



# Ontology Matching Through Absolute Orientation of Embedding Spaces

Jan Portisch<sup>1,2</sup>, Guilherme Costa<sup>1</sup>, Karolin Stefani<sup>1</sup>,  
Katharina Kreplin<sup>1</sup>, Michael Hladik<sup>1</sup>, and Heiko Paulheim<sup>2</sup>

<sup>1</sup> SAP SE, Walldorf, Germany

{jan.portisch, guilherme.costa, karolin.stefani, katharina.kreplin,  
michael.hladik}@sap.com

<sup>2</sup> Data and Web Science Group, University of Mannheim, Mannheim, Germany  
{jan, heiko}@informatik.uni-mannheim.de

**Abstract.** Ontology matching is a core task when creating interoperable and linked open datasets. In this paper, we explore a novel structure-based mapping approach which is based on knowledge graph embeddings: The ontologies to be matched are embedded, and an approach known as absolute orientation is used to align the two embedding spaces. Next to the approach, the paper presents a first, preliminary evaluation using synthetic and real-world datasets. We find in experiments with synthetic data, that the approach works very well on similarly structured graphs; it handles alignment noise better than size and structural differences in the ontologies.

**Keywords:** Ontology matching · Embeddings · Absolute orientation

## 1 Introduction

Ontology matching describes the complex process of finding an alignment  $A$  between two ontologies  $O_1$  and  $O_2$ . An alignment is a set of correspondences where a correspondence is, in its simplest form, a tuple of  $\langle e_1, e_2, R \rangle$  where  $e_1 \in O_1$  is an element from one ontology,  $e_2 \in O_2$  is an element from the other ontology, and  $R$  is the relation that holds between the two elements. Typically,  $R$  is equivalence ( $\equiv$ ).

In this paper, we examine the use of embedding two ontologies for finding an alignment between them. Given two embeddings of the ontologies, we use a set of anchor points to derive a joint embedding space via a rotation operation.

## 2 Related Work

**Knowledge Graph Embeddings.** Given be a (knowledge) graph  $G = (V, E)$  where  $V$  is the set of vertices and  $E$  is the set of directed edges. Further given be a set of relations  $R$ ,  $E \subseteq VxRxV$ . A knowledge graph embedding is a projection

$E \cup R \rightarrow \mathbb{R}^d$ <sup>1</sup>. In this paper, we use the RDF2vec approach, which generates multiple random walks per vertex  $v \in V$ . An RDF2vec sentence resembles a walk through the graph starting at a specified vertex  $v$ . Those random walks are fed into a *word2vec* algorithm, which treats the entities and relations as words and the random walks as sentences, and consequently outputs numeric vectors for entities and relations.

**Absolute Orientation.** Multiple approaches exist for aligning embeddings. In this paper, the extension by Dev et al. [2] of the *absolute orientation* approach is used. The approach showed good performance on multilingual word embeddings. The calculation of the rotation matrix is based on two vector sets  $A = \{a_1, a_2, \dots, a_n\}$  and  $B = \{b_1, b_2, \dots, b_n\}$  of the same size  $n$  where  $a_i, b_i \in \mathbb{R}^d$ . In a first step, the means  $\bar{a} = \frac{1}{n} \sum_{i=1}^n a_i$  and  $\bar{b} = \frac{1}{n} \sum_{i=1}^n b_i$  are calculated. Now,  $\bar{a}$  and  $\bar{b}$  can be used to center  $A$  and  $B$ :  $\hat{A} \leftarrow (A, \bar{a})$  and  $\hat{B} \leftarrow (B, \bar{b})$ . Given the sum of the outer products  $H = \sum_{i=1}^n \hat{b}_i \hat{a}_i^T$ , the singular value decomposition of  $H$  can be calculated:  $svd(H) = [U, S, V^T]$ . The rotation is  $R = UV^T$ . Lastly,  $\hat{B}$  can be rotated as follows:  $\tilde{B} = \hat{B}R$ .

**Matching with Embeddings.** Embedding-based matching approaches have gained traction recently, mostly using embeddings of the textual information contained in ontologies [7]. OntoConnect [1], for example, uses fastText within a larger neural network to match ontologies; DOME [3] exploits doc2vec; TOM [5] and F-TOM [4] use S-BERT. With the exception of ALOD2vec Matcher [8], knowledge graph embeddings are rarely used. The work presented in this paper is different in that it does not rely on labels or an external knowledge graph. Instead, an embedding is learnt for the ontologies to be matched.

### 3 Approach

We first train two separate embedding spaces for the two ontologies to be matched (i.e.,  $O_1$  and  $O_2$ ). This is done in two independent RDF2vec training processes. In a second step, we then perform the absolute orientation operation to rotate one embedding space onto the other.

