Lifted Rule Injection for Relation Embeddings

Thomas Demeester
Ghent University - iMinds
Ghent, Belgium
tdmeeste@intec.ugent.be

Tim Rocktäschel and **Sebastian Riedel**
University College London
London, UK
{t.rocktaschel,s.riedel}@cs.ucl.ac.uk

Abstract

Methods based on representation learning currently hold the state-of-the-art in many natural language processing and knowledge base inference tasks. Yet, a major challenge is how to efficiently incorporate commonsense knowledge into such models. A recent approach regularizes relation and entity representations by propositionalization of first-order logic rules. However, propositionalization does not scale beyond domains with only few entities and rules. In this paper we present a highly efficient method for incorporating implication rules into distributed representations for automated knowledge base construction. We map entity-tuple embeddings into an approximately Boolean space and encourage a partial ordering over relation embeddings based on implication rules mined from WordNet. Surprisingly, we find that the strong restriction of the entity-tuple embedding space does not hurt the expressiveness of the model and even acts as a regularizer that improves generalization. By incorporating few commonsense rules, we achieve an increase of 2 percentage points mean average precision over a matrix factorization baseline, while observing a negligible increase in runtime.

1 Introduction

Current successful methods for automated knowledge base construction tasks heavily rely on learned distributed vector representations (Nickel et al., 2012; Riedel et al., 2013; Socher et al., 2013; Chang et al., 2014; Neelakantan et al., 2015; Toutanova et al., 2015; Nickel et al., 2015; Verga et al., 2016; Verga and McCallum, 2016). Although these models are able to learn robust representations from large amounts of data, they often lack commonsense knowledge. Such knowledge is rarely explicitly stated in texts but can be found in resources like PPDB (Ganitkevitch et al., 2013) or WordNet (Miller, 1995).

Combining neural methods with symbolic commonsense knowledge, for instance in the form of implication rules, is in the focus of current research (Rocktäschel et al., 2014; Wang et al., 2014; Bowman et al., 2015; Wang et al., 2015; Vendrov et al., 2016; Hu et al., 2016; Rocktäschel and Riedel, 2016; Cohen, 2016). A recent approach (Rocktäschel et al., 2015) regularizes entity-tuple and relation embeddings via first-order logic rules. To this end, every first-order rule is propositionalized based on observed entity-tuples, and a differentiable loss term is added for every propositional rule. This approach does not scale beyond only a few entity-tuples and rules. For example, propositionalizing the rule $\forall x : \texttt{isMan}(x) \Rightarrow \texttt{isMortal}(x)$ would result in a very large number of loss terms on a large database.

In this paper, we present a method to incorporate simple rules while maintaining the computational efficiency of only modeling training facts. This is achieved by minimizing an upper bound of the loss that encourages the implication between relations to hold, entirely independent from the number of entity pairs. It only involves representations of the relations that are mentioned in rules, as well as a general rule-independent constraint on the entity-tuple embedding space. In the example given above, if we require that every component of the

vector representation of `isMan` is smaller than the corresponding component of relation `isMortal`, then we can show that the rule holds for any *non-negative* representation of an entity-tuple. Hence our method avoids the need for separate loss terms for every ground atom resulting from propositionalizing rules. In statistical relational learning this type of approach is often referred to as *lifted* inference or learning (Poole, 2003; Braz, 2007) because it deals with groups of random variables at a first-order level. In this sense our approach is a lifted form of rule injection. This allows for imposing large numbers of rules while learning distributed representations of relations and entity-tuples. Besides drastically lower computation time, an important advantage of our method over Rocktäschel et al. (2015) is that when these constraints are satisfied, the injected rules always hold, even for unseen but inferred facts. While the method presented here only deals with implications and not general first-order rules, it does not rely on the assumption of independence between relations, and is hence more generally applicable.

