Conditional Constraints for Knowledge Graph Embeddings

Michael Weyns[1][0000−0002−6157−5997], Pieter Bonte[1][0000−0002−8931−8343], Bram Steenwinckel[1][0000−0002−3488−2334], Filip De Turck[1][0000−0003−4824−1199], and Femke Ongenae[1][0000−0003−2529−5477]

IDLab, Ghent University – imec
{firstname.lastname@ugent.be}

Abstract. To train knowledge graph embedding models, negative training examples have to be generated artificially. A few attempts were already made in the past to improve negative sampling by exploiting schematic domain and range constraints. However, such approaches did not exploit the full range of schematic axioms. To remedy this situation, we introduce a new type of *conditional* constraints based on OWL restrictions. Additionally, we investigate two alternative views on how constraints might be used to generate false triples—an open world view and a closed world view—and perform empirical evaluations for both of them. Our results indicate that in the closed world view, even a limited number of OWL constraints can improve link prediction performance. Also, for regular constraints, both open world and closed world views offer significant improvements with respect to baseline negative sampling techniques.

Keywords: Knowledge graph embeddings · Negative sampling · Type constraints.

1 Introduction

With the advent of Big Data, dealing with data heterogeneity has become a pressing issue. By linking data with its contextual characteristics, knowledge graphs (KGs) are able to consolidate multiple sources with diverse schemas, making them queryable through a uniform interface. When data is made available in this fashion, it becomes open to consumption by all manner of intelligent agents. The Semantic Web (SW), which includes most of the larger, publicly available KGs, such as DBpedia, YAGO, and WordNet in its Linked Open Data (LOD) cloud[1], was developed with precisely this vision of intelligent, automatic consumption [4]. Unfortunately, due to the prevailing characteristics of the underlying data, these large KGs often exhibit high levels of sparsity and noise

[1] https://www.lod-cloud.net/

[16]. To mitigate the sparsity of KGs, we can rely on various graph completion techniques.

Two primary alternatives exist to decrease the amount of missing information inside KGs. On the one hand, logic-based reasoners (inference engines) can be used to infer new facts inside the KG via deductive entailment [9]. On the other hand, statistical relational learning (SRL) techniques are able to make inductive predictions about the existential likelihood of unknown facts [10]. In fact these two approaches to graph completion are not so much alternatives as they are complementary. Deductive inference can be applied prior to statistical learning in order to increase the latter's effectiveness.

Concerning statistical approaches, the family of latent feature models assumes that the existential values of all potential facts inside the KG are conditionally independent given certain global latent features. This class of models has also been called the class of KG embedding techniques [14]. Because these techniques omit having to model local interactions and already presuppose a dependency structure, in contrast to other SRL approaches they are able to bypass dependency structure learning entirely. This boon has made research into KG embeddings especially attractive.

When KG embeddings are used for graph completion this is called link prediction. To perform link prediction, embedding models require a sense of falseness. Most KGs adhere to the SW's open world assumption (OWA), which states that any uncertainty with respect to the truth of a fact does not imply its falsity. Per this assumption, except where logical contradiction is concerned, or negation is explicitly invoked, conclusively negative facts do not exist. Yet, to gain insight into the boundary between fact and fiction, embedding models need to be supplied with plausible counterfactuals. The process by which these are generated is called *negative sampling*.

Many negative sampling strategies have built on the standard approach involving the random corruption of positive examples. A few attempts were even made to exploit the constraints inside the KG to minimise the number of useless negatives. However, these attempts focused on the relation-specific RDFS domain and range constraints. Our own work builds on this by exploring the possibility of exploiting conditional constraints based on OWL restrictions. Specifically, our contributions are the following:

- We provide closed-world interpretations of open world OWL restrictions and integrate these into a negative sampling scheme to improve link prediction performance.
- We contrast an open world interpretation of constraint-based negative sampling with a closed world interpretation and suggest a way to modulate between them.
- We evaluate our proposed enhancements on two datasets (AIFB and MUTAG) that come supplied with elaborate schemas.

2 Related Work

The most basic form of negative sampling takes a hard-line stance on the closed world assumption (CWA): All triples not observed to be true are false. Because the KG is incomplete, such an assumption is always necessarily incorrect—and therefore ineffective. Better alternatives are to perturb existing triples (by replacing either the head or the tail with another entity) or to assume a locally closed world in which any valid triple entails a whole set of false triples with the same subject and relationship but with different objects [10]. Importantly, the latter option is only valid for functional relationships. These alternatives are preferable to the basic CWA because they only generate negative triples that are more likely to be actually false.

