number of similar words between the two sentences. Next, the same procedure is applied to start with words in *sentence*2 to identify the semantic similarity of *sentence*2 with respect to *sentence*1. Finally, the resulting similarity scores are combined using a simple average. Based on the similarity results, the best translation is selected and $\mathcal{C}'_{S-translated}$ is updated. For example, in Figure 2, the class "رئيس" in Arabic, has a list of English translations such as "President, Head, Chief". After computing the similarity between $\mathcal{C}'_{S-translated}$ and $\mathcal{C}'_T$, "President" has the highest similarityScore of 1.00 with the class "Chairman", in $\mathcal{C}'_T$, because they are semantically equivalent. Therefore, "President" is selected to be the best translation for "رئيس". The output of this phase is the list of matched terms M between $\mathcal{C}'_T$ and the updated $\mathcal{C}'_{S-translated}$.

Algorithm 2: Triple Retrieval

Data: S, $\mathcal{C}'_{S-translated}$, $\mathcal{C}'_T$, M
Result: T_e triples to be enriched

$S_{translated} \leftarrow$ translateOntologyClasses(S, $\mathcal{C}'_{S-translated}$)
$newClasses \leftarrow M$
while *!newClasses.isEmpty()* **do**
  $tempTriples \leftarrow$ getTriplesForNewClasses($S_{translated}$, $newClasses$)
  $newClasses \leftarrow$ getClasses($tempTriples$).subtract($newClasses$)
  $newTriples \leftarrow newTriples$.union($tempTriples$)
$otherLangTriples \leftarrow$ getOtherLangTriples($newTriples$, $\mathcal{C}'_{S-translated}$)
$T_e \leftarrow newTriples$.union($foreignLanguageTriples$)

3.4 Triple Retrieval

The aim of this phase is to identify which and where the new information can be added to T. Each class in S is replaced by its best translation found in $\mathcal{C}'_{S-translated}$ from the previous phase in order to get a translated ontology $S_{translated}$ (see algorithm 2). We design an iterative process in order to obtain $\mathcal{T}_e$, which is represented by $\langle s, p, o \rangle$, that has all possible multilingual information from S to be added to T. We initiate the iterative process with all matched terms ($newClasses = M$) in order to get all related classes, if exist. The iterative process has three steps: 1) for each class $c \in newClasses$, all triples $tempTriples$ are retrieved from $S_{translated}$ where c is a subject or an object, 2) a new list of new classes is obtained from $tempTriples$, 3) $tempTriples$ is added to $newTriples$ which will be added to T. These three steps are repeated until no new classes can be found ($newClasses$.isEmpty() $= true$). Next, we retrieve all available information from the other language for each class in $newTriples$ such as ⟨ president, `rdfs:label`, "رئيس"@ar⟩. The output of this phase is $\mathcal{T}_e$ which contains all multilingual triples (i.e., in L_s and L_t languages) to be added to T.

3.5 Enrichment

The aim of this phase is to enrich T using triples in $\mathcal{T}_e$. By using OECM, the target ontology can be enriched from several ontologies in different natural languages sequentially, i.e. one-to-many enrichment. In this case, $T_{enriched}$ can have more than two natural languages. For instance, English T can be enriched from a German ontology, then the enriched ontology can be enriched again form a different Arabic ontology, i.e. the final result for $T_{enriched}$ is presented in English, German, and Arabic. With the completion of this phase, we have successfully enriched T and create a multilingual ontology from monolingual ones.

3.6 Validation

The aim of this phase is to validate the enriched ontology, which is a crucial step to detect inconsistencies and syntax errors, which might be produced during

```
### https://w3id.org/seo#Publisher
seo:Publisher rdf:type owl:Class ;
        rdfs:subClassOf <http://xmlns.com/foaf/0.1/Organization> ;
        rdfs:comment "The publisher of the event proceedings."@en ;
        rdfs:label  "Publisher"@en .
                    "Herausgeber"@de .

### http://conference_de#CommitteeMember
conference_de:CommitteeMember rdf:type owl:Class ;
        rdfs:subClassOf <http://xmlns.com/foaf/0.1/Person> ;
        rdfs:label  "committee member"@en .
                    "Angehörige des Ausschusses"@de .

### https://w3id.org/seo#Chair
seo:Chair rdf:type owl:Class;
        rdfs:subClassOf conference_de:CommitteeMember ;
        rdfs:label  "Chair"@en .
                    "Vorsitzender"@de .
```

