Graph-Convolution-Based Classification for Ontology Alignment Change Prediction

Matthias Jurisch, Bodo Igler

RheinMain University of Applied Sciences
Department of Design – Computer Science – Media
Unter den Eichen 5
65195 Wiesbaden, Germany
matthias.jurisch@hs-rm.de, bodo.igler@hs-rm.de

Abstract. Finding alignments between ontologies is a challenging and time-consuming task. When the aligned ontologies change, these alignments need to be changed as well. A recent approach to this problem proposes using embeddings as a representation for classifying changes. In this work, we compare embedding-based approaches to a neural network architecture built for node classification in knowledge graphs, namely relational graph convolutional networks. In our evaluation on two datasets from the biomedical domain, the best-performing embedding-based methods are RDF2Vec and TransE. The Graph convolution approach achieves similar results as the best-performing embedding based methods on a smaller dataset but outperforms all other approaches in standard classification metrics on a bigger dataset.

Keywords: Ontology Alignment · Alignment Adaption · Graph Embedding · Graph Neural Network.

1 Introduction

Ontologies that cover overlapping topics are often connected by so-called ontology alignments, that describe the relation of concepts in different ontologies. Finding these alignments is challenging and requires some degree of manual work, which can be supported by approaches from the area of ontology matching. Ontology matching has been an important area of semantic web research for years [16]. However, finding these alignments is only a part of the puzzle. As ontologies should change with the knowledge they represent, not only the ontologies need to be adapted, but the alignments as well. As with finding alignments, adapting them is often done manually and is very time-consuming. This is especially the case in the area of biomedical ontologies, given the changes required in this area as well as the size of the ontologies. Hence, automation of this task is desirable.

Rule-based approaches like [7] and [13] can be used to automate this task. These methods are based on a set of hand-crafted rules, that need to be adapted and maintained as ontology evolution itself may change over time. To automatically classify changes, we proposed a learning based method that first learns

graph embeddings as a change representation and then applies established classification approaches [10].

In this work, we examine the application of relational graph convolutional networks [15] to the same problem. This approach can be applied directly to the graph and does not require a separate pre-training to generate a meaningful graph representation. Also, we compare its performance regarding established classification methods to approaches with intermediate representation learning. To achieve a fair comparison, we evaluate several embedding-based methods on an established dataset for mapping adaption and compare the results to results obtained by applying a graph convolution.

The remainder of this paper is structured as follows: Section 2 presents foundations and related work and identifies a research gap. Our approach is discussed in Section 3. In Section 4, an evaluation of different classification approaches is presented. Results of this evaluation are discussed in Section 5. Section 6 closes this paper with a conclusion and an outlook.

2 Foundations and Related Work

Ontology Alignments (sometimes referred to as Ontology Mappings) are used to connect concepts in different ontologies. When these ontologies change, an adaption of these alignments is usually done manually. The problem of adapting these alignments is referred to as the mapping adaption problem [7]. Work on this topic is usually divided into the area of compositional mapping adaption and incremental adaption [4]. To create a new alignment $A'_{\mathcal{O}'_1,\mathcal{O}'_2}$ when the ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ evolve to $\mathcal{O}'_1$ and $\mathcal{O}'_2$, the compositional approach creates a *composition of the alignment* $A_{\mathcal{O}_1,\mathcal{O}_2}$ and $A^{+}_{\mathcal{O}_2,\mathcal{O}'_2}$, the alignment between $\mathcal{O}_2$ and $\mathcal{O}'_2$. The incremental approach applies a *set of rules* that determine how alignments should be adapted to ontology changes. While these approaches stem from the area of database and XML schema adaption [18,21], works by Gross [6] and Dos Reis [5] have shown that both ideas can be applied to ontologies.

The mentioned approaches either rely on ontology mapping approaches between versions or a set of hand-crafted rules. Therefore, adaptions are either dependent on the quality of automatic ontology matching techniques or rely on manual work by an expert for creating rules. To automate incremental approaches using machine learning, a representation of nodes in the ontologies is required. An approach with manual feature engineering for predicting changes in ontologies in general was proposed by [2]. Features considered included background information from other ontologies on the same topic and records from publication databases as well as simple structural features such as the number of siblings and sub- or superclasses. [10] proposed using graph embeddings, specifically RDF2Vec-Embeddings [14] as a node representation, which provides a more detailed representation of the graph structure. This representation is then used to train established classification approaches for the mapping adaption problem. However, only one kind of graph embedding, namely RDF2Vec, is regarded.

