Measuring Clusters of Labels in an Embedding Space to Refine Relations in Ontology Alignment

Molka Tounsi Dhouib, Catherine Faron, Andrea G. B. Tettamanzi

Noname manuscript No.
(will be inserted by the editor)

Measuring Clusters of Labels in an Embedding Space to Refine Relations in Ontology Alignment

Molka Tounsi Dhouib · Catherine Faron · Andrea G. B. Tettamanzi

Accepted: 23 September 2021

Abstract Ontology alignment plays a key role in the management of heterogeneous data sources and metadata. In this context, various ontology alignment techniques have been proposed to discover correspondences between the entities of different ontologies. This paper proposes a new ontology alignment approach based on a set of rules exploiting the embedding space and measuring clusters of labels to discover the relationship between entities. We tested our system on the OAEI conference complex alignment benchmark track and then applied it to aligning ontologies in a real-world case study. The experimental results show that the combination of word embedding and a measure of dispersion of the clusters of labels, which we call the radius measure, makes it possible to determine, with good accuracy, not only equivalence relations, but also hierarchical relations between entities.

Keywords Ontology Alignment · Word Embedding · Semantic Web

1 Introduction

With the importance and the exponential growth of business data and Web volumes, the exploitation of ontologies in applications has become crucial, in order to make it possible to share and reuse knowledge (Ochieng and Kyanda 2018). As a consequence, the number of ontologies developed for a given domain has increased and several ontologies exist in the same or different domains with some overlap and some level of heterogeneity among them. Many reasons can explain this: (i) there are different actors with different interests, (ii) (ii) these actors are using different methodologies, different tools to design their ontologies, and they may express their knowledge with different levels of details (Euzenat et al. 2007). As a result, Ontology alignment is thus a crucial yet difficult task to deal with this heterogeneity and achieve interoperability on the Semantic Web. (Euzenat et al. 2007). As a result, Ontology alignment is thus a crucial yet difficult task to deal with this heterogeneity and achieve interoperability on the semantic web.

Ontology alignment is the task of finding the correspondence between entities of two ontologies (i.e between concepts, or classes or properties). This correspondence is the semantic mapping of one entity in the source ontology to one entity in the target ontology. It is usually expressed as a (perhaps loose) equivalence relation, but its exact nature might quite often be better described in terms of a hierarchical relation.

Formally, we adopt the ontology alignment definition introduced by (Euzenat et al. 2007; Shvaiko and Euzenat 2011). A correspondence between a source ontology O_1 and a target ontology O_2 is defined as a tuple $\{(e_1, e_2, r, con)\}$, where:

- e_1 is an entity in O_1,
- e_2 is an entity in O_2,
- r is the semantic relationship between e_1 and e_2 such as equivalence($\equiv$), more general ($\sqsupseteq$), and

M. Tounsi
Université Côte d'Azur, Inria, CNRS, I3S, Sophia Antipolis, France
E-mail: dhouib@i3s.unice.fr

C. Faron
Université Côte d'Azur, Inria, CNRS, I3S, Sophia Antipolis, France
E-mail: faron@i3s.unice.fr A. Tettamanzi
Université Côte d'Azur, Inria, CNRS, I3S, Sophia Antipolis, France
E-mail: tettamanzi@i3s.unice.fr

– *con* is the confidence score (typically in the [0, 1] range) holding for the correspondence between e_1 and e_2.

The most commonly used semantic relations are equivalence and subsumption relations. For OWL ontologies we can use *owl:equivalentClass* for equivalent alignment of classes, *owl:sameAs* for equivalence alignment of individuals and *rdfs:subClassOf* for subsumption alignment of classes. For SKOS vocabularies, we can use *skos:narrowMatch* and *skos:broadMatch* for hyponymy relations between concepts, and *skos:exactMatch* or *skos:closeMatch* for synonymy relations. Alignments can be of various cardinalities: (i) one-to-one (1:1), (ii) one-to-many (1:m), (iii) many-to-one (n:1), or (iv) many-to-many (n:m). There are two kinds of matches: (i) a simple match is about linking two atomic entities represented by their identifiers; and (ii) a complex match allows to express logical formulas between entities (Thiéblin et al. 2017).

To better appreciate the subtleties involved in describing correspondences between entities of different ontologies, one might consider, for example, pairs of entities like (i) "IT consultant" vs. "Information system consultant"; (ii) "Computer programming" vs. "Language and Programming software"; and (iii) "Computer programming" vs. "Computer programming services".

Various ontology alignment techniques are used to discover the semantic relations between entities of different ontologies. These techniques focus on special features ranging from lexical information to semantic information through structural and external information.

In this article, we address the following research questions:

– *How can we align two ontologies?*
– *How can we define a similarity measure between entities?*
– *How can we refine the nature of the relationship between two entities?*

We propose a novel approach to ontology alignment based on a set of rules exploiting mainly semantic information using a similarity measure defined in the embedding space of a word embedding. The underlying assumptions behind our approach are: (i) all the labels of the entities which share the same parents are close to each other in the embedding space; (ii) each entity in an ontology can be represented as a cluster of its instances in the embedding space and such a cluster can be described by its centroid and its radius (Ristoski et al. 2017; Alshargi et al. 2018b,a); (iii) a cluster whose radius is smaller than the radius of another cluster whose centroid coincides or is very close to its centroid is likely to represent a specialization of the entity associated with the broader cluster.

