Embedding OWL ontologies with OWL2Vec*

Ole Magnus Holter[1], Erik B. Myklebust[1,2], Jiaoyan Chen[3], and Ernesto Jimenez-Ruiz**[1,4]

[1] Department of Informatics, University of Oslo, Norway
[2] Norwegian Institute for Water Research, Oslo, Norway
[3] Department of Computer Science, University of Oxford, UK
[4] The Alan Turing Institute, London, UK

Abstract. In this paper, we present a preliminary study to compute embeddings for OWL 2 ontologies by projecting the ontology axioms into a graph and performing (random) walks over the ontology graph to create a corpus of sentences. This corpus is then given to a neural language model to create concept embeddings. The conducted preliminary evaluation shows promising results.

1 Introduction

In the literature we can find a number of approaches that perform embeddings over (RDF) knowledge graphs [17] to conduct knowledge graph completion (*e.g.*, [4, 11, 14]). Most of the approaches, however, only focus on the embedding of the data instances. Although some approaches also learn embeddings for concepts involved in instance type definitions (*e.g.*, [11, 8]), the embeddings rely on data instances and the knowledge provided by the ontology (*e.g.*, subsumption axioms) is typically ignored. Alshahrani et al. [2] performs reasoning to expand the knowledge graph with new facts (*e.g.*, types of the instances), but the main focus is on the instance embeddings for biological link prediction.

Regarding concept embeddings, there have been some efforts to leverage word embeddings to associate a vector to the lexical information of the ontology concepts. This approach has typically been applied to ontology alignment tasks (*e.g.*, [13, 9]). The main limitation of this approach is the dependence on a relevant text corpus or a pretrained set of word embeddings, which may have some limitations when applying to ontologies with domain-specific vocabulary. Some works refine the word embeddings using semantic lexicons (*e.g.*, [9]) to compensate for the lack of domain-specific training corpora. Nevertheless, the computed word embeddings neglect the rich semantics of the ontologies (*e.g.*, concept hierarchy, relationships among concepts).

The approach followed by the systems Onto2Vec [15] and OPA2Vec [16] deserves special mention. Both Onto2Vec and OPA2Vec consider each axiom in the ontology as a sentence. The set of axioms (including some inferred axioms) in the ontology form a document that is then given to Word2Vec [10]. Word2Vec computes vectors for each of the elements in the document including concept identifiers, relationships and OWL

** Corresponding author: ernesto.jimenez.ruiz@gmail.com

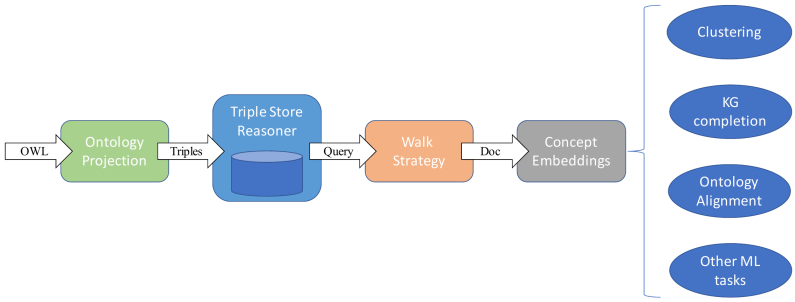


Fig. 1: Current pipeline in OWL2Vec

constructs. Although this approach represents an interesting effort, it has the following limitations: *(i)* the corpus of sentences may be limited for small-medium ontologies to create meaningful vectors, *(ii)* OWL constructs may introduce noise in the embeddings, and *(iii)* Word2Vec does not differentiate between sentences like "A SubClassOf: B" and "A DisjointWith: C" which will lead to similar embeddings for A, B and C.

In order to overcome the limitations of state-of-the-art approaches, we have implemented a framework to compute semantic embeddings from OWL 2 ontologies. Our approach *(i)* projects the ontology into a graph, *(ii)* implements several strategies to walk the ontology graph, *(iii)* creates a corpus of sentences according to the walking strategies, and *(iv)* generates concept embeddings from that corpus.

2 Methods

Figure 1 summarises the current architecture of the OWL2Vec framework, composed of three main components: ontology projection, walk strategy, and concept embeddings.