Various extensions have been proposed to improve on this basic perturbation scheme. The most basic of these was suggested by Bordes et al. when they introduced their TransE embedding model: using so-called *filtered* negative samples [1]. Filtered negative samples are subjected to perturbation as per usual, but are then made to endure an additional step of being checked against the valid triples in the train and test sets. Should the perturbed triple appear in either of these sets, a new perturbation is generated so as to avoid populating the negative sample set with triples that are actually valid. An early addition to this simple scheme was introduced by Wang et al. for TransH and is sometimes called the Bernoulli trick [15]. The Bernoulli trick involves trying to reduce false negative triples by using different probabilities for the head and tail when performing a perturbation. This discrepancy is based on the mapping property of the relationship (i.e. one-to-one, many-to-one, one-to-many, and many-to-many).

We conclude by mentioning a few approaches that tried to enhance negative sampling specifically by exploiting type information [6]. While TRESCAL explicitly tries to make use of type constraints, it only explores their applicability to the RESCAL model [2]. On the other hand, the work by Toutanova et al. does not consult schema information directly, but instead defines entity types as a pair of sets [13]. The first set in this pair contains all the relationships for which the given entity has served as a subject, while the second contains those relationships for which the entity has served as an object. This is similar to the locally closed world approach proposed by Krompaß et al. [7]. In this same work, the general approach introduced by TRESCAL is extended to translation-based approaches. None of these investigate the possibility of using conditional constraints or consider an open world interpretation of constraint enforcement.

3 Problem Description

To perform graph completion, we focus on link prediction. The objective of link prediction is to estimate the degree of certainty with which an arbitrary link between two nodes might be said to exist. For the formal specification of the problem we follow Nickel et al. [10]. Taking RDF graphs as our template, we will refer to a KG as a tuple $(\mathcal{E}, \mathcal{R})$, where $\mathcal{E} = \{e_1, \dots e_{N_e}\}$ refers to the set of all distinct entities (subjects or objects, depending on the entity's role within a given relationship) in the graph and $\mathcal{R} = \{r_1, \dots r_{N_r}\}$ refers to the set of all dyadic relationships between such entities [10,11]. Every relationship $r_i \in \mathcal{R}$ is a binary relationship between entities. The knowledge graph can therefore be formulated as a subset of the collection of all possible triples: $(e_i, r_k, e_j) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. This is the most basic way of describing a KG, where all entities are treated as being ontologically equivalent. However, this description forgoes the conceptual differences between different *kinds* of entities (as indeed there are different *kinds* of relationships) expressed in most KGs adhering to the Resource Description Framework (RDF). If we distinguish classes $\mathcal{C}$ (entities of type *rdfs:Class* or *owl:Class*, which represent categories of entities) from other entities and also note that relationships can figure as subjects (e.g. when they are defined in a given schema), the definition of a KG can be reformulated as a subset of $(e_i, r_k, e_j) \in (\mathcal{E} \cup \mathcal{R} \cup \mathcal{C}) \times \mathcal{R} \times (\mathcal{E} \cup \mathcal{R} \cup \mathcal{C})$. Each *possible* triple $x_{ikj} = (e_i, r_k, e_j)$ is associated with a random variable $y_{ikj} \in \{0, 1\}$, for which:

$$y_{ikj} = \begin{cases} 1, & \text{if } x_{ikj} \text{ exists} \qquad (1) \\ 0, & \text{otherwise} \qquad (2) \end{cases}$$

We want to estimate $P(Y)$ with $y_{ikj} \in Y$, so that $Y \subseteq \{0,1\}^{N_e \times N_r \times N_e}$ (where N_e is the total number of assertional entities and N_r the total number of assertional relations), given a set of observed triples O and a parameter set Θ, i.e. $P(Y|O, \Theta)$ [10]. Here, O is composed of those triples for which we know that $y_{ikj} = 1$ (i.e. O^+), as well as those triples which have been fabricated to be false via negative sampling (i.e. O^-). Importantly, any triple (e_i, r_k, e_j) for which either $e_i \in \mathcal{C}$ or $e_j \in \mathcal{C}$, will not be included in the set of positive facts used to train the model, nor will such a triple be used for the purpose of evaluation. Triples belonging to the TBox or domain ontology of the KG will be employed strictly as supplementary knowledge for augmentation. Essentially,

$$\forall i, k, j, (e_i, r_k, e_j) \in O \implies e_i \notin \mathcal{C}, e_j \notin \mathcal{C} \qquad (3)$$

$$\forall (e_i, r_k, e_j) \in O, (e_i, r_k, e_j) \in O^+ \iff y_{ikj} = 1 \qquad (4)$$

$$\forall (e_i, r_k, e_j) \in O, (e_i, r_k, e_j) \in O^- \implies y_{ikj} = 0 \qquad (5)$$