Our major contribution includes: (i) our capability to handle not only the equivalence relationship, but also the hierarchical relationship between entities; (ii) the introduction of the radius notion as a dispersion measurement of a label cluster that enables to refine the nature of the relationship (equivalence or hierarchical) between two matching entities; (iii) our capability to discover rich *n-m* relationships between entities; (iv) the evaluation of our system on several open datasets in English from the Ontology Alignment Evaluation Initiative (OAEI)[1] benchmark and two real-world cases studies provided by the *Silex* company[2] and *ONISEP*[3] requiring to match datasets in French.

This paper is organized as follows: Section 2 presents the related works. Section 3 describes our ontology alignment approach. Section 4 reports and discusses the results of our experiments on several datasets. Section 5 concludes with an outline of future work.

2 Related Works

A variety of ontology alignment techniques has been presented in the literature, and probably over a hundred different alignment systems exist to date. Due to this very wide scope, we cannot provide an exhaustive account of all research directions in this domain. Instead, we focus on giving an overview of alignment techniques with some references of systems. Several surveys on ontology alignment techniques have been written (Ardjani et al. 2015; Euzenat et al. 2007; Kalfoglou and Schorlemmer 2003; Otero-Cerdeira et al. 2015; Shvaiko and Euzenat 2005; Rahm and Bernstein 2001; Doan and Halevy 2005). Most of these surveys focus on input and process dimensions to classify the ontology alignment techniques. Doan and Halevy (2005) consider both input and process dimensions and differentiate in their classification between: (i) rule-based techniques that exploit schema-level information in specific rules; and (ii) learning-based techniques that exploit data instance information with machine-learning or statistical analysis. However, Rahm and Bernstein (2001) analyze the two dimensions in a different way. For the input dimension they distinguish between instance classification matchers (i.e. exploiting information from the TBox) and schema classification matchers (i.e. exploiting information from the ABox). For the process dimension they introduce classification axes such as element vs structure

[1] http://oaei.ontologymatching.org/
[2] https://www.Silex-france.com/Silex/
[3] http://www.onisep.fr/

or linguistic vs constraint-based. But the most complete and extensive classification of ontology alignment techniques available to date is probably the one proposed by Euzenat et al. (2007). This classification introduces some additional criteria to further detail the different aspects of matching techniques (i.e. granularity of the matcher, the interpretation of the input information, the origin and the kind of input information). We have adopted this last classification in the rest of this article, although we are aware that it presents some limitations.

Most researchers on ontology alignment has focused on engineering features from lexical, structural information, and external resource (Kolyvakis et al. 2018). Lexical information computes the similarities between the lexical information of the entities. Among the systems which use this kind of information, we can mention RIMOM (Li et al. 2008), ASMOV (Jean-Mary et al. 2009), AgreementMaker (Cruz et al. 2009), COMA (Do and Rahm 2002), COMA++ (Aumueller et al. 2005), OLA (Euzenat and Valtchev 2004), Anchor-Prompt- (Noy and Musen 2001), S-Match (Giunchiglia et al. 2004) and (Monge et al. 1996). Structural information consider the position of the entities in the graph and their relations with others entities. Among these systems we can mention: Yam++ (Ngo and Bellahsene 2012), MEDLEY (Hassen 2012), Cupid (Madhavan et al. 2001), Anchor-Prompt (Noy and Musen 2001), COMA (Do and Rahm 2002), OLA (Euzenat and Valtchev 2004), QOM (Ehrig and Staab 2004), RiMOM (Li et al. 2008). Despite the fact that lexical and structural information are widely used in ontology alignment, these techniques suffer from their weakness in capturing the semantics of lexical information of entities. To overcome this problem, many systems consider extensional information which involves exploiting an auxiliary resource, such as WordNet, to add lexical relationships (e.g. synonym, antonyms, hypernyms or hyponyms) to the system (Mohammadi et al. 2018). Many systems consider linguistic-based similarities such as AROMA (David 2007), Falcon (Jian et al. 2005), OLA (Euzenat and Valtchev 2004), Cupid (Madhavan et al. 2001), COMA (Do and Rahm 2002).

Many others attempts have been made to use representation learning technique for ontology alignment. Word embedding techniques are now used more and more in the ontology alignment task (Zhang et al. 2014; Vieira and Revoredo 2017; Kolyvakis et al. 2018; Lastra-Díaz et al. 2019). The first approach that explored word embedding in the ontology alignment task is described by (Zhang et al. 2014). The authors proposed a hybrid method to combine word embedding and the edit distance together. The matching strategy is to consider the maximum similarity, i.e to return for every entity in the source ontology the most similar entity in the target ontology. (Nkisi-Orji et al. 2018) introduce a classifier-based approach for ontology alignment which combines string-based similarity, semantic similarity, and semantic context. Word embedding was used to generate semantic features for a random forest classifier. (Kolyvakis et al. 2018) use information from ontologies and additional knowledge sources to extract synonymy and antonymy relations. These information are then used to refine and adapt pre-trained word vectors to compute the similarity distance between entities. (Schmidt et al. 2018) compare two similarity measures for synset disambiguation: (i) the Lesk measure (Lesk 1986) and (ii) the distance between word embedding to match domain and top-level ontologies. Based on their experiments, the authors show that the results obtained using word embedding are better than the results obtained with Word Sense Disambiguation. (Gromann and Declerck 2018) use a multilingual word embedding for multilingual ontology alignment.

Inspired by word embedding, knowledge graph embedding methods have been explored for ontology alignment. Knowledge graph embedding consists in learning a continuous vector space for each entity (node or edge) of a graph. As a result, similar entities have similar vector representations. MTransE (Chen et al. 2016), IPTransE (Sun et al. 2017) and BootEA (Sun et al. 2018) are three systems using knowledge graph embeddings to compare entities.

