Embedding Cardinality Constraints in Neural Link Predictors

Emir Muñoz
Data Science Institute,
National University of Ireland Galway
Galway, Ireland
emir@emunoz.org

Pasquale Minervini
University College London
London, United Kingdom
p.minervini@cs.ucl.ac.uk

Matthias Nickles
Data Science Institute,
National University of Ireland Galway
Galway, Ireland
matthias.nickles@nuigalway.ie

ABSTRACT

Neural link predictors learn distributed representations of entities and relations in a knowledge graph. They are remarkably powerful in the link prediction and knowledge base completion tasks, mainly due to the learned representations that capture important statistical dependencies in the data. Recent works in the area have focused on either designing new scoring functions or incorporating extra information into the learning process to improve the representations. Yet the representations are mostly learned from the observed links between entities, ignoring commonsense or schema knowledge associated to the relations in the graph. A fundamental aspect of the topology of relational data is the cardinality information, which bounds the number of predictions given for a relation between a minimum and maximum frequency. In this paper, we propose a new regularisation approach to incorporate *relation cardinality constraints* to any existing neural link predictor without affecting their efficiency or scalability. Our regularisation term aims to impose boundaries on the number of predictions with high probability, thus, structuring the embeddings space to respect commonsense cardinality assumptions resulting in better representations. Experimental results on Freebase, WordNet and YAGO show that, given suitable prior knowledge, the proposed method positively impacts the predictive accuracy of downstream link prediction tasks.

CCS CONCEPTS

• **Computing methodologies** → **Semantic networks**; *Statistical relational learning*;

KEYWORDS

Knowledge graphs, cardinality constraints, commonsense knowledge, regularisation

1 INTRODUCTION

Cognitive development of children indicates that we learn the cardinality-related question *"How many?"* at ca. 3.5 years of age [38]. This ability helps us to recognise physical and abstract things by counting. For example, a hand has commonly five fingers, a car has four wheels, or a meeting has at least two participants. This kind of common sense knowledge is not obvious for machines to acquire, even in contexts where it can be useful, such as Question Answering, Web Search, and Information Extraction [34].

One fundamental application area for cardinality information relates to the completion of Knowledge Graphs (KGs), graph-structured knowledge bases where factual knowledge is represented in the form of relationships between entities. For instance, consider Freebase [2], the core of the Google Knowledge Graph project, where 71% of the people described in it have no known place of birth as reported by Dong et al. [9]. By leveraging cardinality information about the *bornIn* relationship (i.e., each person must have a place of birth), we can quantitatively assess the degree of incompleteness in Freebase and focus the resources on predicting a single place of birth for each person. Yet *link prediction models* aimed at identifying missing facts in KGs do not consider such commonsense or schema knowledge, yielding potentially inconsistent and inaccurate predictions.

Triples	Probability
$(edgar, hasParent, edgar)$	0.989
$(edgar, hasParent, eliza\_poe)$	0.979
$(edgar, hasParent, virginia\_eliza\_clemm\_poe)$	0.974
$(edgar, hasParent, julia\_ward\_howe)$	0.890
$(edgar, hasParent, benjamin\_franklin)$	0.889

Table 1: Top-5 predictions (among 24 results with probability > 0.8) for the *hasParent* relation with *Edgar Allan Poe* given by DistMult [39] on the FB13 dataset [5].

In this work, we focus on a certain class of link prediction models, namely *Neural Link Predictors* [26]. Such models learn low-dimensional distributed representations—also referred to as *embeddings*—of all entities and relations in a knowledge graph. Neural link predictors are currently the state of the art approach to tasks such as link prediction [4, 8, 35, 39], entity disambiguation and entity resolution [3], taxonomy extraction [25, 29], and probabilistic question answering [17].

Recently, research focused mainly on designing new scoring functions, and incorporating additional background knowledge during the learning process. We refer readers to [26, 36] for a recent overview on this topic.

In this paper, we address the problem of incorporating prior knowledge in the form of relation cardinality information into state-of-the-art neural link predictors. For instance, we want to encode prior knowledge in the form of cardinality statements such as *"a person should have at most two parents"* or *"a patient should be taking between 1 and 5 drugs at a time"* in neural link prediction models. Such prior knowledge can be provided by domain experts, or automatically extracted from data [11, 24]. It is expected that such cardinality constraints will be satisfied by both the facts in the knowledge graph and algorithms analysing the graph, such as link predictors. We believe that these constraints can impose commonsense knowledge upon the structure of the embedding space, thus helping us to learn better representations.

Cardinality constraints are one of the most important constraints in conceptual modelling [30, Chapter 4] as they explicit the topology

of data. However, existing neural link prediction models are not designed to incorporate them for learning better representations and more accurate models.

Example. One may expect that when predicting the parents (represented by relation *hasParent*) for the entity *Edgar Allan Poe*, a model will predict at most two parents, preferably *Eliza Poe* and *David Poe Jr.* To illustrate this, let us analyse the actual predictions of a state-of-the-art neural link prediction model, DistMult [39], using the Freebase FB13 dataset [5], containing entities of the Freebase type *deceased people* and their relations. Table 1 shows the top-5 predicted parents for *Edgar Allan Poe*. As we can see, all predictions have a high probability (with 24 entities scored higher than 0.8), albeit some predictions are incorrect.

Nevertheless, the evaluation results of our example model are positive due to the evaluation protocol of link prediction models based on a ranking metric, where correct predictions (e.g., *eliza_poe*) are expected to be ranked higher than incorrect ones (e.g., *benjamin_franklin*).

To address this problem, in this paper we propose an efficient approach for embedding the notion of cardinality in neural link prediction models, without affecting their efficiency and scalability. The proposed approach is based on a novel regularisation term, that constraints the number of predictions for a given relation. Briefly, our idea is to penalise the model when its predictions violate one cardinality constraints, expressed as lower or upper bound on the cardinality of a given relation type. By doing so, the notion of cardinality of a relation will be captured during training, yielding to more accurate link prediction models, that comply with available prior knowledge [37], and learn better representations for entities and relations in the knowledge base.