Our contributions are fourfold: (i) we develop a very efficient way of regularizing relation representations to incorporate first-order logic implications (§3), (ii) we reveal that, against expectation, mapping entity-tuple embeddings to non-negative space does not hurt but instead improves the generalization ability of our model (§5.1) (iii) we show improvements on a knowledge base completion task by injecting mined commonsense rules from WordNet (§5.3), and finally (iv) we give a qualitative analysis of the results, demonstrating that implication constraints are indeed satisfied in an asymmetric way and result in a substantially increased structuring of the relation embedding space (§5.6).

2 Background

In this section we revisit the matrix factorization relation extraction model by Riedel et al. (2013) and introduce the notation used throughout the paper. We choose the matrix factorization model for its simplicity as the base on which we develop implication injection.

Riedel et al. (2013) represent every relation $r \in \mathcal{R}$ (selected from Freebase (Bollacker et al., 2008) or extracted as textual surface pattern) by a k-dimensional latent representation $\boldsymbol{r} \in \mathbb{R}^k$. A particular *relation instance* or *fact* is the combination of a relation r and a tuple t of entities that are engaged in that relation, and is written as $\langle r, t\rangle$. We write $\mathcal{O}$ as the set of all such input facts available for training. Furthermore, every entity-tuple $t \in \mathcal{T}$ is represented by a latent vector $\boldsymbol{t} \in \mathbb{R}^k$ (with $\mathcal{T}$ the set of all entity-tuples in $\mathcal{O}$).

Model F by Riedel et al. (2013) measures the compatibility between a relation r and an entity-tuple t using the dot product $\boldsymbol{r}^\top \boldsymbol{t}$ of their respective vector representations. During training, the representations are learned such that valid facts receive high scores, whereas negative ones receive low scores. Typically no negative evidence is available at training time, and therefore a Bayesian Personalized Ranking (BPR) objective (Rendle et al., 2009) is used. Given a pair of facts $f_p := \langle r_p, t_p\rangle \notin \mathcal{O}$ and $f_q := \langle r_q, t_q\rangle \in \mathcal{O}$, this objective requires that

$$\boldsymbol{r_p}^\top \boldsymbol{t_p} \leq \boldsymbol{r_q}^\top \boldsymbol{t_q}. \tag{1}$$

The embeddings can be trained by minimizing a convex loss function ℓ_R that penalizes violations of that requirement when iterating over the training set. In practice, each positive training fact $\langle r, t_q\rangle$ is compared with a randomly sampled unobserved fact $\langle r, t_p\rangle$ for the same relation. The overall loss can hence be written as

$$\mathcal{L}_R = \sum_{\substack{\langle r,t_q\rangle \in \mathcal{O} \\ t_p \in \mathcal{T},\ \langle r,t_p\rangle \notin \mathcal{O}}} \ell_R\big(\boldsymbol{r}^\top[\boldsymbol{t_p} - \boldsymbol{t_q}]\big). \tag{2}$$

and measures how well observed valid facts are ranked above unobserved facts, thus reconstructing the ranking of the training data. We will henceforth call $\mathcal{L}_R$ the *reconstruction loss*, to make a distinction with the *implication loss* that we will introduce later. Riedel et al. (2013) use the logistic loss $\ell_R(s) := -\log \sigma(-s)$, where $\sigma(s) := (1 + e^{-x})^{-1}$ denotes the sigmoid function. In order to avoid overfitting, an L_2 regularization term on the $\boldsymbol{r}$ and $\boldsymbol{t}$ embeddings is added to the reconstruction loss. The overall objective to minimize hence is

$$\mathcal{L}_F = \mathcal{L}_R + \alpha\big(\textstyle\sum_{\boldsymbol{r}} \|\boldsymbol{r}\|_2^2 + \sum_{\boldsymbol{t}} \|\boldsymbol{t}\|_2^2\big) \tag{3}$$

where α is the regularization strength.