Fig. 3. Small fragment from SEO_{en-de} ontology after the enrichment. The newly added information is marked in bold.

the enrichment process [8]. There are two types of validations: syntactic and semantic validation. In the syntactic validation, we validate $T_{enriched}$ to conform with the W3C RDF standards using the online RDF validation service[13] which detects syntax errors, such as missing tags. For semantic validation, we use two reasoners, FaCT++ and HermiT, for detecting inconsistencies in $T_{enriched}$ [8].

4 Use Case: Enriching the Scientific Events Ontology

In this use case, we use an example scenario to enrich the SEO_{en}[14] ontology (with 49 classes), in English, using the MultiFarm dataset (see section 5). We use the Conference ontology (60 classes) and the ConfOf ontology (38 classes), in German and Arabic respectively, as source ontologies. This use case aims to show the whole process starting from submitting the source and target ontologies until producing the enriched multilingual ontology. Here, the source ontology is the German ontology $Conference_{de}$ and the target ontology is the English ontology SEO_{en}. The output is the enriched ontology SEO_{en-de}, which becomes a multilingual ontology in English and German. Table 1 demonstrates the enrichment process for SEO_{en} from $Conference_{de}$ and shows the output sample of each phase starting from the translation phase to the produced set of triples which are used to enrich SEO_{en}. In the terminological matching task, the relevant matching results (with similarity scores in bold) are identified with $\theta \geq 0.9$. The iterative process, in the triple retrieval phase, is initiated with the identified matched terms, for example, `person` class. At the first iteration, six triples

[13] `https://www.w3.org/RDF/Validator/`
[14] `https://w3id.org/seo`

Table 1. Use case: the sample output of each phase, from translation to triple retrieval.

Phase	Output
Translation	(`Thema`)$_{de}$ → (subject, topic)$_{en}$ (`Gutachter`)$_{de}$ → (reviewer, expert)$_{en}$ (`Herausgeber`)$_{de}$ → (publisher, editor)$_{en}$ (`Fortschritte der Konferenz`)$_{de}$ → (Progress of the conference)$_{en}$
Pre-processing	`SizeOrDuration` → size duration `WorkshopProposals` → workshop proposal `InvitedSpeaker` → invite speaker `In-useTrack` → use track
Terminological matching score results	(`invited speaker`, `keynote speaker`, 0.57) (`person`, `person`, **1.00**) (`tutorial`, `tutorial proposals`, 0.78) (`prize`, `award`, **1.00**) (`conference document`, `license document`, 0.61) (`publisher`, `publisher`, **1.00**) (`conference series`, `event series`, 0.79) (`conference series`, `symposium series`, 0.75) (`proceedings`, `proceedings`, **1.00**) (`poster`, `posters track`, 0.78)
Triple Retrieval (Iterative process)	1^{st} Iteration: ⟨**`conference contributor`**, `rdfs:subClassOf`, `person`⟩ ⟨**`committee member`**, `rdfs:subClassOf`, `person`⟩ 2^{nd} Iteration: ⟨`committee member`, `rdf:type`, `Class`⟩ ⟨**`chairman`**, `rdfs:subClassOf`, `committee member`⟩ ⟨`conference contributor`, `rdf:type`, `Class`⟩ ⟨**`invited speaker`**, `rdfs:subClassOf`, `conference contributor`⟩ ⟨**`regular author`**, `rdfs:subClassOf`, `conference contributor`⟩
Triple Retrieval ($\mathcal{T}_e$)	⟨`committee member`, `rdf:type`, `Class`⟩ ⟨`committee member`, `rdfs:label`, `"committee member"@en`⟩ ⟨`committee member`,`rdfs:label`,`"Angehörige des Ausschusses"@de`⟩ ⟨`chairman`, `rdfs:subClassOf`, `committee member`⟩