Over the last ten years, several approaches for embedding knowledge graphs have been proposed. Some of these approaches are inspired by language modeling techniques such as Word2Vec [11]. The aforementioned RDF2Vec [14] and an approach based on global embeddings [3] are methods from this category. Another category of techniques is based on knowledge base completion approaches, where entity and relation embeddings are learned to provide some kind of scoring function to predict whether a triple should be part of the graph. The scoring function is mostly based on some kind of translation [1,19] or multiplication [20,17]. Embeddings from both categories can theoretically be useful when predicting alignment changes, however only one approach from the first category has been evaluated in [10].

The aforementioned idea focuses on first creating embeddings and then learning the actual task at hand, namely, predicting which part of an ontology alignment should change as a consequence of an ontology change. [15] has presented a graph network architecture that can be used for end-to-end learning on RDF graphs. However, this kind of approach has not yet been evaluated for predicting ontology alignment changes.

To our knowledge, the state of the art on ontology mapping adaption lacks an evaluation of graph network approaches and a comparison of different knowledge base embedding methods as a foundation for change classification tasks. This aspect is at the core of the research presented in this paper.

3 Approach

The classification task we try to solve in this work is the same as in [10]: For each changed entity c that is near an alignment statement, we predict whether an alignment statement near c needs to be changed. To do so, we train a classifier on data extracted from a version history. The classes we extract represent whether in a given version change, for a changed entity c, alignment statements in the neighbourhood of c have also been changed. For the classification itself we use two approaches in several variants: (1) the *approach with intermediate representation learning*, where we train a model for embedding all entities in the knowledge graph and subsequently apply established classification techniques and (2) the *graph-network-based approach*, where we use a relational graph convolutional network [15] for end-to-end classification.

For the approach with intermediate representation learning, an embedding is learned by creating a single graph out of both ontologies and the mapping and applying a knowledge graph embedding approach to this graph. As embedding approaches we compare the established knowledge base completion approaches TransE [1], TransH [19], Distmult [20] and Complex [17] and RDF2Vec [14] from the area of language-modeling based approaches. These approaches have been chosen because of their easy accessibility in the OpenKE framework and their general popularity. The classification approaches we apply on top of these embeddings are from the area of Regression, Naive Bayes, Tree-Based Algorithms as well as Support Vector Machines and Multilayer Perceptrons.

For the graph-network-based approach we apply relational graph convolutional networks (RGCNs) [15]. RGCNs are an extension of graph convolutional networks [**?**] to relational graphs. The core idea of RGCNs is that the propagation between RGCN-layers is based on the direct connections of a node in the relational graph. At each layer of the network, each neuron represents the activation at a graph node. In detail, the propagation function for a forward pass is

$$h_i^{(l+1)} = \sigma(\sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)})$$

where $h_i^{(l)}$ is the activation in node i at layer l of the neural network, R is the set of all relations, N_i^r is the set of all nodes connected to i by relation r, σ is an activation function, $c_{i,r}$ is a normalization constant and W_r^l is a weight matrix in layer l for relation r. Hence, neural network activation travels through the graph. The weight matrices determine what kind of relations at each layer transport what kind of information. The depth of the network determines how many steps in the graph the activation is propagated. To create the sets R and N_i^r for all relations, we use a graph constructed from both ontologies and the alignment prior to the changes we want to classify.

4 Evaluation

The main research question of our evaluation is how a graph-convolution-based approach compares to a two-step approach with separated representation learning and classification. To evaluate this question, the impact of the graph embedding choice in relation to the performance of the approach when separating classification and representation learning also needs to be examined. To address these issues, we conducted a series of experiments that are described in the following subsections.

4.1 Dataset

The dataset we use has been extracted from biomedical ontologies by [9] and has been made publically available by the authors of [6] on the web[1]. This dataset consist of three biomedical ontologies – SNOMED, FMA and the NCI Thesaurus, with version from 2009-2012 and mappings between all ontologies for each of those versions.

These ontology versions are used as a silver standard, since ontology and mapping versions are not necessarily perfect but contain errors. For each changed entity close to an alignment statement in a new version of the ontology, we examine, if an alignment statement has been changed. If this is the case, we

[1] https://dbs.uni-leipzig.de/de/research/projects/evolution_of_ontologies_and_mappings/ontology_mapping_adaption

assign the changed entity the class C_{change}, else, we assign the class $C_{nochange}$. Table 1 gives an overview of the datasets we use. For each pair of ontologies, we use two sets of changes: one set consisting of changes from 2009-2011, which is always used as the training set, and one set of changes from 2011-2012, which is used as the test set. In general, C_{change} is always smaller than $C_{nochange}$. This effect is present to a larger extent in SNOMED-FMA than in FMA-NCI.