4 Objectives

In this section we first discuss how constraints might be derived from ontological axioms defined within an open world view of knowledge. Once we have established how we might derive constraints in general, we will move on to the construction of conditional constraints based on OWL restrictions. Enforcing constraints can itself happen in an open or closed world fashion. Both of these alternatives will be discussed together with a way to modulate between them.

4.1 Constraint-Based Negative Sampling

As stated in section 1, the SW assumes an open world view of knowledge. The OWL language guide specifically states that "...OWL makes an *open world* assumption" and that "[new] information can be contradictory, but facts and entailments can only be *added*, never *deleted* [8]." Because of the OWA, what are considered to be constraints in fact serve as *logical axioms* for further inferencing. For instance, when one looks at the specification of *rdf:type* one finds that its possible subjects are "constrained" by *rdfs:domain* to instances of type *rdfs:Resource* and that its objects are similarly "constrained" by *rdfs:range* to instances of type *rdfs:Class*; in other words, *rdf:type* can express type instantiation between any entity (everything is an *rdfs:Resource*) and a class, which is what one would expect.

However, given the OWA, these "constraints" do not rule out that the *rdfs:-range* of *rdf:type* could later be expanded to include *rdfs:Resource.* Indeed, *rdfs:domain* and *rdfs:range* are not actually used for definitive exclusions of certain types with respect to relations, but rather serve as a vehicle for type inference. Rather than constraining subjects or objects of a certain relationship to certain types, such "constraints" are used to infer *additional* type information for these subjects or objects. To derive integrity constraints from logical axioms, an artificial closed world interpretation must be imposed. In fact, one can first make use of the axiomatic interpretation to expand the original ontology. As stated in section 1, deductive reasoning should be considered complementary to SRL for link prediction. Making use of the standard supra-domain ontologies[2,3,4], one can compute the deductive closure of any given domain ontology. Once the ontology has been expanded according to the OWA's internal logic, one can impose a restrictive interpretation on each logical axiom within the context of a negative sampling scheme. Indeed, given that it may be assumed that each entity's type declarations have been expanded beforehand according to what is already presupposed to be terminologically valid—according to the KG's TBox or domain ontology—whenever one encounters a triple where the participating entities' types are not axiomatically consistent, one can meaningfully say this triple must be invalid. In a sense, when imposing such a restrictive view, we are pursuing the same route as SHACL, which was introduced explicitly to address the practical unwieldiness of OWL's open world semantics [5].

4.2 Conditional Constraints

The domain and range constraints already exploited by previous approaches can easily be derived from their open world formulations [3]:

- *rdfs:domain* is an instance of *rdf:Property* that is used to state that any resource that has a given property *must be* an instance of one or more classes.
- *rdfs:range* is an instance of *rdf:Property* that is used to state that the values of a property *must be* instances of one or more classes.

Formally, one can define the domain and range *axioms* as follows:

$$\forall k \in \mathcal{K}, \forall c \in \mathcal{C}, \forall (r_k, \text{rdfs:domain}, c) \in \text{TBox} \implies \forall i,j \in \mathcal{I}, (e_i, r_k, e_j) \implies (e_i, \text{a}, c) \tag{6}$$

$$\forall k \in \mathcal{K}, \forall c \in \mathcal{C}, \forall (r_k, \text{rdfs:range}, c) \in \text{TBox} \implies \forall i,j \in \mathcal{I}, (e_i, r_k, e_j) \implies (e_j, \text{a}, c) \tag{7}$$

[2] `http://www.w3.org/1999/02/22-rdf-syntax-ns`(rdf)

[3] `http://www.w3.org/2000/01/rdf-schema` (rdfs)