(not all results are exist in the table because of the limited space) are produced such as ⟨**`conference contributor`**, `rdfs:subClassOf`, `person`⟩, where the matched term `person` is located at the object position. New classes are determined from the produced triples such as `conference contributor` and `committee member` (in bold). At the second iteration, all triples that have these new classes, as subject or object, are retrieved, for example; for the `committee member` class, the triples ⟨`committee member`, `rdf:type`, `Class`⟩ and ⟨**`chairman`**, `rdfs:subClassOf`, `committee member`⟩ are retrieved. This process is repeated again and new classes are identified from the produced triples such as `chairman`. The iterative process ended at the fifth iteration where three triples are produced without any new classes. The output of this phase is $\mathcal{T}_e$ which has 40 new triples (with 20 new classes and their German labels), to be added to SEO$_{en}$

and produce SEO_{en-de}. Figure 3 shows a small fragment of the enriched ontology SEO_{en-de}, in Turtle, after completing the enrichment process. The resulting multilingual ontology contains a newly added class `CommitteeMember` with its English and German labels, a new relation `rdfs:subClassOf` between the two classes `CommitteeMember` and `Chair`, and new German labels such as `Herausgeber` and `Vorsitzender` for classes `Publisher` and `Chair` respectively. Similarly, SEO_{en-de} is enriched from the Arabic ontology $ConfOf_{ar}$, where all classes with English labels in SEO_{en-de} are matched with class labels in $ConfOf_{ar}$. The produced $SEO_{en-de-ar}$ has 113 new triples with 37 new classes with their Arabic labels. Final output results can be found at the OECM GitHub repository[6].

5 Evaluation

The aim of this evaluation is to measure the quality of the cross-lingual matching process in addition to the enrichment process. We use ontologies in MultiFarm benchmark[15], a benchmark designed for evaluating cross-lingual ontology matching systems. MultiFarm consists of seven ontologies (*Cmt*, *Conference*, *ConfOf*, *Edas*, *Ekaw*, *Iasted*, *Sigkdd*) originally coming from the Conference benchmark of OAEI, their translation into nine languages (Chinese, Czech, Dutch, French, German, Portuguese, Russian, Spanish and Arabic), and the corresponding cross-lingual alignments between them.

Experimental Setup. All phases of OECM have been implemented using Scala and Apache Spark[16]. SANSA-RDF library[17] [17] with Apache Jena framework[18] are used to parse and manipulate the input ontologies (as RDF triples). In order to process the class labels, the Stanford CoreNLP[19] [19] is used. All experiments are carried out on Ubuntu 16.04 LTS operating system with an Intel Corei7-4600U CPU @ 2.10GHz x 4 CPU and 10 GB of memory. In our experiments, we consider English ontologies as target ontologies to be enriched from German and Arabic ontologies.

Our evaluation has three tasks: 1) evaluating the effectiveness of the cross-lingual matching process in OECM compared to the reference alignment provided in the MultiFarm benchmark, 2) comparing OECM matching results with four state-of-the-art approaches, in addition to our previous work (OECM 1.0) [14], and 3) evaluating the quality of the enrichment process.

Effectiveness of OECM. In this experiment, we use the English version of *Cmt* ontology as the source ontology, and German and Arabic versions of *Conference*, *ConfOf*, and *Sigkdd* ontologies as target ontologies. We match class labels in *Cmt* ontology with class labels of German and Arabic versions of *Conference*, *ConfOf*, and *Sigkdd* ontologies separately. The resulting alignments are

[15] `https://www.irit.fr/recherches/MELODI/multifarm/`
[16] `https://spark.apache.org/`
[17] `https://github.com/SANSA-Stack/SANSA-RDF`
[18] `https://jena.apache.org/`
[19] `https://stanfordnlp.github.io/CoreNLP/`