Organisation. The remainder of this paper is organised as follows. First we present the definitions of knowledge graphs and neural link prediction models in Section 2. Next we present the concept of relation cardinality constraint for knowledge graphs in Section 3. In Section 4, we introduce a cardinality regularisation term which allows neural link predictors to leverage available cardinality constraints. We evaluate the application of our regularisation term over different datasets and models in Section 5. Section 6 briefly discusses the existing works in link prediction over knowledge graphs. Finally, Section 7 concludes this paper.

2 BACKGROUND

We start by introducing the fundamentals of knowledge graphs and neural link predictors.

Definition 1 (Knowledge Graphs)**.** A *knowledge graph* is a graph representation of a knowledge base. Let $\mathcal{E}$ be the set of all entities, and $\mathcal{R}$ the set of all relation types (predicates). We denote by $\mathcal{G}$ a knowledge graph comprising a set of (h, r, t) facts or triples, where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$. We refer to h, t as *subject* and *object* entities and to r as *relation* of a triple. Let $N_e = |\mathcal{E}|$ and $N_r = |\mathcal{R}|$ be the number of entities and relations, respectively.

The goal of *link prediction* models is to learn a scoring function ϕ that given a triple (h, r, t) returns its corresponding *score*, $\phi(h, r, t) \mapsto \mathbb{R}$. Such a score can then be used for ranking missing triples according to the likelihood that the corresponding facts hold true.

Definition 2 (Neural Link Predictors)**.** *Neural link prediction models* [26, 36] can be interpreted as neural networks consisting of an *encoding layer* and a *scoring layer*. Given a triple (h, r, t), the encoding layer maps entities $h, t \in \mathcal{E}$ to their k-dimensional distributed representations $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$. Then, the scoring layer computes the likelihood of the triple based on a relation-dependent function ϕ_r. Henceforth, the scoring function ϕ is defined as $\phi(h, r, t) = \phi_r(\boldsymbol{e}_h, \boldsymbol{e}_t,)$, where $\phi_r : \mathbb{R}^k \times \mathbb{R}^k \mapsto \mathbb{R}$, $\boldsymbol{e}_h, \boldsymbol{e}_t \in \mathbb{R}^k$, and $r \in \mathcal{R}$.

A neural link predictor with parameters Θ defines a conditional probability distribution over the truth value of a triple (h, r, t) [26]:

$$p(y_{hrt} = 1 \mid \Theta) = \sigma(\phi_r(\boldsymbol{e}_h, \boldsymbol{e}_t)), \tag{1}$$

where $y_{hrt} \in \{0, 1\}$ is the truth label of the triple, $\Theta = \{\boldsymbol{e}_i\}_{i=1}^{N_e} \cup \{\boldsymbol{r}\}_{j=1}^{N_r}$ denotes the set of all entity and relation embeddings (the parameters Θ), $\sigma(x) = 1/(1 + \exp(-x))$ is the standard logistic function, and ϕ_r denotes the model's scoring function (cf. Table 2). Most models consider the k-dimensional embeddings as real-valued $\boldsymbol{e}_h, \boldsymbol{e}_t, \boldsymbol{r}_r \in \mathbb{R}^k$; however, there are exceptions like ComplEx [35], where $\boldsymbol{e}_h, \boldsymbol{e}_t, \boldsymbol{r}_r \in \mathbb{C}^k$.

A neural link prediction model is trained by minimising a loss function defined over a target knowledge graph $\mathcal{G}$, usually using stochastic gradient descent. Since knowledge graphs only contain positive examples (i.e. facts), a way to provide negative learning examples—motivated by the Local Closed World Assumption (LCWA) [9]—is to generate negative examples by *corrupting* the triples in the graph [4, 26, 31]. Given a (positive) triple $(h, r, t) \in \mathcal{G}$, corrupted triples (negative examples) can be generated by replacing either the subject or object with a random entity sampled uniformly from $\mathcal{E}$ [5]. Formally, given a positive example (h, r, t), negative examples are sampled from the set of possible corruptions of (h, r, t), namely $C(h, r, t) \triangleq \{(h', r, t) \mid h' \in \mathcal{E}\} \cup \{(h, r, t') \mid t' \in \mathcal{E}\}$.

Let $\mathcal{D}^+$ be the set of positive examples, and $\mathcal{D}^-$ the set of negatives generated accordingly with function C. The training consists of learning the parameters Θ that best explain $\mathcal{D}^+$ and $\mathcal{D}^-$ according to Eq. (1). For that, models such as TransE [4], DistMult [39] and HolE [27] minimise a pairwise margin loss:

$$\mathcal{L}(\Theta) = \sum_{\tau^+ \in \mathcal{D}^+} \sum_{\tau^- \in \mathcal{D}^-} \left[\gamma + \sigma(\phi(\tau^-)) - \sigma(\phi(\tau^+))\right]_+, \tag{2}$$

where $\tau^+ = (h, r, t)$ is a positive example, $\tau^- = (h', r, t')$ is a negative one, $[x]_+ = \max(0, x)$, and γ is the margin hyperparameter. The entity embeddings are also constrained to unit norm, i.e. $\forall i \in \mathcal{E} : \|\boldsymbol{e}_i\|_2 = 1$. Whereas other models like ComplEx [35] minimise the logistic loss:

$$\mathcal{L}(\Theta) = \sum_{\tau \in \mathcal{D}^+ \cup \mathcal{D}^-} \log(1 + \exp(-y_\tau \phi(\tau)))$$

where $\tau = (h, r, t)$ is an example (triple), and $y_\tau \in \{-1, 1\}$ is the label (negative or positive) associated with the example.

3 RELATION CARDINALITIES

A relation type can have associated cardinality bounds, which restrict the number of object values that a subject can have.