When compared to the state of the art, we propose a hybrid approach combining three types of information: (i) lexical, (ii) structural, and (iii) semantic information to align ontologies. Our first challenge was to fit with the real-world use cases of the *Silex* company. The analysis of the *Silex* data showed that the labels of the entities of ontologies to be aligned are not very close at the lexical level. Therefore, string-based metrics are not very useful in this case. Then we moved towards word embedding. We experimented training our own embedding model, but we got poor results as the available corpus is not rich enough. Finally we decided to use the fastText model as it is the only model that provides word embedding for French. Additionally, we considered extracting the semantics of the concepts based on the structure of the ontology. According to Aristotle's fundamental predictive theory, the semantics of a concept is mainly defined by the difference between this concept and its genus, or more generally its ascendants in the ontology (Parrochia and Neuville 2014). Therefore, in the ontology alignment literature, several works use information associated to more general concepts when searching matchings between two concepts, as this generalization of concepts is bringing more con-

text. In our approach, we also consider taking into account the specialization of concepts when computing matchings, considering that more specific concepts will also bring additional context and semantics, as previously investigated by Giunchiglia et al. (2007).

3 Overview of Our Approach to Ontology Alignment

3.1 Problem Statement

The goal of ontology alignment is to discover the relationships between entities of ontologies.

Our ontology alignment approach is based on the notion of cluster. Broadly speaking, a cluster is a collection of data objects that are more similar to one another than to any object that does not belong to it. As we will see below in Section 3.5, we will give a specific definition of this notion in the context of the proposed approach.

Our alignment process, illustrated in Figure 1, is a hybrid approach combing lexical information, structural information and semantic information expressed in the embedding space to refine the nature of the relationship between entities. In the rest of this section, we detail the four successive steps of our approach. We consider indifferently RDFS, OWL or SKOS vocabularies, and two languages, namely French and English. The language must be chosen at the beginning of the alignment process to ensure that the right word embedding model is selected.

3.2 Extracting Lexical and Structural Information from Ontologies

The first step of our approach consists in the extraction of lexical information and structural information from the ontologies to be aligned. To achieve this, the two ontologies are parsed with rdflib and queried with the SPARQL query shown in Listing 1.

Lexical information is extracted from the values of the properties `rdfs:label` for RDFS or OWL ontologies or `skos:prefLabel` for SKOS vocabularies.

Structural information is captured by associating the labels of all child entities to their parent entities, considering `rdfs:subClassOf` or `rdfs:subPropertyOf` properties instead of `skos:broader`. As a result, we consider clusters of entities specializing the root entity in each cluster.

While it could be interesting to also exploit other types of information on entity (e.g. property domains and ranges) in the alignment process, this is out of the scope of our current proposal.

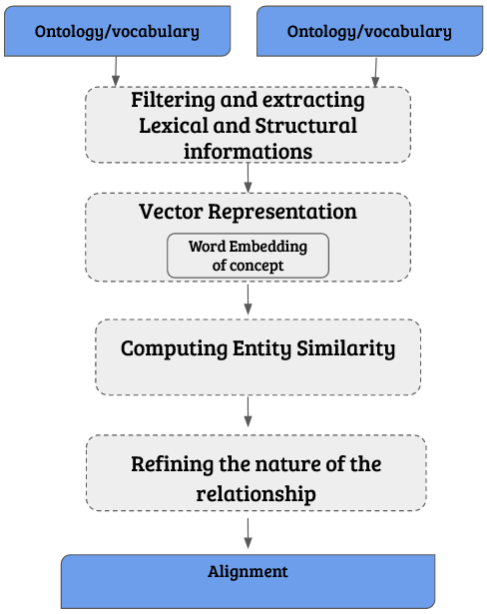

Fig. 1 Workflow of the proposed ontology alignment approach.

Listing 1 SPARQL query to extract lexical and structural information from a SKOS vocabulary

```
SELECT ?uri ?label
       (group_concat
       (DISTINCT ?mid_label; separator=":")
        AS ?lineage)
WHERE {
        ?uri skos:prefLabel ?label
         FILTER (lang(?label)='fr')
        ?uri ^skos:broader* ?mid.
        ?mid skos:prefLabel ?mid_label.
        FILTER (lang(?mid_label)='fr')
      } GROUP BY ?mid ORDER BY count(?label)
```

Let us illustrate it using the hierarchy of Figure 2 as an example:

- lexical_information(#61) = {Telecommunications}
- structural_information(#61) = {Telecommunications, Wired telecommunications activities, Wireless telecommunications activities, Satellite telecommunication activities, Other telecommunications activities}.
- lexical_information(#J) = {Information and communication}.
- structural_information(#J) = {Information and communication, Publishing activities, Computer programming, consultancy and related activities, Telecommunications, Wired telecommunications activities, Wireless telecommunications activities, Satellite telecommunication activities, Other telecommunications activities}.

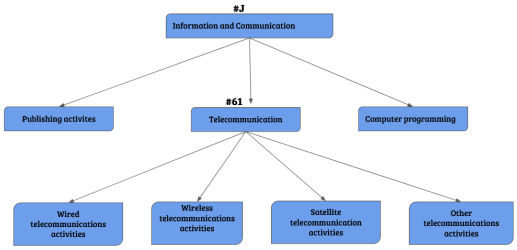

Fig. 2 An example of a hierarchy of concepts.