Table 2. Precision, recall and F-measures for the cross-lingual matching

Ontology pairs	German× English			Arabic × English					
	Precision	Recall	F-measure	Precision		Recall		F-measure	
				Before	After	Before	After	Before	After
Conference × Cmt	1.00	0.38	0.56	1.00	1.00	0.33	0.42	0.50	0.59
ConfOf × Cmt	1.00	0.70	0.82	1.00	1.00	0.30	0.60	0.46	0.75
Sigkdd × Cmt	1.00	0.90	0.95	1.00	1.00	0.40	0.80	0.57	0.89

compared with the reference alignments, as a gold standard, provided in the benchmark for each pair of ontologies. Table 2 shows the precision, recall and F-measure of the matching process for each pair of ontologies. OECM achieves the highest precision of 1.00 for all pair of ontologies. Meanwhile, OECM achieves the highest recall and F-measure of 0.90 and 0.95 respectively for matching the German *Sigkdd* with the English *Cmt*. As two authors of this work are native speakers of Arabic, we found some linguistic mistakes in the Arabic ontologies which negatively affect the translation and the matching results. Therefore, we correct these mistakes and make it available at the OECM GitHub repository[6]. Matching results *before* and *after* the corrections are presented in the table, where such corrections have greatly improved the matching results in terms of recall and F-measure. For instance, in matching the Arabic *Sigkdd* with the English *Cmt*, recall and F-measure are enhanced by 40% and 32% respectively.

Comparison with the state-of-the-art. We identified four of the related approaches (AML, KEPLER, LogMap, and XMap) to be included in our evaluation in addition to OECM 1.0. The other related work, neither publish their code, nor their evaluation datasets [25,11,10]. In order to compare our results with the state-of-the-art, we use German (Conference_{de}) and Arabic (Conference_{ar}) versions of the *Conference* ontology as the source ontologies, and Ekaw_{en} and Edas_{en} ontologies as the target English ontologies. We choose Ekaw_{en} and Edas_{en} ontologies in this evaluation because they are used in the state-of-the-art systems for evaluation, as mentioned in the results of OAEI 2018[9]. We generate the gold standard alignments between each pair of ontologies using the Alignment API 4.9[20], as used by the state-of-the-art systems, in order to compute precision, recall, and F-measures. Table 3 shows the comparison between our results against four state-of-the-art approaches and OECM 1.0 (results for matching English and German ontologies only). In addition, we add the updated Arabic ontology (Conference'_{ar}) with our linguistic correction in the matching process in order to show the effectiveness of such corrections. The current version of OECM (OECM 1.1) outperforms all other systems in precision, recall and F-measure. For instance, when matching Conference_{de} × Ekaw_{en}, OECM 1.1 outperforms LogMap, the highest precision, recall and F-measure among the others, by 29%, 60% and 58% in terms of precision, recall and F-measure respectively. The use of semantic similarity in OECM 1.1 significantly improves the matching results compared to the results of OECM 1.0. For instance, when matching

[20] `http://alignapi.gforge.inria.fr/`

Table 3. State-of-the-art comparison results. Bold entries are the top scores.

Approaches	Conference$_{de}$ × Ekaw$_{en}$			Conference$_{de}$ × Edas$_{en}$		
	Precision	Recall	F-measure	Precision	Recall	F-measure
AML [7]	0.56	0.20	0.29	0.86	0.35	0.50
KEPLER [16]	0.33	0.16	0.22	0.43	0.18	0.25
LogMap [15]	0.71	0.20	0.31	0.71	0.29	0.42
XMap [28]	0.18	0.16	0.17	0.23	0.18	0.20
OECM 1.0 [14]	0.75	0.67	0.71	0.93	0.76	0.84
OECM 1.1	**1.00**	**0.80**	**0.89**	**1.00**	**0.78**	**0.88**
	Conference$_{ar}$ × Ekaw$_{en}$			Conference$_{ar}$ × Edas$_{en}$		
AML [7]	0.64	0.39	0.28	0.71	0.42	0.29
KEPLER [16]	0.40	0.30	0.24	0.40	0.30	0.24
LogMap [15]	0.40	0.13	0.08	0.40	0.18	0.12
XMap [28]	**1.00**	0.0	0.0	**1.00**	0.00	0.00
OECM 1.1	**1.00**	**0.50**	**0.67**	0.86	**0.67**	**0.75**
	Conference'$_{ar}$ × Ekaw$_{en}$			Conference'$_{ar}$ × Edas$_{en}$		
OECM 1.1	0.88	**0.70**	**0.78**	**1.00**	**0.78**	**0.88**

Conference$_{de}$ × Ekaw$_{en}$, matching results in OECM 1.0 have been enhanced by 25%, 13%, and 18% in terms of precision, recall and F-measure respectively. When matching Conference$_{ar}$ × Edas$_{en}$, XMap outperform OECM by 14% in terms of precision, while OECM outperforms it in both recall and f-measure. It is observed that the precision of OECM slightly decreased because of the linguistic mistakes found in Conference$_{ar}$. When considering Conference'$_{ar}$, which has the linguistic correction, as a source ontology in this matching, the matching results are improved.