Model	Scoring Function	Parameters
ER-MLP	$\boldsymbol{w}^T \tanh\left(\boldsymbol{W}^T[\boldsymbol{e}_h;\boldsymbol{e}_t;\boldsymbol{r}_r]\right)$	$\boldsymbol{r}_r \in \mathbb{R}^k, \boldsymbol{w} \in \mathbb{R}^{k'}$ $\boldsymbol{W} \in \mathbb{R}^{3k\times k'}$
DistMult	$\langle \boldsymbol{e}_h, \boldsymbol{r}_r, \boldsymbol{e}_t \rangle$	$\boldsymbol{r}_r \in \mathbb{R}^k$
ComplEx	$\mathrm{Re}(\langle \boldsymbol{e}_h, \boldsymbol{r}_r, \bar{\boldsymbol{e}}_t \rangle)$	$\boldsymbol{r}_r \in \mathbb{C}^k$

Table 2: Scoring functions $\phi_r(\boldsymbol{e}_h, \boldsymbol{e}_t)$ of three state-of-the-art knowledge graph embedding models.

Definition 3 (Relation Cardinality Bound)**.** Let $\varphi_r = (\varphi_r^{\downarrow}, \varphi_r^{\uparrow})$ be a *cardinality bound* for the relation $r \in \mathcal{R}$, where $\varphi_r^{\downarrow} \in \mathbb{N}$ denotes the *lower bound* and $\varphi_r^{\uparrow} \in \mathbb{N} \cup \{\infty\}$ denotes the *upper bound* of the cardinality, s.t. $0 \leq \varphi_r^{\downarrow} \leq \varphi_r^{\uparrow}$ [24]. A knowledge graph $\mathcal{G}$ *satisfies* a cardinality bound φ_r with $r \in \mathcal{R}$ iff

$$\forall h \in \mathcal{E}, (\varphi_r^{\downarrow} \leq count(r, h) \leq \varphi_r^{\uparrow}),$$

where $count(r, h)$ is the number of triples with h as subject and r as relation [24].

Example. Given a cardinality bound $\varphi_{hasParent} = (0, 2)$, encoding the constraint *"a person should have at most two parents"*, we would like to ensure that the embeddings learned by a neural link predictor yield predictions for the *hasParent* relation within the boundaries. In other words, we want to have the sum of probabilities over all possible parent entities of *Edgar Allan Poe* precisely between zero and two.[1] We express this constraint over the triple $\tau = (edgar\_allan\_poe, hasParent, t)$ as:

$$0 \leq \sum_{t \in \mathcal{E}} p(y_{hrt} = 1 \mid \Theta) \leq 2, \tag{3}$$

where the conditional probabilities $\forall t \in \mathcal{E}$ are given by the neural link prediction model.

This term in Eq. (3) expresses a supervision signal, not based on labelled data, that can be input to the training of neural link prediction models. It is worth to mention that such cardinality boundaries can be provided by experts, gathered from literature [23], or extracted from knowledge bases [11, 24].

4 REGULARISATION BASED ON CARDINALITY

In this section, we propose an approach to incorporate cardinality bounds in the training of neural link prediction models. Specifically, we propose to leverage the available cardinality bounds, expressed as in Eq. (3), to define a regularisation term that encourages models to respect the available cardinality constraints.

Let $\Phi = \{\varphi_r = (\varphi_r^{\downarrow}, \varphi_r^{\uparrow})\}_{r \in \mathcal{R}}$ be the set of cardinality constraints for each relation in a given knowledge graph $\mathcal{G}$, where $\varphi_r^{\downarrow}$ and $\varphi_r^{\uparrow}$ are the lower and upper bound for relation r, respectively.

Given $r \in \mathcal{R}$ and $h \in \mathcal{E}$, let $\mathcal{A}_{hr}[\mathcal{E}] \triangleq \{(h, r, t) : \forall t \in \mathcal{E}\}$ be the set of all possible triples with relation r and subject h, where the object t was selected from $\mathcal{E}$. Following our toy example, assume that r denotes the relation *hasParent*, and h denotes the entity *edgar_allan_poe*. Hence, we can take the set of possible triples to define the following hard constraint on the conditional probability of the triples in $\mathcal{A}_{hr}[\mathcal{E}]$:

$$\varphi_r^{\downarrow} \leq \left(\mathcal{X}_{hr}[\mathcal{E}] \triangleq \sum_{x_{hrt} \in \mathcal{A}_{hr}[\mathcal{E}]} p_\Theta(y_{hrt} = 1 \mid \Theta) \right) \leq \varphi_r^{\uparrow}. \tag{4}$$

However, the inequality constraint in Eq. (4) is impractical to incorporate directly in neural link predictors.

In this work, we propose a continuous relaxation of the constraint in Eq. (4) to a *soft constraint*, by defining a continuous and differentiable loss function that penalises violations of such a constraint. Specifically, we define a function G_{hr} that is strictly positive if the cardinality constraint for a given entity h and relation r is violated, and zero otherwise. Given a cardinality constraint φ_r, the function $G_{hr}[\mathcal{E}; \Phi]$ (or G_{hr} for simplicity) is defined as follows:

$$\begin{aligned} G_{hr}[\mathcal{E}; \Phi] = &\max(0, \varphi_r^{\downarrow} - \mathcal{X}_{hr}[\mathcal{E}]) + \\ &\max(0, \mathcal{X}_{hr}[\mathcal{E}] - \varphi_r^{\uparrow}). \end{aligned} \tag{5}$$

Figure 1 shows the values of G_{hr} (Eq. (5)) based on $\mathcal{X}_{hr}[\mathcal{E}]$ and a cardinality bound $\varphi_r \in \Phi$. Notice that for the general case where the upper bound corresponds to ∞ and lower bound to 0, the loss $G_{hr}[\mathcal{E}; \Phi]$ vanishes.