3.3 Computing Word Embedding Representations

Based on the extracted information, we compute the word embedding representation of entities. We define two types of vector representations: (i) the vector representation of an entity (lexical information) and (ii) the vector representation of a cluster of entities (structural information).

We use the pre-trained word vectors for French and English, learned using fastText[4] on a Wikipedia dump. The French model contains 1,152,449 tokens, and the English model contains one million tokens. Both are mapped to 300-dimensional vectors (Mikolov et al. 2013).

A pre-processing step is necessary to convert words to lower case and remove all stop words.

The process of computing the vector representation of the entities is similar to creating the vector representation of sentences since in several cases the label of an entity is composed of multiple words. So the vector representation of the entity is computed by averaging the word embedding vectors along each dimension of all the words contained in its label and occurring in the dictionary:

$$entityWordEmbedding(c) = \frac{1}{n}\sum_{i=1}^{n} w_i, \quad (1)$$

where n is the number of words in the dictionary occurring in the label of an entity c and $w_i \in R^{300}$ denotes the word embedding vector of the ith such word (if a word in a label does not appear in the dictionary, it is just ignored). The vector representation of a cluster of entities is constructed by averaging the word embedding vector representations of the entities belonging to it:

$$clusterWordEmbedding(cl) = \frac{1}{k}\sum_{i=1}^{k} entityWordEmbedding(c_i), \quad (2)$$

where c_i is an entity in the cluster cl and $k = |cl|$.

[4] https://fasttext.cc/docs/en/pretrained-vectors.html

Listing 2 Pseudo-code to search for matching entities

```
input: source ontology O1,
       target ontology O2,
       threshold_sim
output: list of correspondences
  list=null
  for each e1 in O1 do
    for each e2 in O2 do
        sim=cosine_sim{O1,O1}
        if sim> threshold_sim then
           list.append (e1,e2, sim)
        end if
    end for
  end for
```

3.4 Searching for Matching entities

The semantic similarity between an entity of the source ontology and an entity of the target ontology is calculated by considering their vector representations. The common similarity metric for embeddings is the cosine similarity measure[5]. We consider that a correspondence exists between two entities when the cosine similarity between them is bigger than a given threshold. Our algorithm aims at collecting all the possible correspondences between entities to propose many-to-many mappings: one entity from an ontology can correspond to more than one entity in the other ontology. Listing 2 shows the pseudo code of our algorithm to discover the correspondences.

3.5 Refining the Nature of the Relationship Between Two Matching entities

We begin by defining the notion of a cluster in the context of our method. By *cluster*, we mean here a set of vector representations w_i of labels that are all directly or indirectly subsumed by the same *root* entity (or concept) and are closer to it than any other labels subsumed by it but not included in the set. Thus, if we refer once more to Figure 2, the vector representing the label "Information and communication" would constitute a singleton cluster, having concept #J as its root; the four vectors representing the labels "Information and communication", "Publishing activities", "Telecommunication", and "Computer programming" would constitute another cluster having the same concept #J as its root; and the five vectors representing the labels "Telecommunication", "Wired telecommunication activities", "Wireless telecommunication activities", "Satellite telecommunication activities", and "Other telecommunication activities" would constitute

[5] https://en.wikipedia.org/wiki/Cosine_similarity

a cluster having concept #61 as its root. In all cases, the entities represented by the cluster members are more closely related to one another, from a hierarchical point of view, than to any other entity, because they share a common ancestor.

At this stage, for each entity in the source ontology we have a list of matching entities in the target ontology. We must now decide of the nature of the relationships holding between entities of the source and target ontologies: an equivalence relationship or a hierarchical relationship depending on the degree of similarity between two matching entities, considering the clusters of which they are the root.

More precisely, the relationship between two matching entities e_1 and e_2 is refined by comparing the radii of their respective embedding vector clusters, computed by taking into account the hierarchical structure of the two ontologies: The radius of a cluster is the maximum distance between the centroid of the cluster and all the other entities in the cluster. We define the radius of a cluster of entities as the standard deviation of their cosine dissimilarity with respect to the centroid:

$$radius = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{w_i \cdot \overline{w}}{|w_i| \cdot |\overline{w}|}\right)^2}, \tag{3}$$

where $w_i \in R^{300}$ is the vector representation of the i^th entity in the cluster, N is the size of the cluster, and $\overline{w} \in R^{300}$ is the centroid of the cluster, defined as

$$\overline{w} = \frac{1}{N}\sum_{i=1}^{N} w_i.$$

Figure 3 shows two example clusters associated to entity *Information and communication* (the bigger circle) and (the smaller circle). To define the type of the relationship we compare the radii of two matching clusters. These two clusters are formed mainly using structural information. We suppose that the cluster whose result has the smallest average distance between a label and the centroid is in broader relation with the cluster which has the largest radius. As shown in Figure 3, the blue circle (which represents the cluster of telecommunication service, voice mail administration service, and mobile phone administration services) is in broader relation with the big circle (which represent the cluster including telecommunications, wired telecommunications activities, and satellite telecommunications activities).