[4] `http://www.w3.org/2002/07/owl` (owl)

where $\mathcal{K} = \{1 \dots N_r\}, \mathcal{I} = \{1 \dots N_e\}$, and the relationship a is short for *rdf:type*. The corresponding *integrity constraints* can be derived as follows:

$$\begin{aligned} \forall k \in \mathcal{K}, \forall c \in \mathcal{C}, \forall(r_k, \text{rdfs:domain}, c) \in \text{TBox} &\implies \\ \forall i, j \in \mathcal{I}, (e_i, \text{a}, c) &\implies (e_i, r_k, e_j) \text{ is valid} \qquad (8) \\ \forall k \in \mathcal{K}, \forall c \in \mathcal{C}, \forall(r_k, \text{rdfs:range}, c) \in \text{TBox} &\implies \\ \forall i, j \in \mathcal{I}, (e_j, \text{a}, c) &\implies (e_i, r_k, e_j) \text{ is valid} \qquad (9) \end{aligned}$$

Drawing inspiration from SHACL, we note that "[property restrictions] can only be [defined] within the context of an *owl:Restriction*... [where] [t]he *owl:onProperty* element indicates the restricted property [5]." Conditional constraints can thus be derived in the following manner:

- The *owl:allValuesFrom* restriction requires that for every instance of the class that has instances of the specified property, the values of the property *must all be* members of the class indicated by the *owl:allValuesFrom* clause [8].
- The *owl:someValuesFrom* restriction describes a class of all individuals for which at least one value of the property concerned *must be* an instance of the class description or a data value in the data range [12].

To clarify, *owl:allValuesFrom* and *owl:someValuesFrom* are *local* to their containing class definitions, meaning that their application is contingent on the subject being of the correct type (the type corresponding to the restricted class). For these restrictions, the *axioms* can formally be defined as follows:

$$\begin{aligned} &\forall k \in \mathcal{K}, \forall c, c' \in \mathcal{C}, \forall(b(c, r_k), \text{owl:onProperty}, r_k) \in \text{TBox} \;\; \& \\ &\qquad \forall(b(c, r_k), \text{owl:allValuesFrom}, c') \in \text{TBox} \\ &\qquad \implies \forall i, j \in \mathcal{I}, (e_i, a, c) \implies (e_i, r_k, e_j) \implies (e_j, \text{a}, c') \qquad (10) \\ &\forall k \in \mathcal{K}, \forall c, c' \in \mathcal{C}, \forall(b(c, r_k), \text{owl:onProperty}, r_k) \in \text{TBox} \;\; \& \\ &\qquad \forall(b(c, r_k), \text{owl:someValuesFrom}, c') \in \text{TBox} \\ &\qquad \implies \forall i \in \mathcal{I}, \exists j \in \mathcal{I}, (e_i, a, c) \implies (e_i, r_k, e_j) \;\; \& \;\; (e_j, \text{a}, c') \qquad (11) \end{aligned}$$

where $b(c, r_k)$ projects a restricted class $c \in \mathcal{C}$ onto the blank node representing its restriction for relation r_k. The corresponding *integrity constraints* are:

$$\begin{aligned} &\forall k \in \mathcal{K}, \forall c, c' \in \mathcal{C}, \forall(b(c, r_k), \text{owl:onProperty}, r_k) \in \text{TBox} \;\; \& \\ &\qquad \forall(b(c, r_k), \text{owl:allValuesFrom}, c') \in \text{TBox} \\ &\qquad \implies \forall i, j \in \mathcal{I}, (e_i, a, c) \implies (e_j, \text{a}, c') \\ &\qquad \implies (e_i, r_k, e_j) \text{ is valid} \qquad (12) \\ &\forall k \in \mathcal{K}, \forall c, c' \in \mathcal{C}, \forall(b(c, r_k), \text{owl:onProperty}, r_k) \in \text{TBox} \;\; \& \\ &\qquad \forall(b(c, r_k), \text{owl:someValuesFrom}, c') \in \text{TBox} \\ &\qquad \implies \forall i \in \mathcal{I}, \exists j \in \mathcal{I}, (e_i, r_k, e_j) \;\; \& \;\; (e_j, \text{a}, c') \implies (e_i, a, c) \\ &\qquad \implies (e_i, r_k, e_j) \text{ is valid} \qquad (13) \end{aligned}$$

Interpretations Based on these constraints, one can assume two alternative interpretations with respect to negative sampling. On the one hand, one can make use of an open world interpretation, where O^- contains only invalid triples (i.e. triples that do *not* satisfy the constraints), while on the other, a closed world interpretation can be imposed, where O^- contains only valid triples (i.e. triples that do satisfy the constraints).

In the prior case, we know that none of the negative examples will ever appear in the test set. Under this interpretation, every negative example is truly false; no possible facts are excluded except when they are nonsensical. It is no coincidence that this interpretation aligns best with the SW's OWA, where falseness is impossible except where nonsense is concerned.