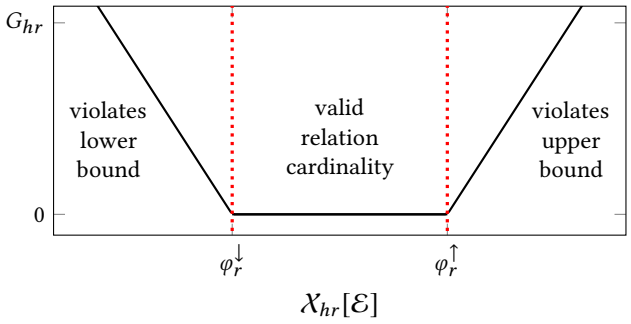


Figure 1: Regularisation term G_{hr} based on the bounds of a cardinality constraint $\varphi_r = (\varphi_r^{\downarrow}, \varphi_r^{\uparrow})$.

Therefore, we define a cardinality-regularised objective function, denoted by $\mathcal{L}^C(\Theta)$, for neural link prediction models:

$$\mathcal{L}^C(\Theta) = \mathcal{L}(\Theta) + \lambda \sum_{\Phi} G_{hr}[\mathcal{E}; \Phi], \tag{6}$$

where $\lambda \in \mathbb{R}_+$ weights the relative contribution of the regularisation term, and $\mathcal{L}(\Theta)$ can be either the pairwise ranking loss or the logistic loss. The regularised loss Eq. (6) can be minimised using stochastic gradient descent (SGD) [32] in mini-batch mode, outlined in Algorithm 1.

Although our approach considers both upper and lower bounds, the latter cannot be meaningfully imposed in all cases. For instance, given a constraint $\varphi_{spouse} = (1, 1)$, the regularisation term $G_{hr}[\mathcal{E}; \Phi]$ can yield inconsistent results if the knowledge graph is incomplete, and does not contain the *spouse* link of every person. In such cases, a zero lower bound can be used to address the knowledge graph incompleteness.

[1] Note that by considering a lower bound equals to zero, we can account for the possible incompleteness of the KG.

Algorithm 1 Learning the model parameters Θ via projected SGD

Input: Observed facts $\mathcal{D}^+$, epochs τ, initial learning rate $\eta \in \mathbb{R}$
Output: Optimal model parameters Θ (see [26])

Initialise embeddings $\boldsymbol{e}$ and $\boldsymbol{r}$ according to [13]
for $i = 1, \ldots, \tau$ **do**
  ◃ *Build batch for training*
  $\mathcal{T} \leftarrow$ sample a batch from $\mathcal{D}^+$
  $\mathcal{B}^+ \leftarrow \emptyset, \mathcal{B}^- \leftarrow \emptyset$
  for $\tau^+ = (h, r, t) \in \mathcal{T}$ **do**
    $\tau^- \in C(h, r, t)$ ◃ *Sample negative example*
    $\mathcal{B}^+ \leftarrow \mathcal{B}^+ \cup \{\tau^+\}, \mathcal{B}^- \leftarrow \mathcal{B}^- \cup \{\tau^-\}$
  end for
  ◃ *Compute the gradient of the loss function* $\mathcal{L}$
  $g_i \leftarrow \nabla \mathcal{L}(\Theta)$ using $\mathcal{B}^+$ and $\mathcal{B}^-$
  ◃ *Model parameters update via gradient descent*
  $\Theta_i \leftarrow \Theta_{i-1} - \eta_i g_i$
  ◃ *Projection step normalising all entity embeddings*
  $\boldsymbol{e} \leftarrow \boldsymbol{e}/||\boldsymbol{e}||, \ \forall e \in \mathcal{E}$
end for
return Θ

Our approach is intuitive and easy to implement for any neural link prediction model. However, it is limited by the cost of computing the sum in Eq. (4): the set $\mathcal{A}_{hr}[\mathcal{E}]$ can easily grow in some KGs and become too expensive to obtain the sum of probabilities. In the following section, we propose to use sampling techniques to overcome this problem by approximating the sum of probabilities.

4.1 Lower Bound Estimation

We can sample a subset of all entities $\mathcal{S} \subseteq \mathcal{E}$ and obtain the following lower bound:

$$\mathcal{X}_{hr}[\mathcal{S}] \leq \mathcal{X}_{hr}[\mathcal{E}]. \tag{7}$$

The tightness of the bound in Eq. (7) is determined by the selection of the entities in $\mathcal{S}$. In this work, we consider *uniform sampling*. More specifically, a random set of indices $\mathcal{S} \triangleq \{i_1, \ldots, i_S\}$ is taken uniformly, where $i_s \in \{1, \ldots, C\}$, and form the following lower bound:

$$\sum_{x_{hrt} \in \mathcal{A}_{hr}[\mathcal{S}]} p(y_{hrt} = 1 \mid \Theta) \leq \mathcal{X}_{hr}[\mathcal{E}],$$

where the sum is over all elements in $\mathcal{S}$ with no repetitions.

4.2 Sum Estimation

Instead of defining a lower bound to $\mathcal{X}_{hr}[\mathcal{E}]$, we can also approximate $\mathcal{X}_{hr}[\mathcal{E}]$ directly by *sampling*. Let us consider a sum over a large collection of elements $Z \triangleq \sum_c z_c$. We consider two standard methods for approximating sums via Monte Carlo estimates, namely Importance Sampling (IS) and Bernoulli Sampling [6].

Importance Sampling. Based on the identity $Z = \sum_c \frac{q(c) z_c}{q(c)}$, a set of indices $\mathcal{S} \equiv \{i_1, \ldots, i_S\}$ is selected from a distribution q, where $i_s \in \{1, \ldots, C\}$, and yielding the following approximation:

$$Z \approx \frac{1}{S} \sum_{s \in \mathcal{S}} \frac{z_s}{q(s)},$$

where $q(s)$ defines the probability of sampling s from $\mathcal{S}$.