We define the two following rules to identify the relationship holding between two similar entities:

$$\begin{aligned} &|radius(e_1) - radius(e_2)| < 0.1 \\ &\qquad \Rightarrow e_1\ closeMatch\ e_2 \end{aligned} \tag{4}$$

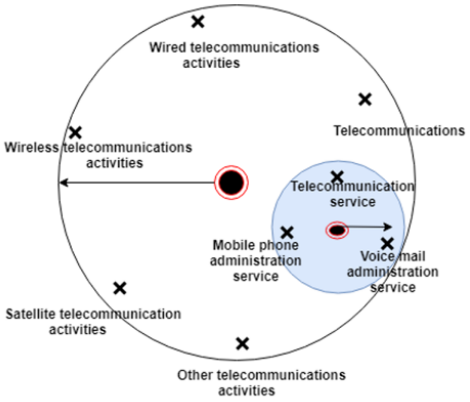

Fig. 3 Two example clusters of entities, one included into the other.

$$\begin{aligned} &|radius(e_1) - radius(e_2)| > 0.1 \\ &\qquad \Rightarrow e_1\ narrowMatch\ e_2 \\ &\qquad \wedge e_2\ broadMatch\ e_1 \end{aligned} \tag{5}$$

In particular, the first condition above is trivially satisfied when both e_1 and e_2 are leaf nodes of their respective ontologies and their radii are both zero.

We represent equivalence relationships by using *owl:sameAs* properties for when aligning RDFS or OWL vocabularies and *skos:closeMatch* properties when aligning SKOS vocabularies. We represent hierarchical relationships by using *rdfs:subClassOf* or *rdfs:subPropertyOf* properties for RDFS or OWL vocabularies and *skos:broader* and *skos:narrower* for SKOS vocabularies.

4 Experiments

4.1 Datasets

4.1.1 Experiments on Task-Oriented Complex Alignment on Conference Organization

We experimented our approach on the conference complex alignment benchmark (Thieblin 2019) for ontology merging. We chose it because it contains not only equivalence relations but also hierarchical relations. This benchmark has been constructed within the framework of the OAEI and contains 57 correspondences and five ontologies (*cmt*, *conference*, *confOf*, *edas*, *ekaw*) available in OWL format. Table 1 summarizes the number of entities by type contained in these ontologies (Thiéblin et al. 2018).

Table 1 Number of entities by type for each ontology.

Ontology	Classes	Object properties	Data properties
cmt	30	49	10
conference	60	46	18
confOf	39	13	23
edas	104	30	20
ekaw	74	33	0

4.1.2 Silex Use Case

In the context of a collaboration with `Silex`, a french company offering sourcing solutions, we experimented the proposed approach on the vocabularies gathered for their use case in the Information Technology sector (IT).

`Silex` develops a SaaS sourcing tool for the identification of the service providers that are best suited to meet some service requests expressed by companies. The `Silex` platform allows companies to provide a textual description of their professional activities, their offers and the services they are looking for. To help to do this recommendation, an important step into the process of `Silex` is to build an ontology to represent the `Silex` knowledge. For that an ontology engineering process is conducted to semantically annotate the textual description of companies and service requests with three types of knowledge: (i) skills, (i) occupations and (iii) business sectors.

The `Silex` ontology is built by combining several of meta-data repositories such as ESCO,[6] ROME,[7] Cigref,[8] NAF,[9] UNSPSC,[10] and an internal `Silex` business sectors repository. A manual alignment task was carried out by an expert in the Silex company to establish correspondences between these metadata: (i) ESCO to Cigref, (ii) ESCO to ROME, (iii) NAF to UNSPSC. Table 2 presents the number of concepts in each of the modules building up the `Silex` vocabulary for the computing sector, and Table 3 presents the number alignment per relation. We consider the set of the manually stated alignments as a test-bed for the automatic alignment approach we propose.

4.1.3 ONISEP Use Case

ONISEP (Office national d'information sur les enseignements et les professions) is a State operator that reports to the Ministry of National Education and Youth and the Ministry of Higher Education, Research and Innovation. As a public publisher, ONISEP produces and distributes all information on training and trades. It also offers services to students, parents and educational teams. In this context, ONISEP provided us with an occupation directory in XML format, and the goal was to align it with ROME. The ONISEP vocabulary contains 5325 concepts and the ROME vocabulary contains 12255 concepts. We started by transforming the ONISEP vocabulary into a SKOS vocabulary then we applied our approach to align ONISEP and ROME. A gold standard, composed of 290 links and produced by an expert, is used for the evaluation of our automatic alignment approach. It contains 259 close relations and 31 hierarchical relations.

Table 2 Number of concept for the IT sector ontology.

Skills and Occupations		Business sector	
Ontology	Number	Ontology	Number
ESCO	160	NAF	53
ROME	117	UNSPSC	153
Cigref	42		

Table 3 Number of relation types between concepts for the Silex ontology for the computing sector.

Ontologies	Relation types	Number
ESCO to ROME	Close	68 links
	Hierarchical	33 links
ESCO to Cigref	Close	24 links
	Hierarchical	31 links
NAF to UNSPSC	Close	21 links
	Hierarchical	54 links

4.2 Evaluation Protocol

The performances of our approach are measured by calculating precision, recall and F-measure Ochieng and Kyanda (2018).