Ontology projection. We follow a simplified version of the (RDF-based) graph projection of the ontology used by Agibetov et al. [1]. The nodes in the projected RDF graph represent concepts in the ontology while edges are labelled with possible relations among those concepts. The key property of this projection is that every edge (*i.e.*, triple $\langle A, R_o, B \rangle$) in the graph is justified by one or more axioms entailed by the ontology which "semantically relates" two concepts (*e.g.*, A and B) via a property (*e.g.*, R_o). Table 1 shows the type of axioms currently considered in the ontology projection.

Walk strategy. We have implemented a set of strategies to walk the ontology graph. We initially relied on a modified version of RDF2Vec [14]. The main difference with respect to the original RDF2Vec algorithm is the use of the ontology projection as input and the inclusion of weighted edges for the walks (as also proposed in [5]). One could give more weight to the taxonomic relationships or to the object properties to walk from one hierarchy branch to another. The modified algorithm also allowed the creation of sentences with the concept URI and/or the concept labels. We encountered, however, a scalability limitation for long walks over large ontologies. To overcome the limitations of the RDF2Vec approach, we implemented a more flexible strategy inspired by node2vec [6]. This strategy *(i)* scales with large ontologies, *(ii)* allows to bias the walks, *(iii)* enables semantic similarity not only for closely connected elements but also for similar structures, *(iv)* has flexibility to change the direction of a walk to avoid

Condition 1	Condition 2	Triple(s)
$A\ SubClassOf$: $R_o\ Restriction\ D$ $R_o\ Restriction\ D\ SubClassOf$: A	$D \equiv B \mid B_1 \sqcup ... \sqcup B_n \mid B_1 \sqcap ... \sqcap B_n$	$\langle A, R_o, B \rangle$ or $\langle A, R_o, B_i \rangle$ for i∈1..n
$R_o\ Domain$: A $A\ SubClassOf$: $R_o\ value\ b$ $R_o\ InverseOf$: R_o^- $S_1 \circ ... \circ S_n\ SubPropertyOf$: R_o	$R_o\ Range$: B $b\ type\ B$ $\langle A, R_o^-, B \rangle$ in graph $\langle A, S_1, C_1 \rangle ... \langle C_n, S_n, B \rangle$ in graph	
$B\ SubClassOf$: A		$\langle B, SubClassOf, A \rangle$ $\langle A, SubClassOf^-, B \rangle$

Table 1: Projection of the OWL 2 ontology axioms into an RDF graph. *Restriction* is one of: *min*, *max*, *exactly*, *some*, *only*. A, B, B_i and C_i are atomic concepts, S_i, R_o and R_o^- are object properties, b an individual.

dead ends, and *(v)* uses the (in-memory) triple store reasoner RDFox [12] to enhance the access to the projected ontology graph.

Concept embeddings. The walk strategies in OWL2Vec are flexible and allow the creation of different types of corpora of sentences that will lead to concept embeddings with different characteristics. For example, the computed embeddings may favour the semantic similarity among concepts within the same hierarchy (*e.g.*, between $Person$ and $Researcher$) or among concepts related with other properties (*e.g.*, between $Paper$ and $Researcher$). We currently rely on Word2Vec [10] and FastText [7] to compute the embedding from the resulting documents.

3 Preliminary Evaluation and Future Work

Figure 2 shows a subset of our preliminary set of experiments.[5] We have computed (agglomerative) clusters of the concepts in the EKAW conference ontology based on the embeddings provided by RDF2Vec, Onto2Vec and OWL2Vec. We can observe that the clusters (of related concepts) obtained with the OWL2Vec embeddings are well differentiated while for RDF2Vec and Onto2Vec the cloud of points is more sparse.[6]

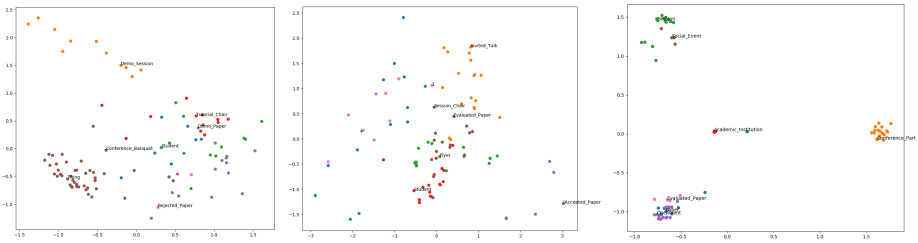
These clustering results are encouraging, but more evaluation is required to evaluate the usefulness of the computed embeddings. We plan to conduct an extensive evaluation to obtain quality measures similar to the ones proposed within the Concept2vec framework [3]. We also aim at evaluating OWL2Vec in real-world applications like biomedical link prediction or ecotoxicological effect prediction to analyze if the OWL2Vec concept embeddings improve the state-of-the-art solutions. Furthermore, we are adopting OWL2Vec within our ontology alignment system as the different OWL2Vec walking strategies has led to a promising set of concept similarities.