Evaluating the Enrichment Process. According to [4], the enriched ontology can be evaluated by comparing it against a predefined reference ontology (Gold standard). In this experiment, we evaluate the enriched ontology SEO$_{en-de}$ (cf. section 4). A gold standard ontology has been manually created by ontology experts. By comparing SEO$_{en-de}$ with the gold standard, OECM achieves 1.00, 0.80, and 0.89 in terms of precision, recall, and F-measure respectively. This finding confirms the usefulness of our approach in cross-lingual ontology enrichment.

6 Conclusion

We present a fully automated approach, OECM, for building multilingual ontologies. The strength of our contribution lies on building such ontologies from monolingual ones using cross-lingual matching between ontologies concepts. Indo and non-Indo-European languages resources are used for enrichment in order to illustrate the robustness of our approach. Considering multiple translations of concepts and the use of semantic similarity measures for selecting the best translation have significantly improved the quality of the matching process. Iterative triple retrieval process has been developed to determine which information, from

the source ontology, can be added to the target ontology, and where such information should be added. We show the applicability of OECM by presenting a use case for enriching an ontology in the scholarly communication domain. The results of the cross-lingual matching process are found promising compared to five state-of-the-art approaches, involving the previous version of OECM. Furthermore, evaluating the quality of the enrichment process emphasizes the validity of our approach. Finally, we propose some linguistic corrections for the Arabic ontologies in the MultiFarm benchmark that used in our experiment, which considerably enhanced the matching results. In conclusion, our approach provides a springboard for a new way to build multilingual ontologies from monolingual ones. In the future, we intend to further consider properties and individuals in the enrichment process. In addition, we aim to apply optimization methods in order to evaluate the efficiency of OECM when enriching very large ontologies.

Acknowledgments

This work has been supported by the BOOST EU project no. 755175. Shimaa Ibrahim and Said Fathalla would like to acknowledge the Ministry of Higher Education (MoHE) of Egypt for providing scholarships to conduct this study.

References

1. Ali, M., Fathalla, S., Ibrahim, S., Kholief, M., Hassan, Y.F.: Cloe: a cross-lingual ontology enrichment using multi-agent architecture. Enterprise Information Systems pp. 1–21 (2019)
2. Cheatham, M., Hitzler, P.: String similarity metrics for ontology alignment. In: International Semantic Web Conference. pp. 294–309. Springer (2013)
3. Cross, V.: Semantic similarity: A key to ontology alignment. In: Ontology Matching: OM-2018: Proceedings of the ISWC Workshop. p. 61 (2018)
4. Dellschaft, K., Staab, S.: On how to perform a gold standard based evaluation of ontology learning. In: International Semantic Web Conference. vol. 4273, pp. 228–241. Springer (2006)
5. Embley, D.W., Liddle, S.W., Lonsdale, D.W., Tijerino, Y.: Multilingual ontologies for cross-language information extraction and semantic search. In: International Conference on Conceptual Modeling. pp. 147–160. Springer (2011)
6. Espinoza, M., Gómez-Pérez, A., Mena, E.: Enriching an ontology with multilingual information. In: European Semantic Web Conference. pp. 333–347. Springer (2008)
7. Faria, D., Pesquita, C., Balasubramani, B.S., Tervo, T., Carrio, D., Garrilha, R., Couto, F.M., Cruz, I.F.: Results of aml participation in oaei 2018. In: Proceedings of the 13th International Workshop on Ontology Matching. pp. 125–131. CEUR-WS (2018)
8. Fathalla, S., Lange, C., Auer, S.: Eventskg: A 5-star dataset of top-ranked events in eight computer science communities. In: European Semantic Web Conference. pp. 427–442. Springer (2019)
9. Fathalla, S., Vahdati, S., Auer, S., Lange, C.: Seo: A scientific events data model. In: International Semantic Web Conference. p. In Press. Springer (2019)