Bernoulli Sampling. An alternative to IS is Bernoulli Sampling (BS), considering the following identity:

$$Z = \sum_c z_c = \mathbb{E}_{\mathbf{s} \sim \mathbf{b}} \left(\sum_c \frac{s_c}{b_c} z_c \right),$$

where each independent Bernoulli variable $s_c \in \{0, 1\}$ denotes whether z_c will be sampled or not, and $p(s_c = 1) = b_c$ is the probability of sampling z_c. This leads to the following approximation:

$$Z \approx \sum_{c : s_c = 1} \frac{z_c}{b_c},$$

where the sum is computed over the components with non-zero elements in the vector $\mathbf{s}$. Note that, when calculating an approximation to Z, IS relies on sampling with replacement, while BS relies on sampling without replacement.

By using our regularisation term with sampling, we add a time complexity $O(cd)$, where c is the total number of (sampled) triples when computing the regularisation term, and d the number of triples per batch. Since c can be smaller than the number of triples in a batch, we ensure that the time complexity of neural link predictors is not sensibly affected during training, and not affected at all at test time. The proposed method does not increase the space complexity of the models, since the proposed regulariser does not change the number of model parameters.

5 EVALUATION

In this section, we investigate the benefits of cardinality regularisation for the state-of-the-art neural link prediction models. We compare the performance of original and regularised losses in the link prediction task across different benchmark datasets, which are partitioned into train, validation and test set of triples (cf. Table 3).

5.1 Evaluation Protocol

The link prediction task consists of predicting a missing entity h or t when given a pair (r, t) or (h, r), respectively. During testing, for each test triple (h, r, t), we replace the subject or object entity with all entities in the knowledge graph as corruptions [4]. The evaluation then ranks the entities in descending order w.r.t. the scores calculated by a scoring function and gets the rank of the correct entity h or t. We report results based on the ranks assigned to correct entities measured using mean reciprocal rank (MRR) and Hits@n with $n \in \{1, 3, 5, 10\}$.[2] During the ranking process some positive test triples could be ranked after another true triples, which should not be considered a mistake. Therefore, the above metrics have two settings: *raw* and *filtered* [4]. In the filtered setting, metrics are computed after removing all true triples appearing in train, validation, or test sets from the ranking, whereas in the raw setting they are not removed.

5.2 Datasets

Three widely used datasets for evaluating link prediction models are WordNet [19], Freebase [2], and YAGO [18]. In this work, we

[2] For MRR and Hits@n, the higher the better.

Dataset	N_r	N_e	|train|	|validation|	|test|
FB13	13	81,065	350,517	5,000	5,000
WN18	18	40,943	14,1442	5,000	5,000
WN18RR	11	40,943	86,835	3,034	3,134
YAGO3-10	37	123,182	1,079,040	5,000	5,000

Table 3: Statistics for each of the datasets.

/people/person/place_of_birth	(0, 2)
/people/person/parents	(0, 2)
/people/person/gender	(1, 1)
_hyponym	(0, 380)
_has_part	(0, 73)
_hypernym	(0, 4)
livesIn	(0, 12)
hasGender	(0, 1)
hasChild	(0, 19)

Table 4: Cardinality constraints extracted from FB13, WN18 (WN18RR) and YAGO3-10.

use four benchmark datasets generated from them: FB13, WN18, WN18RR and YAGO3-10.

The FB13 dataset [5] is a subset of Freebase containing 13 relation types and entities of type *deceased_people*, where entities appear in at least 4 relations and relation types at least 5,000 times. [3] We also use two datasets derived from WordNet, namely, WN18 and WN18RR. These datasets contain hyponym, hypernym, and other lexical relations of English concepts and words. It is known that WN18 contains ca. 72% of redundant and inverse relations, which were removed in the WN18RR dataset [7]. YAGO3-10 consists of entities in YAGO3 (mostly of the people type) linked with at least 10 relations, such as citizenship, gender and profession. FB13, WN18RR, and YAGO3-10 datasets were shown to have no redundant or trivial triples [7]. In Table 3 we summarise the characteristics of each of the datasets.

We mine the relation cardinality constraints from the training set of each dataset, following the algorithm proposed by Muñoz and Nickles [24] using the normalisation option but without filtering outliers. Table 4 gives examples of the cardinality constraints mined from each dataset.

5.3 Results

For our experiments, we re-implemented three models using the TensorFlow framework [1], namely, ER-MLP [9], DistMult [39] and ComplEx [35] (which was recently proven to be equivalent to HolE [16]). We compare the performance over the four benchmark datasets of each model as originally stated by their authors and with the cardinality regularisation term (cf. Eq. (6)).

[3]We use the corrected version by [33] that contains only positive samples.

As recommended by [35], we minimise the logistic loss to train each model by using SGD, and AdaGrad [10] to adaptively select the learning rate, initialised as $\eta_0 = 0.1$. For each model and dataset, we selected hyperparameters maximising filtered Hits@10 on the validation set using an exhaustive grid search.

The evaluation of our approach is three-fold: (i) we measure the effects of the regulariser in the link prediction task; (ii) we measure the effects of the different sampling techniques; and (iii) we measure the violations to the cardinality constraints before and after regularisation. To reduce the search space, during the grid search in (i) we fix the sampling technique to uniform. In (ii), we use the best model identified in (i) to study the effect of different sampling techniques, whilst in (iii) we use the overall best model per dataset.

Link Prediction. We train each model for 1,000 epochs with a mini-batches approach over the training set of each dataset, generating two negative examples per positive triple in each batch. We set $\lambda = 0$ to obtain the performance results of original models (without regularisation), and use uniform sampling with sizes $\mu \in \{10, 100\}$, $\omega \in \{10, 100, 1000\}$ of subjects and objects.[4]

Tables 5 and 6 show the link prediction results, confirming that in general our cardinality-based regularisation term helps to improve (or at least maintain) the performance of the original ER-MLP, DistMult and ComplEx models across all datasets. The only exception we observed is ComplEx over YAGO3-10, where the model without the regularisation term reaches better Hits@10 and MRR. We believe that a reason for this is that constraining a lower bound on the sum of probabilities may not be the best technique to use when the number of entities is very large. In our experiments we also compare two alternative approaches, namely estimating the sum of probabilities via IS and BS.