3 Lifted Injection of Implications

In this section, we show how an implication

$$\forall t \in \mathcal{T} : \langle r_p, t\rangle \Rightarrow \langle r_q, t\rangle, \tag{4}$$

can be imposed independently of the entity-tuples. For simplicity, we abbreviate such implications as $r_p \Rightarrow r_q$ (*e.g.*, `professorAt` $\Rightarrow$ `employeeAt`).

3.1 Grounded Loss Formulation

The implication rule can be imposed by requiring that every tuple $t \in \mathcal{T}$ is at least as compatible with relation r_p as with r_q. Written in terms of the latent representations, eq. (4) therefore becomes

$$\forall t \in \mathcal{T} : \boldsymbol{r_p}^\top \boldsymbol{t} \leq \boldsymbol{r_q}^\top \boldsymbol{t} \tag{5}$$

If $\langle r_p, t\rangle$ is a true fact with a high score $\boldsymbol{r_p}^\top \boldsymbol{t}$, and the fact $\langle r_q, t\rangle$ has an even higher score, it must also be true, but not vice versa. We can therefore inject an implication rule by minimizing a loss term with a separate contribution from every $t \in \mathcal{T}$, adding up to the total loss if the corresponding inequality is not satisfied. In order to make the contribution of every tuple t to that loss independent of the magnitude of the tuple embedding, we divide both sides of the above inequality by $\|\boldsymbol{t}\|_1$. With $\tilde{\boldsymbol{t}} := \boldsymbol{t}/\|\boldsymbol{t}\|_1$, the implication loss for the rule $r_p \Rightarrow r_q$ can be written as

$$\mathcal{L}_I = \sum_{\forall t \in \mathcal{T}} \ell_I\big([\boldsymbol{r_p} - \boldsymbol{r_q}]^\top \tilde{\boldsymbol{t}}\big) \tag{6}$$

for an appropriate convex loss function ℓ_I, similarly to eq. (2). In practice, the summation can be reduced to those tuples that occur in combination with r_p or r_q in the training data. Still, the propositionalization in terms of training facts leads to a heavy computational cost for imposing a single implication, similar to the technique introduced in Rocktäschel et al. (2015). Moreover, with that simplification there is no guarantee that the implication between both relations would generalize towards inferred facts not seen during training.

3.2 Lifted Loss Formulation

The problems mentioned above can be avoided if instead of $\mathcal{L}_I$, a tuple-independent upper bound is minimized. Such a bound can be constructed, provided all components of $\boldsymbol{t}$ are restricted to a non-negative embedding space, *i.e.*, $\mathcal{T} \subseteq \mathbb{R}^{k,+}$. If this holds, Jensen's inequality allows us to transform eq. (6) as follows

$$\mathcal{L}_I = \sum_{\forall t \in \mathcal{T}} \ell_I\Big(\sum_{i=1}^{k} \tilde{t}_i \,[\boldsymbol{r_p} - \boldsymbol{r_q}]^\top \mathbf{1}_i\Big) \tag{7}$$

$$\leq \sum_{i=1}^{k} \ell_I\big([\boldsymbol{r_p} - \boldsymbol{r_q}]^\top \mathbf{1}_i\big) \sum_{\forall t \in \mathcal{T}} \tilde{t}_i \tag{8}$$

where $\mathbf{1}_i$ is the unit vector along dimension i in tuple-space. This is allowed because the $\{\tilde{t}_i\}_{i=1}^k$ form convex coefficients ($\tilde{t}_i > 0$, and $\sum_i \tilde{t}_i = 1$), and ℓ_I is a convex function. If we define

$$\mathcal{L}_I^U := \sum_{i=1}^{k} \ell_I\big([\boldsymbol{r_p} - \boldsymbol{r_q}]^\top \mathbf{1}_i\big) \tag{9}$$

we can write

$$\mathcal{L}_I \leq \beta \mathcal{L}_I^U \tag{10}$$

in which β is an upper bound on $\sum_t \tilde{t}_i$. One such bound is $|\mathcal{T}|$, but others are conceivable too. In practice we rescale β to a hyper-parameter $\tilde{\beta}$ that we use to control the impact of the upper bound to the overall loss. We call $\mathcal{L}_I^U$ the *lifted loss*, as it no longer depends on any of the entity-tuples; it is grounded over the unit tuples $\mathbf{1}_i$ instead.