Precision (P) is used to check the degree of correctness of the ontology alignment algorithm. It is calculated as shown in Equation 6:

$$\text{precision} = \frac{\text{correct correspondences}}{\text{total returned correspondences}}. \quad (6)$$

Recall (R) is used to check the degree of completeness of the ontology alignment algorithm. It is calculated as shown in Equation 7:

$$\text{recall} = \frac{\text{correct correspondences}}{\text{expected correspondences}}. \quad (7)$$

[6] https://ec.europa.eu/esco/portal/home
[7] http://www.pole-emploi.org/accueil/mot-cle.html?tagId=94b2eaf6-d7bd-4244-bddc-01415605563b
[8] http://cigref.hr-ingenium.com/accueil.aspx
[9] https://www.insee.fr/fr/information/2406147
[10] https://www.unspsc.org/

F-measure (F1) is the harmonic average of recall and precision. It is calculated as shown in Equation 8:

$$\text{F-measure} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}. \quad (8)$$

In addition to this state-of-the-art evaluation method and taking into account the fact that our system was not designed to achieve a fully automatic matching process but rather to support end-users responsible for the sourcing task, by presenting a list of possible matches, we defined another evaluation method assuming that if a system is able to propose a list of k best possible matches which includes the correct match, we consider that the matching is correct. This way of evaluation does not only concern the precision metric but also the recall and F1 metrics since the correspondence is no longer considered as False Positive but as True Positive. We conducted the parameter learning (i.e threshold (Thr)) through 5-fold cross validation.

4.3 Results and discussion

4.3.1 Experiments on Task-Oriented Complex Alignment on Conference Organisation

We compared our matching results with the results of three state-of-the-art complex ontology matchers that were evaluated in Thiéblin et al. (2018), namely

1. Ritze *et al.* 2009 Ritze et al. (2009), a rule-based approach mostly relying on string similarity;
2. Ritze *et al.* 2010 Ritze et al. (2010), another rule-based approach using linguistic evidence
3. the KAOM system by Jiang *et al.* 2016 Jiang et al. (2016), using a probabilistic framework based on Markov Logic networks.

We searched the literature for other, more recent ontology alignment systems evaluated against the same benchmark, but we could not find any, probably due to the novelty of the benchmark. Table 4 shows that our system clearly outperforms the others on this benchmark, with a F1 of 0.27 and we can reach 0.51 using our evaluation methods, confirming the interest of looking at clusters of entities in an embedding space both to establish correspondences between them and to resolve the nature of their relations.

In addition to the evaluation protocol described in Thiéblin et al. (2018) where the performances of the systems are computed globally without distinguishing the type of the matching relationship, we have evaluated the performances of our approaches for each type of relationship. Table 5 summarizes the performance of our system on the OAEI benchmark depending on the relationship type. Depending on the ontologies to be aligned, the precision value ranges between 0.32 and 0.68 for the equivalence relation, and ranges between 0.06 to 0.52 for the subsumption relation. On the other hand, the recall value ranges between 0 and 0.7 for the equivalence relation, and ranges between 0 and 0.43 for the subsumption relation.

Table 4 Evaluation of our approach on the OAEI benchmark using the standard evaluation methods

Systems	P	R	F1
Our System	0.32	**0.31**	**0.27**
Ritze *et al.* 2009	0.30	0.13	0.19
Ritze *et al.* 2010	0.83	0.09	0.18
Jiang *et al.* 2016	0.09	0.11	0.10

Table 5 Evaluation of our approach using the standard evaluation methods depending on the relationship type of the OAEI benchmark

Relation Type	Precision	Recall	F1
Equivalence	0.29	0.35	0.32
Subsumption	0.02	0.02	0.02

Table 6 Evaluation of our approach on real world data using the standard evaluation methods

Dataset	Thr	P	R	F1
ESCO-ROME	0.85	0.49	0.74	0.58
ESCO-Cigref	0.80	0.51	0.72	0.59
NAF-UNSPSC	0.80	0.40	0.71	0.50
ONISEP-ROME	0.87	0.42	0.73	0.52

Table 7 Evaluation of our approach on real world data using our evaluation method

Dataset	Thr	P	R	F1
ESCO-ROME	0.70	0.99	0.94	0.96
ESCO-Cigref	0.80	0.92	0.72	0.80
NAF-UNSPSC	0.70	1.0	0.95	0.97
ONISEP-ROME	0.70	1.0	0.88	0.93

4.3.2 Silex and ONISEP use cases

Table 6 and 7 present the result of our system on real world data from *Silex* and ONISEP use cases. For the *Silex* data, the F1 value is around 0.5 and we can reach an F1 ranges between 0.8 and 0.97 using our evaluation

Table 8 Evaluation of our approach depending on the relationship type of *Silex* use cases using the standard evaluation methods

Dataset	Relation type	Thr	P	R	F1
ESCO-ROME	equivalence	0.70	0.77	0.40	0.51
	subsumption	0.86	0.60	0.24	0.32
ESCO-Cigref	equivalence	0.74	1.0	0.84	0.90
	subsumption	0.80	1.0	0.84	0.90
NAF-UNSPSC	equivalence	0.71	0.27	0.12	0.17
	subsumption	0.73	0.04	0.12	0.06

method. For the ONISEP data, the F1 value is 0.52 and we can reach an F1 of 0.93 using our evaluation methods. Table 8 summarizes the performance of our system on *Silex* real word data depending on the relationship type. Depending on the vocabularies to be aligned, the precision value ranges between (i) 0.37 and 0.46 for the equivalence relation, (ii) 0.04 and 1 for the subsumption relation. On the other hand, the recall value ranges between (i) 0.39 and 0.68 for the equivalence relation, (ii) 0.12 and 0.84 for the subsumption relation.

We also conducted some additional experiments in which we add the parent label to the label of a concept. We decided to conduct these experiments because we noted that several state-of-the-art proposals (Gracia and Mena 2012) have been made on the basis of such a bottom-up approach instead of a top-down approach. The experiment shows that the use of this information severely decreases the performance of our alignment system. For example, the F1 value when matching NAF and UNSPSC decreases from 0.50 to 0.11, and when matching ESCO and cigref it decreases from 0.60 to 0.1.