Table 1. Datasets

	Version	*Entities*	*Triples*	$\#C_{change}$	$\#C_{nochange}$
FMA-NCI	2009-2011	2M	7M	725	984
	2011-2012			352	421
SNOMED-FMA	2009-2011	4M	15M	1526	13435
	2011-2012			177	6925

4.2 Approach with Intermediate Representation Learning

For comparing different embedding methods as a part of the representation learning process, we first train embeddings and then compare the performance of classifiers that use these embeddings as features on our dataset. To train embeddings from the knowledge graph completion area, we use the OpenKE-Framework [8]. To train RDF2Vec-embeddings, we use the implementation[2] provided by the authors of [14]. Hyperparameters where chosen based on recommendations in the documentation.

The classification itself was implemented using scikit-learn[12]. We used classifiers from several areas, including classic regression, naive bayes and nearest neighbour approaches as well as tree-based algorithms, SVMs and a feed-forward neural network. A list of all classifiers used is shown in Table 2. In order to find the optimal embedding-classifier-combination all possible combinations were evaluated, yielding a total of 40 combinations.

As evaluation metrics we use standard classification metrics, namely precision, recall, f1-measure, accuracy, roc-auc score and average precision. Precision, recall and f1-score are measured regarding the class C_{change}, since classification performance regarding this aspect is the most important for this task. We evaluated each classification method on each embedding on both datasets, FMA-NCI and SNOMED-FMA. The only exception is that we did not use RDF2Vec-embeddings on SNOMED-FMA. RDF2Vec uses a two-step approach: first, random walks through the graph are created (typically around 200 random walks per entity). With the SNOMED-FMA graph containing 4 million entities and more than 15 million triples, creating random walks for all entities would have taken too long to be feasible given the hardware we had available. The main issue here was the size and number of random walks.

[2] http://data.dws.informatik.uni-mannheim.de/rdf2vec/code/

Table 2. Classifiers

Category	Method
Regression	Logistic Regression (LR)
Naive Bayes	Gaussian Naive Bayes (NB)
Nearest Neighbour	KNN
Tree-Based Algorithms	CART, Random Forest (RF)
Support Vector Machines	RBF-Kernel, Linear Kernel
Multilayer Perceptron	MLP

4.3 Graph Network Based Approach

For the graph network based approach, we used the RGCN implementation written by the authors of [15] that is available on GitHub[3]. For trainining

the model we used 5 hidden layers, a l2 penalty of 0.005, a dropout rate of 0.05 and a learning rate of 0.01 with 50 training epochs. We choose only 5 hidden layers, since otherwise the model would consume too much memory on the SNOMED-FMA dataset. The other parameters were determined by a grid search over the hyperparameters. This approach was evaluated on the same dataset with the same metrics as the embedding-based approach.

4.4 Results

Results of this evaluation procedure for the dataset FMA-NCI are shown in Table 3. Underlined entries represent the best values for each metric. For readability purposes, we only show the best and the second-best results for every metric of the combinations of embedding methods and classification approaches. For each embedding method, we also show the two best combinations of classification method and embedding regarding f-measure. The graph embedding is nondeterministic. To account for graph embedding stability, we repeated the embedding step and evaluation. Since the differences between different runs of the embedding models were insignificant, we only report results of the first experiment.

Of the embedding-based methods, RDF2Vec with Naive Bayes achieves the best performance comparing all metrics except precision. At least one of the other classifier/embedding combination achieves a similar performance given one specific metric. The RGCN approach achieves very similar results to the RDF2Vec-based classifier.

Results for SNOMED-FMA are shown in Table 4. When observing the metrics for this dataset, it is important to reconsider the distribution of classes in the test set: only 2.6% of changes in the test set are in class C_{change}. As already mentioned, RDF2Vec is missing from this evaluation, as creating the embeddings would have taken to much time on this dataset. On this dataset, no embedding based method is better than the other methods in nearly all metrics. The RGCN approach clearly outperforms the embedding-based methods on this