The implication $r_p \Rightarrow r_q$ can thus be imposed by minimizing the lifted loss $\mathcal{L}_I^U$. Note that by minimizing $\mathcal{L}_I^U$, the model is encouraged to satisfy the constraint $\boldsymbol{r_p} \leq \boldsymbol{r_q}$ on the relation embeddings, where $\leq$ denotes the component-wise comparison. In fact, a sufficient condition for eq. (5) to hold, is

$$\boldsymbol{r_p} \leq \boldsymbol{r_q} \;\text{ and }\; \forall t \in \mathcal{T} :\; \boldsymbol{t} \geq \mathbf{0} \tag{11}$$

with $\mathbf{0}$ the k-dimensional null vector. This corresponds to a single relation-specific loss term, and the general restriction $\mathcal{T} \subseteq \mathbb{R}^{k,+}$ on the tuple-embedding space.

3.3 Approximately Boolean Entity Tuples

In order to impose implications by minimizing a lifted loss $\mathcal{L}_I^U$, the tuple-embedding space needs to be restricted to $\mathbb{R}^{k,+}$. We have chosen to restrict the tuple space even more than required, namely to the hypercube $\boldsymbol{t} \in [0,1]^k$, as approximately Boolean embeddings (Kruszewski et al., 2015). The tuple

embeddings are constructed from real-valued vectors $\boldsymbol{e}$, using the component-wise sigmoid function

$$\boldsymbol{t} = \sigma(\boldsymbol{e}), \quad \boldsymbol{e} \in \mathbb{R}^k. \tag{12}$$

For minimizing the loss, the gradients are hence computed with respect to $\boldsymbol{e}$, and the L_2 regularization is applied to the components of $\boldsymbol{e}$ instead of $\boldsymbol{t}$.

Other choices for ensuring the restriction $\boldsymbol{t} \geq \boldsymbol{0}$ in eq. (11) are possible, but we found that our approach works better in practice than those (*e.g.*, the exponential transformation proposed by Demeester et al. (2016)). It can also be observed that the unit tuples over which the implication loss is grounded, form a special case of approximately Boolean embeddings.

In order to investigate the impact of this restriction even when not injecting any rules, we introduce model FS: the original model F, but with sigmoidal entity-tuples:

$$\mathcal{L}_{FS} = \sum_{\substack{\langle r,t_q\rangle \in \mathcal{O} \\ t_p \in \mathcal{T},\ \langle r,t_p\rangle \notin \mathcal{O}}} \ell_R\big(\boldsymbol{r}^\top[\sigma(\boldsymbol{e_p}) - \sigma(\boldsymbol{e_q})]\big) + \alpha\big(\textstyle\sum_{\boldsymbol{r}} \|\boldsymbol{r}\|_2^2 + \sum_{\boldsymbol{e}} \|\boldsymbol{e}\|_2^2\big) \tag{13}$$

Here, $\boldsymbol{e_p}$ and $\boldsymbol{e_q}$ are the real-valued representations as in eq. (12), for tuples t_p and t_q, respectively.

With the above choice of a non-negative tuple-embedding space we can now state the full lifted rule injection model (FSL):

$$\mathcal{L}_{FSL} = \mathcal{L}_{FS} + \tilde{\beta} \sum_{I \in \mathcal{I}} \mathcal{L}_I^U \tag{14}$$

$\mathcal{L}_I^U$ denotes a lifted loss term for every rule in a set $\mathcal{I}$ of implication rules that we want to inject.