ER-MLP and DistMult models benefit the most across all datasets with improvements of up to 36% in MRR. ComplEx shows to be the overall best performing model outperforming ER-MLP (up to 20x in WN18RR) and DistMult in every dataset and evaluation metric. Still, ComplEx benefits from the regularisation term in most of the datasets. Although we did not perform a thorough search of the hyperparameters space to reach state-of-the-art performance, the results prove the advantages of our approach.

Sampling techniques. To approximate the sum of probabilities we test both Importance Sampling and Bernoulli Sampling, and consider hyperparameters $\mu \in \{10, 50, 100\}$ and $\omega \in \{10, 50, 100, 500, 1000\}$. Starting from the best ComplEx models learned above, we tune the sampling technique for each of the datasets.

Results are shown in Table 7. In general, all sampling techniques work well and there is no *one-size-fits-all* solution: it depends on the dataset. (Information about properties of the data that benefit one of the samplings can be used, and custom sampling is also supported.) YAGO3-10 shows the biggest improvement of 6% in MRR using BS compared with the results in Table 6. This improvement might be correlated to the advantage of BS to handle the large number of entities in YAGO3-10. For FB13, WN18, and WN18RR we see smaller improvements in MRR and Hits@10 compared to the

[4]We identified via independent experiments that larger values for μ do not yield performance improvements.

Method	FB13 Hits@1	FB13 Hits@3	FB13 Hits@5	FB13 Hits@10	FB13 MRR	WN18 Hits@1	WN18 Hits@3	WN18 Hits@5	WN18 Hits@10	WN18 MRR	WN18RR Hits@1	WN18RR Hits@3	WN18RR Hits@5	WN18RR Hits@10	WN18RR MRR
ER-MLP	4.40	7.55	9.14	11.82	6.94	21.64	37.30	44.94	56.52	33.02	1.84	3.29	4.10	5.31	3.10
ER-MLP^C	**5.13**	**8.36**	**10.29**	**12.75**	**7.78**	**32.01**	**51.54**	**60.54**	**70.85**	**45.01**	**2.22**	**4.29**	**5.42**	**7.31**	**3.98**
DistMult	18.07	29.29	32.94	37.01	24.92	64.46	87.47	90.66	93.49	76.62	38.93	43.49	45.93	49.63	42.46
DistMult^C	**18.10**	**29.45**	**33.07**	**37.02**	**25.00**	**65.01**	**87.53**	**90.71**	**93.44**	**76.93**	**39.10**	**44.13**	**46.30**	**49.81**	**42.84**
ComplEx	**25.08**	31.64	34.00	36.90	**29.41**	88.33	93.05	94.14	95.07	90.96	40.87	**46.25**	**48.55**	**51.15**	44.52
ComplEx^C	24.89	**31.78**	**34.10**	**37.16**	29.36	**88.66**	**93.27**	**94.21**	**95.21**	**91.20**	**41.10**	46.06	48.13	51.09	**44.57**

Table 5: Link prediction results (Hits@n and Mean Reciprocal Rank, filtered setting) on FB13, WN18 and WN18RR. In bold the best results comparing both original and cardinality loss, and highlighted is the best value per evaluation metric across all models.

Method	YAGO3-10 Hits@1	YAGO3-10 Hits@3	YAGO3-10 Hits@5	YAGO3-10 Hits@10	YAGO3-10 MRR
ER-MLP	2.22	6.09	9.59	16.01	6.83
ER-MLP^C	**2.33**	**6.16**	**9.65**	**16.54**	**6.95**
DistMult	6.75	14.33	18.86	26.51	13.33
DistMult^C	**7.03**	**14.53**	**19.12**	**26.66**	**13.59**
ComplEx	7.12	**15.61**	**20.76**	**29.11**	14.33
ComplEx^C	**7.56**	15.10	20.30	29.01	**14.47**

Table 6: Link prediction results (Hits@n and Mean Reciprocal Rank, filtered setting) on YAGO3-10. In bold the best results comparing both original and cardinality loss, and highlighted is the best value per evaluation metric across all models

Dataset	Sampling	Hits@1	Hits@3	Hits@5	Hits@10	MRR
FB13	Uniform	25.84	31.85	34.19	37.26	29.89
	Importance	25.17	31.36	34.36	36.18	29.18
	Bernoulli	25.92	31.86	34.11	37.18	29.97
WN18	Uniform	88.98	93.66	94.84	95.98	92.12
	Importance	88.97	93.64	94.73	96.08	91.10
	Bernoulli	89.05	93.57	94.67	95.94	91.09
WN18RR	Uniform	41.27	46.57	48.58	51.51	44.87
	Importance	41.09	46.68	48.81	51.50	44.78
	Bernoulli	41.54	46.79	48.68	51.42	45.04
YAGO3-10	Uniform	8.32	15.52	20.92	29.29	15.30
	Importance	8.23	15.71	20.70	29.49	15.28
	Bernoulli	8.48	15.74	20.82	29.50	15.42

Table 7: Link prediction results (Hits@n and Mean Reciprocal Rank, filtered setting) for the best ComplEx model using different sampling techniques.

results in Table 5. Differences in results for uniform sampling compared to the results in Table 5 are also attributed to the expanded hyperparameters space with more sampling sizes than previously.