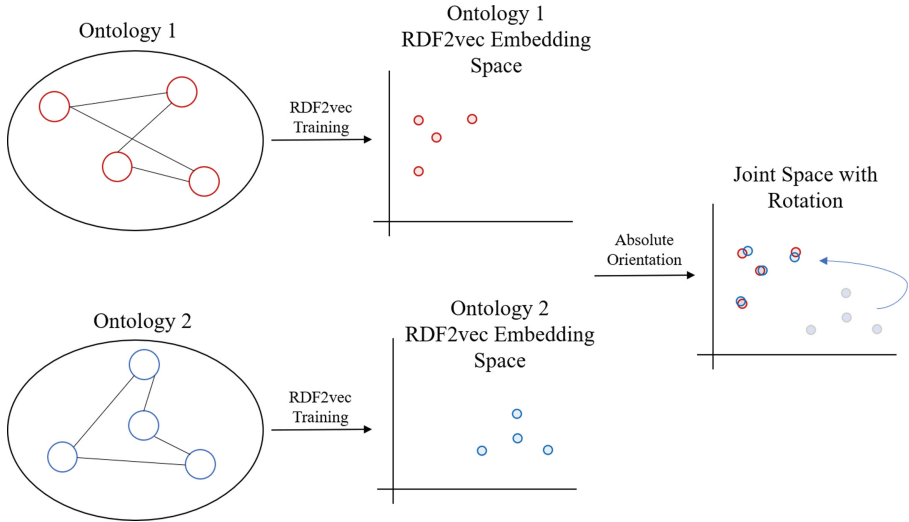
For the matching operation, we assign for each node in  $e \in O_1$  the closest node  $e \in O_2$  according to Euclidean distance (Fig. 1).

### 4 Experiments

For the experiments, jRDF2vec<sup>2</sup> [6] was used to obtain RDF2vec embeddings. We chose the following hyper parameter values: *dimension* = 100, *window* = 6,

<sup>1</sup> Variations of this formulations are possible, e.g., including different dimensions for the vector spaces of  $E$  and  $R$ , and/or using complex instead of real numbers.

<sup>2</sup> see <https://github.com/dwslab/jRDF2Vec>.



**Fig. 1.** High-level overview of the absolute orientation approach.

$depth = 6$ ,  $walks = 150$ . The code together with the complete set of figures and results is available online<sup>3</sup>.

#### 4.1 Synthetic Experiments

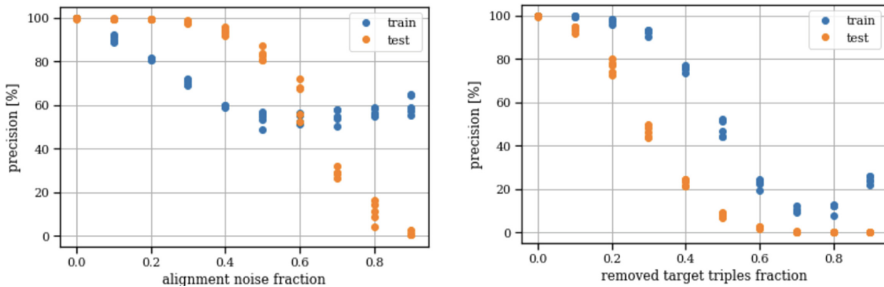
In a first step, we perform sandbox experiments on synthetic data. We generate a graph  $G$  with 2,500 nodes  $V$ . For each node  $v \in V$ , we draw a random  $d$  number using a Poisson distribution  $f(k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$  with  $\lambda = 4$ . We then randomly draw  $d$  nodes from  $V \setminus v$  and add the edge between  $v$  and the drawn node to  $G$ . We duplicate  $G$  as  $G'$  and generate an alignment  $A$  where each  $v \in V$  is mapped to its copy  $v' \in V'$ . We define the matching task such that  $G$  and  $G'$  shall be matched. The rotation is performed with a fraction  $\alpha$  from  $A$ , referred to as the anchor alignment  $A'$ . In all experiments, we vary  $\alpha$  between 0.2 and 0.8 in steps of size 0.2.

*Training Size.* In order to test the stability of the performed rotation, also referred to herein as training, we evaluate varying values for  $\alpha$ . Each experiment is repeated 5 times to account for statistical variance. The matching precision is computed for each experiment on the training dataset  $A'$  and on the testing dataset  $A \setminus A'$ . The split between the training and the testing datasets is determined by  $\alpha$ . We found that the model is able to map the entire graphs regardless of the size of the training set  $A'$  (each run achieved a precision of 100%).

<sup>3</sup> see <https://github.com/guilhermesfc/ontology-matching-absolute-orientation>.

*Alignment Noise.* In order to test the stability in terms of noise in the anchor alignment  $A'$ , we distort a share of the training correspondences by randomly matching other than the correct nodes. We vary this level of alignment noise between 0 (no noise introduced) and 0.9 (90% of the alignments are randomly matched) in steps of size 0.1. Figure 2 (left) shows the performance with  $\alpha = 0.2$ . We observe that the test performance declines with an increasing amount of noise. Interestingly, this relation is not linear. It is visible in Fig. 2 (left) that the approach can handle 40% of noise before dropping significantly in terms of test performance.