3.4 Convex Implication Loss

The logistic loss ℓ_R (see §2) is not suited for imposing implications because once the inequality in eq. (11) is satisfied, the components of $\boldsymbol{r_p}$ and $\boldsymbol{r_q}$ do not need to be separated any further. However, with ℓ_R this would continue to happen due to the small non-zero gradient. In the reconstruction loss $\mathcal{L}_R$ this is a desirable effect which further separates the scores for positive from negative examples. However, if an implication is imposed between two relations that are almost equivalent according to the training data, we still want to find almost equivalent embedding vectors. Hence, we propose to use the loss

$$\ell_I(s) = \max(0, s + \delta) \tag{15}$$

with δ a small positive margin to ensure that the gradient does not disappear before the inequality is actually satisfied. We use $\delta = 0.01$ in all experiments.

The main advantage of the presented approach over earlier methods that impose the rules in a grounded way (Rocktäschel et al., 2015; Wang et al., 2015) is the computational efficiency of imposing the lifted loss. Evaluating $\mathcal{L}_I^U$ or its gradient for one implication rule is comparable to evaluating the reconstruction loss for one pair of training facts. In typical applications there are much fewer rules than training facts and the extra computation time needed to inject these rules is therefore negligible.

4 Related Work

Recent research on combining rules with learned vector representations has been important for new developments in the field of knowledge base completion. Rocktäschel et al. (2014) and Rocktäschel et al. (2015) provided a framework to jointly maximize the probability of observed facts and propositionalized first-order logic rules. Wang et al. (2015) demonstrated how different types of rules can be incorporated using an Integer Linear Programming approach. Wang and Cohen (2016) learned embeddings for facts and first-order logic rules using matrix factorization. Yet, all of these approaches ground the rules in the training data, limiting their scalability towards large rule sets and KBs with many entities. As argued in the introduction, this forms an important motivation for the lifted rule injection model put forward in this work, which by construction does not suffer from that limitation. Wei et al. (2015) proposed an alternative strategy to tackle the scalability problem by reasoning on a filtered subset of grounded facts.

Wu et al. (2015) proposed to use a path ranking approach for capturing long-range interactions between entities, and to add these as an extra loss term, besides the loss that models pairwise relations. Our model FSL differs substantially from their approach, in that we consider tuples instead of separate entities, and we inject a given set of rules. Yet, by cre-

ating a partial ordering in the relation embeddings as a result of injecting implication rules, model FSL can also capture interactions beyond direct relations. This will be demonstrated in §5.3 by injecting rules between surface patterns only and still measuring an improvement on predictions for structured Freebase relations.

Combining logic and distributed representations is also an active field of research outside of automated knowledge base completion. Recent advances include the work by Faruqui et al. (2014), who injected ontological knowledge from WordNet into word representations. Furthermore, Vendrov et al. (2016) proposed to enforce a partial ordering in an embeddings space of images and phrases. Our method is related to such order embeddings since we define a partial ordering on relation embeddings. However, to ensure that implications hold for all entity-tuples we also need a restriction on the entity-tuple embedding space and derive bounds on the loss. Another important contribution is the recent work by Hu et al. (2016), who proposed a framework for injecting rules into general neural network architectures, by jointly training on the actual targets and on the rule-regularized predictions provided by a teacher network. Although quite different at first sight, their work could offer a way to use our model in various neural network architectures, by integrating the proposed lifted loss into the teacher network.

This paper builds upon our previous workshop paper (Demeester et al., 2016). In that work, we tested different tuple embedding transformations in an ad-hoc manner. We used approximately Boolean representations of relations instead of entity-tuples, strongly reducing the model's degrees of freedom. We now derive the FSL model from a carefully considered mathematical transformation of the grounded loss. The FSL model only restricts the tuple embedding space, whereby relation vectors remain real valued. Furthermore, previous experiments were performed on small-scale artificial datasets, whereas we now test on a real-world relation extraction benchmark.

Finally, we explicitly discuss the main differences with respect to the strongly related work from Rocktäschel et al. (2015). Their method is more general, as they cover a wide range of first-order logic rules, whereas we only discuss implications. Lifted rule injection beyond implications will be studied in future research contributions. However, albeit less general, our model has a number of clear advantages:

Scalability – Our proposed model of lifted rule injection scales according to the number of implication rules, instead of the number of rules times the number of observed facts for every relation present in a rule.