Although it looks like a dramatic step ahead with respect to the state of the art, our system still has much room for further improvement. There are four main issues that could be addressed:

1. The cosine similarity between some entities that should be matched is much lower than the matching threshold and as a consequence these matches are ignored. For example the cosine similarity between 'chair main ' and 'demo chair' is 0.37.
2. Our system is not designed to test hierarchical relations between two leaf nodes. This type of relationship must pass through the structural information to calculate the radius and, thus, infer the relationship. For example, in the benchmark, 'country' and 'location' are two leaf nodes that have been matched by `rdfs:subClassOf`.
3. Based on Equation 4, our system can assign an equivalence relation instead of a hierarchical relation because the threshold of the difference of radius between two classes is smaller than 0.1.
4. The quality of the embedding space depends on the context of the data and the similarity between the training data and the ontology data. Therefore, the quality of our system is tightly dependent on the embedding model.
5. A striking difference in performance between the identification of equivalence and subsumption is observed on the OAEI benchmark, while the performance for these two relationship types is much less diverse in real-world data. We believe the particularly poor performance of subsumption identification on the OAEI benchmark depends on the labels found in OAEI being highly specific, whereas the labels in real-world ontologies are more general.

5 Conclusion

We presented a novel approach of ontology alignment, based on measuring the clusters of labels in an embedding space to refine relations in ontology alignment. We reported the results of our experiments on multiples datasets: (i) the OAEI conference complex alignment benchmark, the real-world use case encountered by the Silex company, namely matching skills and competences from several ontologies in the IT field, (iii) the real-world use case encountered by the ONISEP, namely matching occupations between the ONISEP and the ROME vocabularies. These experiments show that our approach outperforms state-of-the-art approaches and is well suited to real world use cases, where the goal would be to propose possible alignments to experts that should be validated, as it is the case for *Silex* or ONISEP.

There are several directions for future work: (i) We aim to overcome the limitations of our approach when dealing with leaf nodes in ontologies. (ii) We aim at defining a specific set of pre-trained word vectors that best covers the Silex B2B use case and to compare the performance of a French word embedding model to a multilingual one which provides a cross-lingual word embedding. Alternatively, we could use BERT (Devlin et al. 2018), ELMO (Peters et al. 2018) or CamemBert(Martin et al. 2019) as models to generate word embedding, given their power to generate different word embedding that capture the context of a word (based on word order). (iii) We also plan to complete the evaluation of our system on the entire dataset, and at performing an empirical study to find the optimal threshold for the radius difference. (iv) We intend to experiment graph embedding techniques in order to consider all types of structure in the ontology. (v) Last but not least, we aim to consider other information types before doing the matching such as domain and range for RDFS or OWL ontologies. (vi) Another interesting perspective will be to compare our approach for ontology alignment with approaches for Named Entity Recognition and Linking.

References

Alshargi F, Shekarpour S, Soru T, Sheth A (2018a) Concept2vec: Metrics for evaluating quality of embeddings for ontological concepts. arXiv preprint arXiv:180304488

Alshargi F, Shekarpour S, Soru T, Sheth AP (2018b) Metrics for evaluating quality of embeddings for ontological concepts

Ardjani F, Bouchiha D, Malki M (2015) Ontology-alignment techniques: Survey and analysis. International Journal of Modern Education & Computer Science 7(11)

Aumueller D, Do HH, Massmann S, Rahm E (2005) Schema and ontology matching with coma++. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp 906–908

Chen M, Tian Y, Yang M, Zaniolo C (2016) Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. arXiv preprint arXiv:161103954

Cruz IF, Antonelli FP, Stroe C (2009) Agreementmaker: efficient matching for large real-world schemas and ontologies. Proceedings of the VLDB Endowment 2(2):1586–1589

David J (2007) Aroma: A method for the discovery of alignments between ontologies from association rules. PhD thesis, Thèse d'informatique. Université de Nantes. Nantes (FR). URL: http://tel. archives-ouvertes. fr/tel-00200040/en

Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805

Do HH, Rahm E (2002) Coma—a system for flexible combination of schema matching approaches. In: VLDB'02: Proceedings of the 28th International Conference on Very Large Databases, Elsevier, pp 610–621

Doan A, Halevy AY (2005) Semantic integration research in the database community: A brief survey. AI magazine 26(1):83–83

Ehrig M, Staab S (2004) Qom–quick ontology mapping. In: International Semantic Web Conference, Springer, pp 683–697

Euzenat J, Valtchev P (2004) Similarity-based ontology alignment in owl-lite. In: Proc. 16th european conference on artificial intelligence (ECAI), IOS press, pp 333–337

Euzenat J, Shvaiko P, et al. (2007) Ontology matching, vol 18. Springer

Giunchiglia F, Shvaiko P, Yatskevich M (2004) S-match: an algorithm and an implementation of semantic matching. In: European semantic web symposium, Springer, pp 61–75

Giunchiglia F, Yatskevich M, Shvaiko P (2007) Semantic matching: Algorithms and implementation. In: Journal on data semantics IX, Springer, pp 1–38

Gracia J, Mena E (2012) Semantic heterogeneity issues on the web. IEEE Internet Comput 16(5):60–67, DOI 10.1109/MIC.2012.116, URL https://doi.org/10.1109/MIC.2012.116

Gromann D, Declerck T (2018) Comparing pretrained multilingual word embeddings on an ontology alignment task. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)

Hassen W (2012) Medley results for oaei 2012. In: Proceedings of the 7th International Conference on Ontology Matching-Volume 946, CEUR-WS. org, pp 168–172