OWL2Vec has the potential of becoming an essential component of machine learning applications that rely on the semantic information of an ontology as input.

Acknowledgements. This work is supported by the AIDA project (The Turing Institute) and the SIRIUS Centre for Scalable Data Access (RCN 237889).

[5] OWL2Vec source codes available from: https://gitlab.com/oholter/owl2vec

[6] The rest of conference-based OntoFarm ontologies [18] led to similar findings.



(a) RDF2Vec (b) Onto2Vec (c) OWL2Vec

Fig. 2: Clustering of the EKAW conference ontology.

References

1. Agibetov, A., et al.: Supporting shared hypothesis testing in the biomedical domain. J. Biomedical Semantics **9**(1), 9:1–9:22 (2018)
2. Alshahrani, M., et al.: Neuro-symbolic representation learning on biological knowledge graphs. Bioinformatics **33**(17), 2723–2730 (2017)
3. Alshargi, F., Shekarpour, S., Soru, T., Sheth, A.P., Quasthoff, U.: Concept2vec: Metrics for Evaluating Quality of Embeddings for Ontological Concepts. CoRR **abs/1803.04488** (2018)
4. Bordes, A., et al.: Translating Embeddings for Modeling Multi-relational Data. In: 27th Conference on Neural Information Processing Systems (NIPS) (2013)
5. Cochez, M., Ristoski, P., Ponzetto, S.P., Paulheim, H.: Global RDF vector space embeddings. In: International Semantic Web Conference (ISWC). pp. 190–207 (2017)
6. Grover, A., Leskovec, J.: node2vec: Scalable Feature Learning for Networks. In: SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 855–864 (2016)
7. Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: FastText.zip: Compressing text classification models. CoRR **abs/1612.03651** (2016)
8. Kejriwal, M., Szekely, P.: Scalable Generation of Type Embeddings Using the ABox. OJSW **4**(1), 20–34 (2017)
9. Kolyvakis, P., et al.: DeepAlignment: Unsupervised Ontology Matching with Refined Word Vectors. In: NAACL Conference on Human Language Technologies (2018)
10. Mikolov, T., et al.: Distributed Representations of Words and Phrases and their Compositionality. In: 27th Conference on Neural Information Processing Systems (NIPS) (2013)
11. Moon, C., Jones, P., Samatova, N.F.: Learning entity type embeddings for knowledge graph completion. In: Conference on Information and Knowledge Management (CIKM) (2017)
12. Nenov, Y., Piro, R., Motik, B., Horrocks, I., Wu, Z., Banerjee, J.: RDFox: A Highly-Scalable RDF Store. In: 14th International Semantic Web Conference (2015)
13. Nkisi-Orji, I., Wiratunga, N., Massie, S., Hui, K., Heaven, R.: Ontology Alignment Based on Word Embedding and Random Forest Classification. In: ECML-PKDD. pp. 557–572 (2018)
14. Ristoski, P., Rosati, J., Noia, T.D., Leone, R.D., Paulheim, H.: RDF2Vec: RDF graph embeddings and their applications. Semantic Web **10**(4), 721–752 (2019)
15. Smaili, F.Z., Gao, X., Hoehndorf, R.: Onto2Vec: joint vector-based representation of biological entities and their ontology-based annotations. Bioinformatics **34**(13), i52–i60 (2018)
16. Smaili, F.Z., Gao, X., Hoehndorf, R.: OPA2Vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. Bioinformatics **35**(12) (2019)
17. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
18. Zamazal, O., Svátek, V.: The Ten-Year OntoFarm and its Fertilization within the Onto-Sphere. J. Web Semant. **43**, 46–53 (2017)