Generalizability – Injected implications will hold even for facts not seen during training, because their validity only depends on the order relation imposed on the relation representations. This is not guaranteed when training on rules grounded in training facts by Rocktäschel et al. (2015).

Training Flexibility – Our method can be trained with various loss functions, including the rank-based loss as used in Riedel et al. (2013). This was not possible for the model of Rocktäschel et al. (2015) and already leads to an improved accuracy as seen from the zero-shot learning experiment in §5.2.

Independence Assumption – In Rocktäschel et al. (2015) an implication of the form $a_p \Rightarrow a_q$ for two ground atoms a_p and a_q is modeled by the logical equivalence $\neg(a_p \wedge \neg a_q)$, and its probability is approximated in terms of the elementary probabilities $\pi(a_p)$ and $\pi(a_q)$ as $1 - \pi(a_p)\big(1 - \pi(a_q)\big)$. This assumes the independence of the two atoms a_p and a_q, which may not hold in practice. Our approach does not rely on that assumption and also works for cases of statistical dependence. For example, the independence assumption does not hold in the trivial case where the relations r_p and r_q in the two atoms are equivalent, whereas in our model, the constraints $\boldsymbol{r}_p \leq \boldsymbol{r}_q$ and $\boldsymbol{r}_p \geq \boldsymbol{r}_q$ would simply reduce to $\boldsymbol{r}_p = \boldsymbol{r}_q$.

5 Experiments and Results

We now present our experimental results. We start by describing the experimental setup and hyperparameters. Before turning to the injection of rules, we compare model F with model FS, and show that restricting the tuple embedding space has a regularization effect, rather than limiting the expressiveness of the model (§5.1). We then demonstrate that model FSL is capable of zero-shot learning (§5.2), and show that injecting high-quality WordNet rules

Test relation	#	R13-F	F	FS	FSL
person/company	106	0.75	0.73	0.74	**0.77**
location/containedby	73	0.69	0.62	0.70	**0.71**
person/nationality	28	0.19	0.20	0.20	**0.21**
author/works_written	27	0.65	**0.71**	0.69	0.65
person/place_of_birth	21	0.72	0.69	**0.72**	0.70
parent/child	19	0.76	0.77	0.81	**0.85**
person/place_of_death	19	0.83	**0.85**	0.83	**0.85**
neighborhood/neighborhood_of	11	**0.70**	0.67	0.63	0.62
person/parents	6	0.61	0.53	**0.66**	**0.66**
company/founders	4	**0.77**	0.73	0.64	0.67
sports_team/league	4	**0.59**	0.44	0.43	0.56
team_owner/teams_owned	2	0.38	**0.64**	**0.64**	0.61
team/arena_stadium	2	**0.13**	**0.13**	**0.13**	0.12
film/directed_by	2	**0.50**	0.18	0.17	0.13
broadcast/area_served	2	0.58	0.83	0.83	**1.00**
structure/architect	2	**1.00**	**1.00**	**1.00**	**1.00**
composer/compositions	2	**0.67**	0.64	0.51	0.50
person/religion	1	**1.00**	**1.00**	**1.00**	**1.00**
film/produced_by	1	0.50	**1.00**	**1.00**	0.33
Weighted MAP		0.67	0.65	0.67	0.69

Table 1: Weighted mean average precision for our reimplementation of the matrix factorization model (F) compared to restricting the entity-pair space (FS) and injecting WordNet rules (FSL). Model F results by Riedel et al. (2013) are denoted as R13-F.

leads to an improved precision (§5.3). We proceed with a visual illustration of the relation embeddings with and without injected rules (§5.4), provide details on time efficiency of the lifted rule injection method (§5.5), and show that it correctly captures the asymmetry of implication rules (§5.6).

