An Ontology Alignment Approach Combining Word Embedding and the Radius Measure

Molka Tounsi Dhouib[1,2](✉), Catherine Faron Zucker[1](✉), and Andrea G. B. Tettamanzi[1](✉)

[1] University Cote d'Azur, Inria, CNRS, I3S, Sophia Antipolis, France
{dhouib,faron,tettamanzi}@i3s.unice.fr
[2] Silex France, Gentilly, France

Abstract. Ontology alignment plays a key role in achieving interoperability on the semantic Web. Inspired by the success of word embedding techniques in several NLP tasks, we propose a new ontology alignment approach based on the combination of word embedding and the radius measure. We tested our system on the OAEI (http://oaei.ontologymatching.org/) conference track and then applied it to aligning ontologies in a real-world case study. The experimental results show that using word embedding and the radius measure make it possible to determine, with good accuracy, not only equivalence relations, but also hierarchical relations between concepts.

Keywords: Ontology alignment · Word embedding

1 Introduction

The Silex[1] company develops a SaaS sourcing tool for the identification of the service providers that are best suited to meet some service requests. The Silex platform allows companies to provide a textual description of their professional activities, their offers and the services they are looking for. The work presented in this paper has been carried out in the context of a collaboration between Silex and the I3S research laboratory, to add a semantic layer to the Silex B2B platform, in order to be able to automatically process the descriptions of service requests and improve the recommendation of relevant providers. An ontology engineering work has been conducted to semantically annotate the text descriptions of companies, offers, and service requests, with three kinds of knowledge: skills, occupations, and business sectors. We developed the Silex ontology by combining several meta-data repositories: ESCO,[2] ROME,[3] Cigref,[4]

[1] https://www.Silex-france.com/Silex/.
[2] https://ec.europa.eu/esco/portal/home.
[3] http://www.pole-emploi.org/accueil/mot-cle.html?tagId=94b2eaf6-d7bd-4244-bddc-01415605563b.
[4] http://cigref.hr-ingenium.com/accueil.aspx.

M. Acosta et al. (Eds.): SEMANTiCS 2019, LNCS 11702, pp. 191–197, 2019.
https://doi.org/10.1007/978-3-030-33220-4_14

NAF,[5] UNSPSC[6], Kompass[7] and an internal Silex business sectors repository. Currently, the Silex ontology covers only the Computer Science (CS) field [1]. Our aim now is to automatically align the entire vocabularies to extend the Silex ontology to all business sectors.

In this paper, we present a new approach to ontology alignment based on word embedding and inspired by an existing proposals [6]. We consider word embedding to represent concepts and we use it to compute not only equivalence relations between concepts but also hierarchical relations. We report our experiments on several open datasets from the Ontology Alignment Evaluation Initiative (OAEI) benchmark and the Silex use case.

This paper is organized as follows: related work is discussed in Sect. 2. Section 3 describes our algorithm for ontology alignment. Section 4 reports and discusses the results of our experiments on the Silex use case. Section 5 draws some conclusions and discusses our perspectives as future work.

2 Related Work

The main issue when using several ontologies is to deal with their semantic heterogeneity when combining them: each ontology has its own designer, its own knowledge area and its own level of details. Ontology alignment is thus a crucial yet difficult task to achieve interoperability on the semantic Web. It aims to discover the correspondences between the entities of different ontologies, and express them as equivalence or hierarchical relations.

There are two main ontology alignment techniques [2]: (i) Element-level techniques are meant to discover correspondences by calculating the surface similarity between lexical information of entities (usually labels), (ii) Structure-level techniques rely on the analysis of the neighbourhood of two entities in order to determine their similarity. Both techniques suffer from their weakness in capturing the semantics of lexical information of entities, and have been extended by exploiting external information sources, such as WordNet or Wikipedia. However, these auxiliary resources still suffer from the incompleteness and non exhaustiveness of their entries. To overcome this problem, the approach presented in [6] uses word embedding to preserve the semantic and syntactic similarities between words. This work mainly extract the lexical information (names, labels and comments of an entity) and search equivalence relations between this informations based on word embeddings similarity. In our work, we have been inspired by [6] to calculate the similarites between entities based only on their labels. We extended this approach by using cluster's radius to find equivalence and hierarchical relations between concepts.

[5] https://www.insee.fr/fr/information/2406147.

[6] https://www.unspsc.org/.

[7] http://www.kompass-international.com/Corporate/home.html.