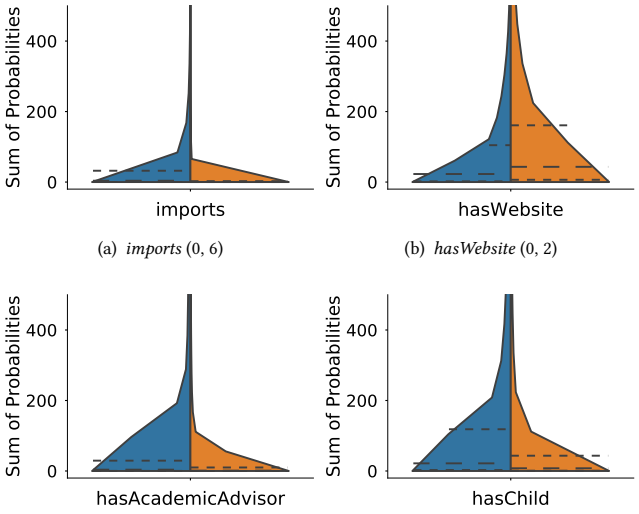
Triple	Probability
$(edgar, hasParent, eliza\_poe)$	0.861
$(edgar, hasParent, maria\_poe)$	0.854
$(edgar, hasParent, david\_poe\_jr)$	0.815

Table 8: Predictions with probability > 0.8 for $(edgar\_allan\_poe, hasParent, ?)$ by DistMult when imposing the cardinality regulariser.

Cardinality Violations in KGs. We have shown that our regulariser is beneficial for the link prediction task, but, more importantly, the predictions that violate the cardinality constraints are significantly reduced. Figure 2 shows the changes on the distribution of $\mathcal{X}_{hr}[\mathcal{E}]$ in four relation cases for ER-MLP in YAGO3-10—one of the most benefited settings. Figures 2(a), 2(c) and 2(d) illustrate positive impacts of the regularisation. We observed that the regulariser decreases the median and long-tail distribution above the third quartile for (almost) every relation, making predictions more accurate. For example, in relation *imports* ($\varphi = (0, 6)$) the mean of $\mathcal{X}_{hr}[\mathcal{E}]$ is reduced by 78%, meaning less violations. Conversely, the biggest negative impact was in relation *hasWebsite* ($\varphi = (0, 2)$, Fig. 2(b)), where violations were increased by 65%. Both constraint are equally restrictive over the number of objects but they differ on their range. For the former, the objects are entities with links to other entities, while in the latter objects are literals (URLs) with no further links. The prediction of literals is a known problem for neural link predictors as there are not many links to other entities [12].

Following the DistMult example using the constraint $\varphi_{hasParent} = (0, 2)$, Table 8 shows the predictions for parents of *Edgar Allan Poe*. There are less predictions with high probability and a correct, but previously missing, entity *David Poe Jr.* is now scored with a high probability proving the effectiveness of regularisation.

We did not note any major difference in results between tight and loose cardinality bounds, or between constraints for relations with few and many instances. Finally, Fig. 3 shows the effects of using different regularisation weights $\lambda \in \{0, 0.0001, 0.001, 0.01, 0.1, 1.0\}$ over the values of average mean of $\mathcal{X}_{hr}[\mathcal{E}]$ and Hits@10



(a) *imports* (0, 6) (b) *hasWebsite* (0, 2)

(c) *hasAcademicAdvisor* (0, 4) (d) *hasChild* (0, 19)

Figure 2: Changes in the distribution of $\mathcal{X}_{hr}[\mathcal{E}]$ without (left, in blue) and with (right, in orange) regularisation using ER-MLP in YAGO3-10. Horizontal lines correspond to quartiles.

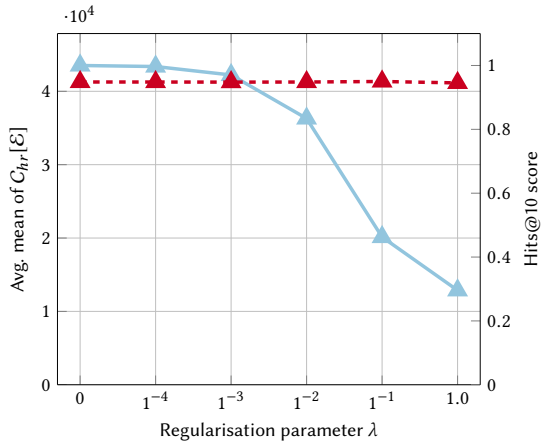


Figure 3: Influence of the regularisation weight over the average mean of $\mathcal{X}_{hr}[\mathcal{E}]$ (solid blue line) and Hits@10 (dashed red line) in WN18 with ComplEx.

across relations in WN18RR. As λ grows, Hits@10 suffers small changes and the average mean of $\mathcal{X}_{hr}[\mathcal{E}]$ decreases. This shows that the regularisation term does not affect negatively Hits@10 (a common evaluation metric) and helps to decrease the number of violations to the cardinality constraints.

6 RELATED WORK

Early works in neural link prediction (e.g., TransE [4], RESCAL [28], DistMult [39]) learn the representations of all entities and relations in the knowledge base by fitting simple scoring functions on the triples in the knowledge graph.

Recently, research focused on either (i) generating more elaborated scoring functions that better capture the nature of each of the relations, or (ii) improving existing models with background knowledge [36]. The former includes HolE [27], where the scoring function is inspired by cognitive models of associative memory; ComplEx [35] that uses complex-valued embeddings to model asymmetric relations; and ConvE [7] that builds a multi-layer convolutional network. The latter is characterised by the incorporation of additional information such as entity types, relation paths, and logical rules. We refer the readers to [26, 36] for a deeper review of neural link predictors.

Our work aligns with the second category that focuses on adding background knowledge. Almost every paper incorporating background knowledge agree that such prior knowledge improves link prediction models [8, 14, 15, 20–22]. However, none of them has considered integrity constraints such as cardinality.

Muñoz and Nickles mine cardinality constraints from knowledge graphs, and suggest their use to improve the accuracy of link prediction models.

In a similar vein, Galárraga et al. use fine-grain cardinality information to prune 'unnecessary' predictions. However, this is done only after the predictions are generated. In [40], a single cardinality bound (one-to-one, one-to-many or many-to-many) is imposed in link prediction over single-relational graphs (such as organisational charts), which differs from the multi-relational nature of knowledge graphs.