In the latter case, we are in fact eliminating *useless* examples from O^-, on the assumption that nonsensical counterfactuals introduce needless model complexity because they only account for noise. Such facts are not useful for deriving a decision boundary between what exists and what does not. The problem with this interpretation is that it does permit O^- to be populated by false negatives.

In fact, both interpretations have merit, and it might be interesting to find a balance between them. To this end, we suggest making use of an additional hyperparameter to tune the reject rate of an invalid triple.

5 Evaluation

To evaluate the various types of constraints and contrast them with one another, TransE is used as a base model [1]. To generate counterfactuals, the basic filtered setting together with the Bernoulli trick is enhanced by introducing either an OWA or a CWA interpretation of constraint based sampling. The overall procedure goes as follows: First, the deductive closure of the domain ontology is computed by combining it with the aforementioned supra-domain ontologies. Having expanded the ontology, we now gather type information for all the entities in the dataset. Subsequently, the constraint rules are harvested from the expanded ontology by retrieving all relationships for which there exists either a *rdfs:domain* or *rdfs:range*, or an *owl:onProperty* (which indicates that the respective relationship is part of an *owl:Restriction* definition). The negative sampling procedure then proceeds by constructing per batch of positive triples a number of negative triples (equal to or a multiple of the number of positives) via Bernoulli-enhanced, filtered perturbation. Each potential triple generated via this scheme is then validated against the harvested constraint rules by making use of the type information gathered earlier. For both interpretations, because we are aware of each entity's type information, suggestions can be generated for possible replacement entities, thus significantly lowering the computation time.

For our evaluations, we make use of the AIFB[5] and MUTAG[6] datasets, the prior of which describes info related to research staff, institutions, and publications, and the latter of which contains details on potentially carcinogenic

[5] http://data.dws.informatik.uni-mannheim.de/rmlod/LOD_ML_Datasets/data/datasets/RDF_Datasets/AIFB/

[6] http://data.dws.informatik.uni-mannheim.de/rmlod/LOD_ML_Datasets/data/datasets/RDF_Datasets/MUTAG/

molecules. While typically used for entity classification rather than link prediction, these datasets have nonetheless been selected for their favourable characteristics. In particular, for AIFB we had access to 152 allValuesFrom OWL restrictions, and for MUTAG we were able to make use of 5087 RDFS constraints. For both of them, a 90-10 train-test split and a 90-10 train-valid split was performed. As a result, for AIFB, the training set contains 19916 entities, the valid set 2213 entities, and the test set 2459 entities. For MUTAG, the training set contains 41999 entities, the valid set 4667 entities, and the test set 5185 entities. For all experiments, the following hyperparameters were used for TransE: batch size 128, embedding size 100, number of epochs 100, learning rate 0.001, margin 1. We conducted separate experiments for each of the constraint interpretations and for different values of the *negative ratio* hyperparameter, which is used to tune the number of the negative samples generated per positive sample. The metrics used to evaluate performance are Hits@10, Mean Rank (MR), and Mean Reciprocal Rank (MRR), which are standard when evaluating link prediction performance.

The test results for AIFB can be found in Table 1, while those for MUTAG can be found in Table 2. Note that for each setting, we include the number of *distinct* false negatives generated by the sampling method.

Setting	neg ratio	MR	MRR	Hits@10	FN
no constraints	1	667.860	0.230	0.452	222
no constraints	5	**653.403**	**0.340**	**0.572**	647
open world constraints	1	663.658	0.208	0.412	173
open world constraints	5	N/A	N/A	N/A	N/A
closed world constraints	1	801.260	0.265	0.486	468
closed world constraints	5	870.899	0.250	0.468	985

Table 1: Results for AIFB

Setting	neg ratio	MR	MRR	Hits@10	FN
no constraints	1	3408.810	0.043	0.089	29
no constraints	5	2341.297	0.093	0.181	167
open world constraints	1	3166.192	0.051	0.105	12
open world constraints	5	2170.180	0.100	0.180	82
closed world constraints	1	3232.653	0.043	0.091	30
closed world constraints	5	**1975.455**	**0.110**	**0.215**	153