Jean-Mary YR, Shironoshita EP, Kabuka MR (2009) Ontology matching with semantic verification. Journal of Web Semantics 7(3):235–251

Jian N, Hu W, Cheng G, Qu Y (2005) Falcon-ao: Aligning ontologies with falcon. In: Proceedings of K-CAP Workshop on Integrating Ontologies, pp 85–91

Jiang S, Lowd D, Kafle S, Dou D (2016) Ontology matching with knowledge rules. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems XXVIII, Springer, pp 75–95

Kalfoglou Y, Schorlemmer M (2003) Ontology mapping: the state of the art. The knowledge engineering review 18(1):1–31

Kolyvakis P, Kalousis A, Kiritsis D (2018) Deepalignment: Unsupervised ontology matching with refined word vectors. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp 787–798

Lastra-Díaz JJ, Goikoetxea J, Taieb MAH, García-Serrano A, Aouicha MB, Agirre E (2019) A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art. Engineering Applications of Artificial Intelligence 85:645–665

Lesk M (1986) Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: Proceedings of the 5th annual international conference on Systems documentation, Citeseer, pp 24–26

Li J, Tang J, Li Y, Luo Q (2008) Rimom: A dynamic multistrategy ontology alignment framework. IEEE Transactions on Knowledge and data Engineering 21(8):1218–1232

Madhavan J, Bernstein PA, Rahm E (2001) Generic schema matching with cupid. In: vldb, Citeseer, vol 1, pp 49–58

Martin L, Muller B, Suárez PJO, Dupont Y, Romary L, de la Clergerie ÉV, Seddah D, Sagot B (2019) Camembert: a tasty french language model. arXiv preprint arXiv:191103894

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119

Mohammadi M, Atashin AA, Hofman W, Tan Y (2018) Comparison of ontology alignment systems across single matching task via the mcnemar's test. ACM Transactions on Knowledge Discovery from Data (TKDD) 12(4):51

Monge AE, Elkan C, et al. (1996) The field matching problem: Algorithms and applications. In: Kdd, vol 2, pp 267–270

Ngo D, Bellahsene Z (2012) Yam++: a multi-strategy based approach for ontology matching task. In: International Conference on Knowledge Engineering and Knowledge Management, Springer, pp 421–425

Nkisi-Orji I, Wiratunga N, Massie S, Hui KY, Heaven R (2018) Ontology alignment based on word embedding and random forest classification. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, pp 557–572

Noy NF, Musen MA (2001) Anchor-prompt: Using non-local context for semantic matching. In: OIS@ IJCAI

Ochieng P, Kyanda S (2018) Large-scale ontology matching: State-of-the-art analysis. ACM Computing Surveys (CSUR) 51(4):75

Otero-Cerdeira L, Rodríguez-Martínez FJ, Gómez-Rodríguez A (2015) Ontology matching: A literature review. Expert Systems with Applications 42(2):949–971

Parrochia D, Neuville P (2014) Taxinomie et réalité: vers une métaclassification. ISTE Group

Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. arXiv preprint arXiv:180205365

Rahm E, Bernstein PA (2001) A survey of approaches to automatic schema matching. the VLDB Journal 10(4):334–

350
Ristoski P, Faralli S, Ponzetto SP, Paulheim H (2017) Large-scale taxonomy induction using entity and word embeddings. In: Proceedings of the International Conference on Web Intelligence, ACM, pp 81–87
Ritze D, Meilicke C, Šváb-Zamazal O, Stuckenschmidt H (2009) A pattern-based ontology matching approach for detecting complex correspondences. In: ISWC Workshop on Ontology Matching, Chantilly (VA US), pp 25–36
Ritze D, Völker J, Meilicke C, Sváb-Zamazal O (2010) Linguistic analysis for complex ontology matching. In: CEUR Workshop Proceedings, RWTH, vol 689, pp Paper–1
Schmidt D, Basso R, Trojahn C, Vieira R (2018) Matching domain and top-level ontologies exploring word sense disambiguation and word embedding. In: Ontology Matching: OM-2018: Proceedings of the ISWC Workshop, p 1
Shvaiko P, Euzenat J (2005) A survey of schema-based matching approaches. In: Journal on data semantics IV, Springer, pp 146–171
Shvaiko P, Euzenat J (2011) Ontology matching: state of the art and future challenges. IEEE Transactions on knowledge and data engineering 25(1):158–176
Sun M, Zhu H, Xie R, Liu Z (2017) Iterative entity alignment via joint knowledge embeddings [c]. In: International Joint Conference on Artificial Intelligence. AAAI Press
Sun Z, Hu W, Zhang Q, Qu Y (2018) Bootstrapping entity alignment with knowledge graph embedding. In: IJCAI, vol 18, pp 4396–4402
Thieblin E (2019) Task-oriented complex alignments on conference organisation
Thiéblin E, Haemmerlé O, Hernandez N, Trojahn C (2017) Un jeu de données d'évaluation de correspondances complexes entre ontologies
Thiéblin É, Haemmerlé O, Hernandez N, Trojahn C (2018) Task-oriented complex ontology alignment: Two alignment evaluation sets. In: European Semantic Web Conference, Springer, pp 655–670
Vieira R, Revoredo K (2017) Using word semantics on entity names for correspondence set generation. In: OM@ ISWC, pp 223–224
Zhang Y, Wang X, Lai S, He S, Liu K, Zhao J, Lv X (2014) Ontology matching with word embeddings. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, Springer, pp 34–45