[3] https://github.com/tkipf/relational-gcn

Table 3. FMA-NCI Results

embedding	classifier	f1	acc	prec	rec	roc_auc	avg. prec
TransE	KNN	0.587	0.659	0.642	0.541	0.648	0.553
	RF	0.551	0.659	0.672	0.467	0.641	0.553
	SVM(RBF)	0.265	0.602	0.771	0.160	0.561	0.500
	MLP	0.550	0.683	0.756	0.432	0.659	0.582
TransH	MLP	0.555	0.655	0.659	0.479	0.638	0.549
	RF	0.567	0.657	0.655	0.500	0.643	0.552
DistMult	RF	0.516	0.639	0.647	0.429	0.619	0.534
	MLP	0.560	0.656	0.657	0.488	0.640	0.551
Complex	MLP	0.572	0.611	0.565	0.580	0.608	0.516
	LR	0.398	0.542	0.485	0.337	0.523	0.461
RDF2Vec	NB	0.657	0.548	0.701	0.619	0.501	0.701
	LR	0.735	0.647	0.774	0.700	0.611	0.752
RGCN		0.6247	0.778	0.706	0.561	0.723	0.540

dataset. Applying RGCNs results in significantly higher values in nearly all metrics except for accuracy, where the combination of TransE and linear regression obtains similar results, which is not a remarkable score given the distribution of classes in the test set.

To adapt our approach to the unbalanced dataset, we conducted two further experiments: We repeated our experiments on the SNOMED/FMA-Dataset using (1) oversampling and (2) undersampling of training data so that the training set was balanced in both experiments. Table 5 shows an excerpt of the results. We only show the results with highest f-measure for each embedding method out of both experiments. All best results of embedding-based methods stem from the experiment with oversampling, whereas the best RGCN-result stems from the undersampling experiment. Using oversampling, the RGCN overfitted despite regularization and dropout. While the f-measure values look similar to the best results without oversampling or undersampling, as expected, recall is higher, whereas precision is lower.

5 Discussion

When comparing the approaches that combine embeddings with traditional classification methods, we can observe that RDF2Vec in combination with Naive Bayes seems to show the best results where the computational effort allows it. However, since generating random walks is very costly, using this approach is not possible for large knowledge graphs. Another embedding method that seems

Table 4. SNOMED-FMA Results

embedding	classifier	f1	acc	prec	rec	roc_auc	avg. prec
TransE	LR	0.070	0.974	0.318	0.040	0.519	0.037
	NB	0.155	0.857	0.091	0.525	0.695	0.060
	KNN	0.155	0.965	0.192	0.130	0.558	0.047
	RF	0.189	0.944	0.148	0.260	0.611	0.057
TransH	MLP	0.248	0.966	0.276	0.226	0.605	0.082
	RF	0.191	0.944	0.149	0.266	0.613	0.058
Distmult	MLP	0.193	0.943	0.150	0.271	0.616	0.059
	RF	0.221	0.943	0.168	0.322	0.641	0.071
Complex	RF	0.243	0.944	0.183	0.362	0.660	0.082
	MLP	0.221	0.937	0.160	0.356	0.654	0.073
RGCN		0.542	0.978	0.508	0.581	0.784	0.305

Table 5. SNOMED-FMA Over/Undersampling

embedding	classifier	f1	acc	prec	rec	roc_auc	avg. prec
TransE	RF	0.222	0.896	0.137	0.598	0.751	0.091
TransH	CART	0.224	0.896	0.138	0.605	0.754	0.093
Distmult	MLP	0.223	0.895	0.136	0.605	0.753	0.092
Complex	MLP	0.222	0.895	0.136	0.599	0.751	0.091
RGCN		0.461	0.965	0.355	0.656	0.814	0.241

very promising is TransE. TransE-based classification is present in all top two performers for every metric. Hence, the choice of embedding makes a difference regarding classification performance. RDF2Vec and TransE as representations perform best on the presented datasets.

Given the performance of the RGCN approach on both datasets we can see that RGCN can achieve similar or better results than a combination of embedding and classification approaches. On the first dataset, its performance was similar to the best combined approach, on the second dataset it was significantly better. Another advantage of this end-to-end learning approach is that it is significantly faster than first training the embeddings, especially for large databases. Training embeddings using OpenKE-Embeddings or RDF2Vec at this scale was slower then the complete RGCN classification by a factor of 20-50. On the larger dataset, embedding the graph using RDF2Vec even took to long to be actually usable. Therefore the answer to the research question is that a graph-network-based approach can achieve similar or superior performance than a separated representation learning and classification approach.

6 Conclusion and Outlook

In this paper, we presented two approaches to predicting, whether alignment statements need to change after an ontology update: a two-step approach that consists of representation learning as a first step and established classification methods as a second step and an end-to-end approach that uses a neural network architecture specialized on node classification in relational graphs. In our evaluation, we could show that on the dataset we used, the best-performing representation learning approaches where RDF2Vec and TransE. The end-to-end learning approach was able to achieve similar results on one dataset and outperform the other approaches on a second, much larger dataset while in both cases needing much less computing time.