Table 2: Results for MUTAG

6 Discussion

Looking at the results, a few things immediately become clear: For both datasets, adding *closed world* constraints offers clear benefits. In AIFB, the OWL restrictions are able to improve the Hits@10 and the MRR when the negative ratio is kept at the default value of 1. In MUTAG, the RDFS domain and range constraints offer significant improvements with respect to every metric for negative ratios of both 1 and 5. The discrepancy between the results achieved for AIFB and MUTAG can be explained in terms of the number of false negatives encountered. For MUTAG, this number remains almost completely invariant when constraints are introduced and is furthermore almost negligible with respect to the size of the test set. Conversely, for AIFB, there is a significant increase in the number of sampled false negatives when introducing constraints. Interestingly enough, despite the increase in false negatives, for a negative ratio of 1, constraints did improve performance. For now, further investigation is required to determine the exact causes behind these discrepancies: likely, they are caused either by the conditional constraints themselves, or by the specific conditions (e.g. size, connectivity) of the AIFB dataset.

Interestingly, the *open world* constraints do seem to offer competitive results in certain cases. Rather than simply introducing unnecessary noise, for RDFS constraints (cfr. MUTAG), when the negative ratio is set to one the lower number of false negatives appears to improve performance. For the conditional constraints, however, this is not the case, mainly because of the low number of available constraints, which is unable to significantly diminish the number of false negatives. A hybrid approach making use of the suggested rejection parameter might be able to balance the benefits and drawbacks of both approaches. Finally, due to the complexity and sparsity of the conditional constraints, running the OWA for larger negative ratios was no longer computationally feasible. This problem seems inherent to the OWA when few constraints are available.

7 Conclusion & Future Work

In this research paper we provided a thorough justification for the derivation of integrity constraints from logical axioms. We supplied formal definitions for each of the types of constraints used during negative sampling. Furthermore, we introduced a new form of conditional constraints based on OWL restrictions and laid out two opposing views on how constraints might be applied to improve negative sampling. We then evaluated these constraints under the alternative interpretations, while also verifying the effect of the negative sampling ratio. In the future we would like to take this research much further. First, we would like to systematically contrast the effect of RDFS constraints with those of OWL restrictions. To this end, we will perform a (partial) conversion from the prior to the latter and vice versa within a given dataset. Second, we would like to empirically validate the effect of adding a rejection hyperparameter. Third, we would like to investigate the effect of these strategies on other embedding techniques besides TransE. Finally, we would like to investigate the possibility of using constraints to combat noise in the training set. All of these possibilities are currently being evaluated.

Acknowledgment: This research was funded by the imec.ICON RADIANCE project, co-funded by VLAIO, imec, Skyline Communications, Barco and ML6.

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. Chang, K.W., Yih, W.t., Yang, B., Meek, C.: Typed tensor decomposition of knowledge bases for relation extraction. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1568–1579 (2014)
3. Guha, R., Brickley, D.: RDF schema 1.1. W3C recommendation, W3C (Feb 2014), http://www.w3.org/TR/2014/REC-rdf-schema-20140225/
4. Hendler, J., Lassila, O., Berners-Lee, T.: The semantic web. Scientific American **284**(5), 34–43 (2001)
5. Knublauch, H., Kontokostas, D.: Shapes constraint language (SHACL). W3C recommendation, W3C (Jul 2017), https://www.w3.org/TR/2017/REC-shacl-20170720/
6. Kotnis, B., Nastase, V.: Analysis of the impact of negative sampling on link prediction in knowledge graphs. arXiv preprint arXiv:1708.06816 (2017)
7. Krompaß, D., Baier, S., Tresp, V.: Type-constrained representation learning in knowledge graphs. In: International semantic web conference. pp. 640–655. Springer (2015)
8. McGuinness, D., Welty, C.: OWL web ontology language guide. W3C recommendation, W3C (Feb 2004), http://www.w3.org/TR/2004/REC-owl-guide-20040210/
9. Mehla, S., Jain, S.: Rule languages for the semantic web. In: Emerging Technologies in Data Mining and Information Security, pp. 825–834. Springer (2019)
10. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proceedings of the IEEE **104**(1), 11–33 (2015)
11. Nickel, M., Tresp, V.: Tensor factorization for multi-relational learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 617–621. Springer (2013)
12. Schreiber, G., Dean, M.: OWL web ontology language reference. W3C recommendation, W3C (Feb 2004), http://www.w3.org/TR/2004/REC-owl-ref-20040210/
13. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality. pp. 57–66 (2015)
14. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017)
15. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI conference on artificial intelligence (2014)
16. Yoon, H.G., Song, H.J., Park, S.B., Park, S.Y.: A translation-based knowledge graph embedding preserving logical property of relations. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 907–916 (2016)