*Graph Heterogeneity.* In order to test the stability in terms of graph heterogeneity, we randomly remove triples from the target graph  $G'$  after setting up the alignment between the source graph  $G$  and the target graph  $G'$ . We vary the fraction of randomly removed triples in  $G'$  between 0 (no triples removed) and 0.9 (90% of the triples removed) in steps of size 0.1. In Fig. 2 (right) it can be observed that with a size deviation of 30%, the performance starts to drop rapidly. Comparing the two plots in the figure, it can be seen that the approach handles noise significantly better than size and structure deviations in graphs.



**Fig. 2.** The effect of distortions. (1) alignment noise (left) and (2) size differences (right). Graphs are given for  $\alpha = 0.2$ .

## 4.2 Experiments on Real Data

We also test our approach on the OAEI<sup>4</sup> multifarm dataset. Here, multilingual ontologies from the conference domain have to be matched. Since the absolute orientation approach does not use textual data, we only evaluate the German-English test case. This is sufficient because the other language combinations of the multifarm dataset use structurally identical graphs. With a sampling rate of 20%, our approach achieves micro scores of  $P = 0.376$ ,  $R = 0.347$ , and

<sup>4</sup> The Ontology Alignment Evaluation Initiative (OAEI) provides reference alignments and carries out yearly evaluation campaigns since 2004. For more information, see <http://oaei.ontologymatching.org/>.

$F_1 = 0.361$ . Compared to the systems participating in the 2021 campaign [11], the recall is on par with state of the art systems; an overall lower  $F_1$  is caused by a comparatively low precision score. While not outperforming top-notch OAEI systems in terms of  $F_1$ , the performance indicates that the approach is able to perform ontology matching and may particularly benefit from the addition of non-structure-based features.

## 5 Conclusion

In this paper, we showed early work on aligning graphs through a graph embedding algorithm combined with an absolute orientation rotation approach. In multiple experiments we showed that the approach works for structurally similar ontologies. It handles alignment noise better than varying sizes and structures of graphs. In the future, we plan to conduct experiments with different variants of embedding approaches [9, 10], as well as to combine the approach with further text-based features in a hybrid matching system.

## References

1. Chakraborty, J., Zahera, H.M., Sherif, M.A., Bansal, S.K.: ONTOCONNECT: domain-agnostic ontology alignment using graph embedding with negative sampling. In: ICMLA 2021, pp. 942–945. IEEE (2021)
2. Dev, S., Hassan, S., Phillips, J.M.: Closed form word embedding alignment. *Knowl. Inf. Syst.* **63**(3), 565–588 (2021). <https://doi.org/10.1007/s10115-020-01531-7>
3. Hertling, S., Paulheim, H.: DOME results for OAEI 2019. In: OM 2019. CEUR Workshop Proceedings, vol. 2536, pp. 123–130. CEUR-WS.org (2019)
4. Knorr, L., Portisch, J.: Fine-tom matcher results for OAEI 2021. In: OM 2021. CEUR Workshop Proceedings, vol. 3063, pp. 144–151. CEUR-WS.org (2021)
5. Kossack, D., Borg, N., Knorr, L., Portisch, J.: TOM matcher results for OAEI 2021. In: OM 2021. CEUR Workshop Proceedings, vol. 3063, pp. 193–198. CEUR-WS.org (2021)
6. Portisch, J., Hladik, M., Paulheim, H.: Rdf2vec light - a lightweight approach for knowledge graph embeddings. In: ISWC 2020 Demos and Industry Track, vol. 2721, pp. 79–84 (2020)
7. Portisch, J., Hladik, M., Paulheim, H.: Background knowledge in ontology matching: a survey (2022)
8. Portisch, J., Paulheim, H.: Alod2vec matcher results for OAEI 2021. In: OM 2021. CEUR Workshop Proceedings, vol. 3063, pp. 117–123. CEUR-WS.org (2021)
9. Portisch, J., Paulheim, H.: Putting rdf2vec in order. In: ISWC Posters and Demos, vol. 2980, pp. 1–5 (2021)
10. Portisch, J., Paulheim, H.: Walk this way! entity walks and property walks for rdf2vec. In: ESWC Posters and Demos (2022)
11. Pour, M.A.N., et al.: Results of the ontology alignment evaluation initiative 2021. In: OM 2021. CEUR Workshop Proceedings, vol. 3063, pp. 62–108. CEUR-WS.org (2021). [http://ceur-ws.org/Vol-3063/oaei21\\_paper0.pdf](http://ceur-ws.org/Vol-3063/oaei21_paper0.pdf)