As future work, the integration of node and change features seems promising, since the current approach only uses the graph structure to reason about possible changes. Naturally, this can not be sufficient information to determine everything about the nature of a change and how to react to it. To evaluate the capabilities of the presented approach in other domains besides biomedical ontologies, another dataset is required. To our knowledge, the dataset used in this paper is the only dataset currently available for ontology mapping adaption. Hence, a new dataset needs to be built that contains knowledge from a different domain.

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)
2. Cardoso, S.D., Pruski, C., Silveira, M.D.: Supporting biomedical ontology evolution by identifying outdated concepts and the required type of change. Journal of Biomedical Informatics **87**, 1 – 11 (2018). https://doi.org/10.1016/j.jbi.2018.08.013
3. Cochez, M., Ristoski, P., Ponzetto, S.P., Paulheim, H.: Global RDF vector space embeddings. In: International Semantic Web Conference. pp. 190–207. Springer (2017). https://doi.org/10.1007/978-3-319-68288-4_12
4. dos Reis, J.C., Pruski, C., Reynaud-Delaître, C.: State-of-the-art on mapping maintenance and challenges towards a fully automatic approach. Expert Systems with Applications **42**(3), 1465–1478 (2015). https://doi.org/10.1016/j.eswa.2014.08.047
5. dos Reis, J.C., Pruski, C., Silveira, M.D., Reynaud-Delaître, C.: Dykosmap: A framework for mapping adaptation between biomedical knowledge organization systems. Journal of Biomedical Informatics **55**, 153 – 173 (2015). https://doi.org/10.1016/j.jbi.2015.04.001
6. Groß, A., dos Reis, J.C., Hartung, M., Pruski, C., Rahm, E.: Semi-automatic adaptation of mappings between life science ontologies. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **7970 LNBI**, 90–104 (2013). https://doi.org/10.1007/978-3-642-39437-9_8
7. Groß, A., Pruski, C., Rahm, E.: Evolution of Biomedical Ontologies and Mappings: Overview of Recent Approaches. Computational and Structural Biotechnology Journal pp. 1–8 (2016). https://doi.org/10.1016/j.csbj.2016.08.002

8. Han, X., Cao, S., Xin, L., Lin, Y., Liu, Z., Sun, M., Li, J.: OpenKE: An open toolkit for knowledge embedding. In: Proceedings of EMNLP (2018)
9. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I., Berlanga, R.: Logic-based assessment of the compatibility of UMLS ontology sources. In: JOURNAL OF BIOMEDICAL SEMANTICS (2010). https://doi.org/10.1186/2041-1480-2-S1-S2
10. Jurisch, M., Igler, B.: RDF2Vec-based classification of ontology alignment changes. In: Proceedings of the First Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies (DL4KGS) co-located with the 15th Extended Semantic Web Conerence (ESWC 2018) (2018)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013), http://arxiv.org/abs/1301.3781
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
13. dos Reis, J.C., Pruski, C., Da Silveira, M., Reynaud-Delaître, C.: The DyKOSMap approach for analyzing and supporting the mapping maintenance problem in biomedical knowledge organization systems. In: Simperl, E., Norton, B., Mladenic, D., Della Valle, E., Fundulaki, I., Passant, A., Troncy, R. (eds.) The Semantic Web: ESWC 2012 Satellite Events. pp. 163–175. Springer Berlin Heidelberg, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46641-4_12
14. Ristoski, P., Paulheim, H.: RDF2Vec: RDF graph embeddings for data mining. In: The Semantic Web - ISWC 20162016. pp. 498–514 (2016). https://doi.org/10.1007/978-3-319-46523-4_30
15. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., Navigli, R., Vidal, M.E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M. (eds.) The Semantic Web. pp. 593–607. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
16. Shvaiko, P., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering **25**(10), 158–176 (2013). https://doi.org/10.1109/TKDE.2011.253
17. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on Machine Learning - Volume 48. pp. 2071–2080. ICML'16, JMLR.org (2016)
18. Velegrakis, Y., Miller, R.J., Popa, L.: Mapping adaptation under evolving schemas. VLDB '03 Proceedings of the 29th international conference on Very large data bases - Volume 29 pp. 584–595 (2003). https://doi.org/10.1016/B978-012722442-8/50058-6
19. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI conference on artificial intelligence (2014)
20. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. CoRR **abs/1412.6575** (2014), http://arxiv.org/abs/1412.6575
21. Yu, C., Popa, L.: Semantic Adaptation of Schema Mappings when Schemas Evolve. Very Large Data Bases pp. 1006 – 1017 (2005). https://doi.org/10.1.1.72.2410