10. Fu, B., Brennan, R.: Cross-lingual ontology mapping and its use on the multilingual semantic web. MSW **571**, 13–20 (2010)
11. Fu, B., Brennan, R., OSullivan, D.: A configurable translation-based cross-lingual ontology mapping system to adjust mapping outcomes. Web Semantics: Science, Services and Agents on the World Wide Web **15**, 15–36 (2012)
12. Gracia, J., Montiel-Ponsoda, E., Cimiano, P., Gómez-Pérez, A., Buitelaar, P., McCrae, J.: Challenges for the multilingual web of data. Web Semantics: Science, Services and Agents on the World Wide Web **11**, 63–71 (2012)
13. Hertling, S., Paulheim, H.: Wikimatch: using wikipedia for ontology matching. Ontology Matching **946** (2012)
14. Ibrahim, S., Fathalla, S., Yazdi, H.S., Lehmann, J., Jabeen, H.: Oecm: A cross-lingual approach for ontology enrichment. In: European Semantic Web Conference. p. In Press. Springer (2019)
15. Jiménez-Ruiz, E., Grau, V.C.: Logmap family participation in the oaei 2018. In: Proceedings of the 13th International Workshop on Ontology Matching. pp. 187–191. CEUR-WS (2018)
16. KACHROUDI, M., DIALLO, G., YAHIA, S.B.: Oaei 2018 results of kepler. In: Proceedings of the 13th International Workshop on Ontology Matching. pp. 173–178. CEUR-WS (2018)
17. Lehmann, J., Sejdiu, G., Bühmann, L., Westphal, P., Stadler, C., Ermilov, I., Bin, S., Chakraborty, N., Saleem, M., Ngomo, A.C.N., et al.: Distributed semantic analytics using the sansa stack. In: International Semantic Web Conference. pp. 147–155. Springer (2017)
18. Lin, F., Krizhanovsky, A.: Multilingual ontology matching based on wiktionary data accessible via sparql endpoint. In: RCDL (2011)
19. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. pp. 55–60 (2014), `http://www.aclweb.org/anthology/P/P14/P14-5010`
20. Meilicke, C., GarcíA-Castro, R., Freitas, F., Van Hage, W.R., Montiel-Ponsoda, E., De Azevedo, R.R., Stuckenschmidt, H., ŠVáB-Zamazal, O., Svátek, V., Tamilin, A., et al.: Multifarm: A benchmark for multilingual ontology matching. Web Semantics: Science, Services and Agents on the World Wide Web **15**, 62–68 (2012)
21. Mihalcea, R., Corley, C., Strapparava, C., et al.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI. vol. 6, pp. 775–780 (2006)
22. Niwattanakul, S., Singthongchai, J., Naenudorn, E., Wanapu, S.: Using of jaccard coefficient for keywords similarity. In: Proceedings of the International MultiConference of Engineers and Computer Scientists. vol. 1 (2013)
23. Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A., Zavitsanos, E.: Ontology population and enrichment: State of the art. In: Knowledge-driven multimedia information extraction and ontology evolution. pp. 134–166. Springer-Verlag (2011)
24. Spohr, D., Hollink, L., Cimiano, P.: A machine learning approach to multilingual and cross-lingual ontology matching. In: International Semantic Web Conference. pp. 665–680. Springer (2011)
25. Tigrine, A.N., Bellahsene, Z., Todorov, K.: Light-weight cross-lingual ontology matching with lyam++. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 527–544. Springer (2015)
26. Trojahn, C., Fu, B., Zamazal, O., Ritze, D.: State-of-the-art in multilingual and cross-lingual ontology matching. In: Towards the Multilingual Semantic Web, pp. 119–135. Springer (2014)

27. Trojahn, C., Quaresma, P., Vieira, R.: A framework for multilingual ontology mapping (2008)
28. Warith Eddine DJEDDI, S.B.Y., KHADIR, M.T.: Xmap results for oaei 2018. In: Proceedings of the 13th International Workshop on Ontology Matching. pp. 210–215. CEUR-WS (2018)