3 Overview of Our Approach to Ontology Alignment

Our alignment process is based on a set of rules exploiting the word embedding similarity to discover the alignment. Our process is divided into four successive steps described in the following subsections. Our system supports two types of input (OWL ontologies and SKOS vocabulary), and two languages (French and English). But we can't work with both languages at the same time as we have a different word embedding model per language.

3.1 Extracting Lexical and Structural Information from Ontologies

We started by extracting two types of information from inputs: (i) lexical information (e.g., labels of concepts) and (ii) structural information (e.g., to associate the labels of all child entities to their parent entities). To achieve this, the two inputs (OWL or SKOS) are parsed with rdflib and queried with a SPARQL query. The Listing 1.1 shows an example of queries that handle with SKOS input and french language. The same query is used for owl ontologies by replacing *rdfs:label* instead of *skos:prefLabel* to extract the label of the class or the properties, and *rdfs:subClass* or *rfs:subproperties* instead of *skos:broader* to get the hierarchical relation between classes or properties.

Listing 1.1. SPARQL query to extract lexical and structural information from 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)
```

3.2 Computing Word Embedding Representations of Concepts

The second step of our approach is to compute the vector representations of concepts. We used a pre-trained word vectors for French and English, learned using fastText.[8] The French model contains 1,152,449 tokens, and the English model contains one million tokens. Both of them are mapped to 300-dimensional vectors [3].

The vector representation of a concept is constructed by averaging the word embedding vectors along each dimension of all the terms contained in its label and occurring in the dictionary $conceptWordEmbedding(c) = \frac{1}{n}\sum_{i=1}^{n} w_i$, where n is the number of words in the dictionary occurring in the label of a concept c and $w_i \in \mathbb{R}^{300}$ denotes the word embedding vector of the ith word. If a term does not appear in the dictionary, it is just ignored.

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

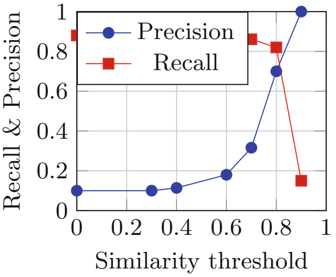

Fig. 1. Precision and recall as a function of the similarity threshold.

In the case of structural information, the vector representation of a cluster is given by averaging the word embedding vector representation of the label of the root concept (which is itself an average) with the vector representations of its child concepts $clusterWordEmbedding(cl) = \frac{1}{k}\sum_{i=1}^{k} conceptWordEmbedding(c_i)$, where k is the number of concepts in cluster cl.

3.3 Searching for Matching Concepts

We match every concept in the source ontology O_1 with the similar concept in the target ontology O_2 using the cosine similarity between vector representations of concept and cluster. The correspondence is then added to the alignment list based on the similarity threshold. Our algorithm aims at collecting all the possible correspondences between concepts. We empirically chose the threshold, by varying its value and calculating for each one the recall and precision measures. Figure 1 shows that an optimal trade-off of performance is achieved by setting the similarity threshold equal to 0.8.

3.4 Refining the Nature of the Relationship Between Two Matching Concepts

The result of the previous step is a list of matching concepts whose relationship must be made more precise. To link two concepts that are sufficiently similar, we used *skos:closeMatch* for SKOS and *owl:sameAs* for OWL. To define a hierarchical mapping link between two concepts, we used *skos:broader* or *skos:narrower* for SKOS and *rdfs:subClassOf* or *rdfs:subPropertiesOf* for OWL.

This relationship between two matching concepts is refined by comparing the radii of their respective embedding vector clusters formed mainly using structural information. The radius of a cluster is the maximum distance between all the vector representing the terms and the centroid. We define the radius of a cluster of concepts as the standard deviation of their cosine dissimilarity with respect to the centroid: $radius = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{w_i \cdot \overline{w}}{|w_i| \cdot |\overline{w}|}\right)^2}$, where $w_i \in \mathbb{R}^{300}$ is the vector representation of the $i$the concept in the cluster, N is the size of the cluster, and $\overline{w} \in \mathbb{R}^{300}$ is the centroid of the cluster, defined as $\overline{w} = \frac{1}{N} \sum_{i=1}^{N} w_i$. We suppose that the cluster whose result has the lowest average distance between a point and the centroid is in broader relation with the cluster which have the biggest radius. We decide of the relationship holding between two similar concepts by comparing their radii based on the following rules:

$$|radius(C1) - radius(C2)| < 0.1 \Rightarrow C1\ closeMatch\ C2 \tag{1}$$

$$\begin{aligned} |radius(C1) - radius(C2)| > 0.1 \Rightarrow C1\ narrowMatch\ C2 \\ \wedge\, C2\ broadMatch\ C1 \end{aligned} \tag{2}$$

4 Experiments

To evaluate the effectiveness of our approach, we performed experiments on two alignment datasets: (i) Task-oriented complex alignment on conference organisation and (ii) the Silex use case. The performances of our approach are measured by calculating precision, recall and F-measure [4].

4.1 Experiments on Task-Oriented Complex Alignment on Conference Organisation

To validate the proposed approach, we experimented it on a conference complex alignment benchmark[9], [10] for ontology merging, which has been constructed within the framework of the OAEI. This data set contains 57 correspondences made on five owl ontologies. Following the evaluation process presented in [5], we have taken into account only the alignments that exist in the complex data set and we ignored the alignment of simple data set. We assume that if our system is able to find the correct match between a proposed list, we consider that the entire proposed list is correct. This decision is justified by the fact that our system was designed to support end-users by presenting a list of possible matches. We compared our matching results with the results of three state-of-the-art systems that were mentioned in [5]: Our system clearly outperforms the others on this benchmark, with a precision value equals to O.89 and recall value equals to 0.69 compared to 0.83, and 0.13 for the best state-of-the-art system. Many reasons can explain our result: (i) the cosine similarity between classes is much smaller, as a consequence this match gets discarded than the threshold (cosine similarity (‘chair main’, ‘demo chair’ = 0)). (ii) 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. (iii) Based on Eq. 1, our system can assign equivalence relation instead of hierarchical relation because the threshold of the difference of radius between two classes is smaller than 0.1.

[9] Thieblin, Elodie (2019): Task-oriented complex alignments on conference organisation. figshare. Dataset.

[10] https://doi.org/10.6084/m9.figshare.4986368.v8.

4.2 Experiments on the Silex Use Case

The second data set used in this evaluation is the vocabularies gathered for the Silex use case in the CS field: we tried to match (i) ESCO (160 concepts to represent occupations) to Cigref (42 concepts), (ii) ESCO to ROME (117 concepts), (iii) NAF to kompass (574 concepts) and (iv) NAF to Silex activity domains (14 concepts). A gold standard of each matching case was provided by an expert in the Silex company. Depending on the vocabularies to be aligned, the precision value ranges between (i) 0.71 and 0.8 for the closeMatch relation, (ii) 0.7 and 0.83 for the narrowMatch relation and (iii) 0.73 and 1 for the broadMatch relation. On the other hand, the recall value ranges between (i) 0.6 and 0.95 for the closeMatch relation, (ii) 0.69 and 1 for the narrowMatch relation and (iii) 0.68 and 1 for the broadMatch relation. For example, the ROME concept "computer developer" is stated to be broader than the ESCO concept of "Applications programmers" which is in broad relation with the ESCO concept of "Usability designer", "System programmer", "System developer".

5 Conclusion

In this paper, we reported the results of a novel ontology alignment method, capable of distinguishing between equivalence and hierarchical relationships. Our first challenge was to answer on the real-world use case encountered by the Silex company. These results show that the proposed approach to ontology alignment based on a vector representation of the concepts to be matched is promising. As future work, we aim at defining a specific set of pre-trained word vectors that best covers the Silex B2B use case. We also aim at performing an empirical study to define the optimal threshold for radius difference.

References

1. Dhouib, M., Zucker, C.F., Tettamanzi, A.: Construction d'ontologie pour le domaine du sourcing. In: 29es Journées Francophones d'Ingénierie des Connaissances, IC 2018, pp. 137–144 (2018)
2. Euzenat, J., Shvaiko, P., et al.: Ontology Matching, vol. 18. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-49612-0
3. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
4. Ochieng, P., Kyanda, S.: Large-scale ontology matching: state-of-the-art analysis. ACM Comput. Surv. (CSUR) **51**(4), 75 (2018)
5. Thiéblin, É., Haemmerlé, O., Hernandez, N., Trojahn, C.: Task-oriented complex ontology alignment: two alignment evaluation sets. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 655–670. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_42
6. Zhang, Y., et al.: Ontology matching with word embeddings. In: Sun, M., Liu, Y., Zhao, J. (eds.) CCL/NLP-NABD -2014. LNCS (LNAI), vol. 8801, pp. 34–45. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12277-9_4