7 CONCLUSIONS

In this paper, we presented a cardinality-based regularisation term for neural link prediction models. The regulariser incorporates background knowledge in the form of relation cardinality constraints that hitherto have been ignored by neural link predictors.

The incorporation of this regularisation term in the loss function significantly reduces the number of violations produced by models at prediction time, enforcing the number of predicted triples with high probability for each relation to satisfy cardinality bounds.

Experimental results show that the regulariser consistently improves the quality of the knowledge graph embeddings, without affecting the efficiency or scalability of the learning algorithms.

ACKNOWLEDGMENTS

This work was partially supported by the TOMOE project funded by Fujitsu Laboratories Ltd., Japan and Insight Centre for Data Analytics at National University of Ireland Galway (supported by the Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289).

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *OSDI*. USENIX Association, 265–283.

[2] Kurt D. Bollacker, Robert P. Cook, and Patrick Tufts. 2007. Freebase: A Shared Database of Structured General Human Knowledge. In *AAAI*. AAAI Press, 1962–1963.

[3] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data - Application to word-sense disambiguation. *Machine Learning* 94, 2 (2014), 233–259.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.

[5] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning Structured Embeddings of Knowledge Bases. In *AAAI*. AAAI Press.

[6] Aleksandar Botev, Bowen Zheng, and David Barber. 2017. Complementary Sum Sampling for Likelihood Approximation in Large Scale Classification. In *AISTATS (Proceedings of Machine Learning Research)*, Vol. 54. PMLR, 1030–1038.

[7] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*. AAAI Press.

[8] Boyang Ding, Quan Wang, Bin Wang, and Li Guo. 2018. Improving Knowledge Graph Embedding Using Simple Constraints. In *ACL (1)*. Association for Computational Linguistics, 110–121.

[9] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *KDD*. ACM, 601–610.

[10] John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* 12 (2011), 2121–2159.

[11] Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M. Suchanek. 2017. Predicting Completeness in Knowledge Bases. In *WSDM*. ACM, 375–383.

[12] Alberto García-Durán and Mathias Niepert. 2017. KBLRN : End-to-End Learning of Knowledge Base Representations with Latent, Relational, and Numerical Features. *CoRR* abs/1709.04676 (2017).

[13] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS (JMLR Proceedings)*, Vol. 9. JMLR.org, 249–256.

[14] Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. Semantically Smooth Knowledge Graph Embedding. In *ACL (1)*. The Association for Computer Linguistics, 84–94.

[15] Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2017. SSE: Semantically Smooth Embedding for Knowledge Graphs. *IEEE Trans. Knowl. Data Eng.* 29, 4 (2017), 884–897.

[16] Katsuhiko Hayashi and Masashi Shimbo. 2017. On the Equivalence of Holographic and Complex Embeddings for Link Prediction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, Regina Barzilay et al. (Eds.). Association for Computational Linguistics, 554–559.

[17] Denis Krompaß, Maximilian Nickel, and Volker Tresp. 2014. Querying Factorized Probabilistic Triple Databases. In *International Semantic Web Conference (2) (Lecture Notes in Computer Science)*, Vol. 8797. Springer, 114–129.

[18] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *CIDR*. www.cidrdb.org.

[19] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[20] Pasquale Minervini, Luca Costabello, Emir Muñoz, Vít Novácek, and Pierre-Yves Vandenbussche. 2017. Regularizing Knowledge Graph Embeddings via Equivalence and Inversion Axioms. In *ECML/PKDD (1) (Lecture Notes in Computer Science)*, Vol. 10534. Springer, 668–683.

[21] Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. 2016. Leveraging the schema in latent factor models for knowledge graph completion. In *SAC*. ACM, 327–332.

[22] Pasquale Minervini, Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2017. Adversarial Sets for Regularising Neural Link Predictors. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017*, Gal Elidan et al. (Eds.). AUAI Press.

[23] Paramita Mirza, Simon Razniewski, Fariz Darari, and Gerhard Weikum. 2017. Cardinal Virtues: Extracting Relation Cardinalities from Text. In *ACL (2)*. Association for Computational Linguistics, 347–351.

[24] Emir Muñoz and Matthias Nickles. 2017. Mining Cardinalities from Knowledge Bases. In *DEXA (1) (Lecture Notes in Computer Science)*, Vol. 10438. Springer, 447–462.

[25] Maximilian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *NIPS*. 6341–6350.

[26] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* 104, 1 (2016), 11–33.

[27] Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016. Holographic Embeddings of Knowledge Graphs. In *AAAI*. AAAI Press, 1955–1961.

[28] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *ICML*. Omnipress, 809–816.

[29] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing YAGO: scalable machine learning for linked data. In *WWW*. ACM, 271–280.

[30] Antoni Olivé. 2007. *Conceptual modeling of information systems*. Springer.

[31] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. AUAI Press, 452–461.

[32] Herbert Robbins and Sutton Monro. 1951. A Stochastic Approximation Method. *Ann. Math. Statist.* 22, 3 (09 1951), 400–407. https://doi.org/10.1214/aoms/1177729586

[33] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *NIPS*. 926–934.

[34] Niket Tandon, Aparna S. Varde, and Gerard de Melo. 2017. Commonsense Knowledge in Machine Intelligence. *SIGMOD Record* 46, 4 (2017), 49–52.

[35] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *ICML (JMLR Workshop and Conference Proceedings)*, Vol. 48. JMLR.org, 2071–2080.

[36] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* 29, 12 (2017), 2724–2743.

[37] Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge Base Completion Using Embeddings and Rules. In *IJCAI*. AAAI Press, 1859–1866.

[38] Karen Wynn. 1990. Childrens understanding of counting. *Cognition* 36, 2 (Aug 1990), 155–193. https://doi.org/10.1016/0010-0277(90)90003-3

[39] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*.

[40] Jiawei Zhang, Jianhui Chen, Junxing Zhu, Yi Chang, and Philip S. Yu. 2017. Link Prediction with Cardinality Constraint. In *WSDM*. ACM, 121–130.