All models were implemented in TensorFlow (Abadi et al., 2015). We use the hyperparameters of Riedel et al. (2013), with $k = 100$ hidden dimensions and a weight of $\alpha = 0.01$ for the L_2 regularization loss. We use ADAM (Kingma and Ba, 2014) for optimization with an initial learning rate of 0.005 and a mini-batch size of 8192. The embeddings are initialized by sampling uniformly from $[-0.1, 0.1]$ and we use $\tilde{\beta} = 0.1$ for the implication loss throughout our experiments.

5.1 Restricted Embedding Space

Before incorporating external commonsense knowledge into relation representations, we were curious how much we lose by restricting the entity-tuple space to approximately Boolean embeddings. We evaluate our models on the New York Times dataset introduced by Riedel et al. (2013). Surprisingly, we find that the expressiveness of the model does not suffer from this strong restriction. From Table 1 we see that restricting the tuple-embedding space seems to perform slightly better (FS) as opposed to a real-valued tuple-embedding space (F), suggesting that this restriction has a regularization effect that improves generalization. We also provide the original results for model F by Riedel et al. (2013) (denoted as R13-F) for comparison. Due to a different implementation and optimization procedure, the results for our model F and R13-F are not identical.

Inspecting the top relations for a sampled dimension in the embedding space reveals that the relation space of model FS more closely resembles clusters than that of model F (Table 2). We hypothesize that this might be caused by approximately Boolean entity-tuple representations in model FS, resulting in attribute-like entity-tuple vectors that capture which relation clusters they belong to.

5.2 Zero-shot Learning

The zero-shot learning experiment performed in Rocktäschel et al. (2015) leads to an important finding: when injecting implications with right-hand sides for Freebase relations for which no or very limited training facts are available, the model should be able to infer the validity of Freebase facts for those relations based on rules and correlations between textual surface patterns.

We inject the same hand-picked relations as used by Rocktäschel et al. (2015), after removing all Freebase training facts. The lifted rule injection (model FSL) reaches a weighted MAP of 0.35, comparable with 0.38 by the Joint model from Rocktäschel et al. (2015) (denoted R15-Joint). Note that for this experiment we initialized the Freebase relations implied by the rules with negative random vectors (sampled uniformly from $[-7.9, -8.1]$). The reason is that without any negative training facts for these relations, their components can only go up due to the implication loss, and we do not want to get values that are too high before optimization.

Figure 1 shows how the relation extraction performance improves when more Freebase relation training facts are added. It effictively measures how well the proposed models, matrix factorization (F), propositionalized rule injection (R15-Joint), and our model (FSL), can make use of the provided rules and correlations between textual surface form pat-

Table 2: Top patterns for a randomly sampled dimension in non-restricted and restricted embedding space .

Model F (non-restricted)	Model FS (restricted)
nsubj<-represent->dobj	rcmod->return->prep->to->pobj
appos->member->prep->of->pobj->team->nn	nn<-return->prep->to->pobj
nsubj<-die->dobj	nsubj<-return->prep->to->pobj
nsubj<-speak->prep->about->pobj	rcmod->leave->dobj
appos->champion->poss	nsubj<-quit->dobj

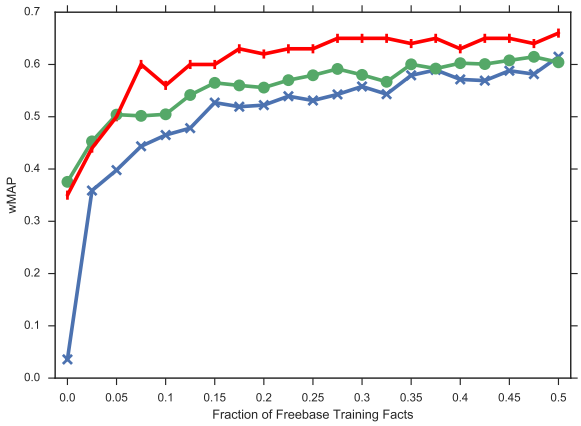


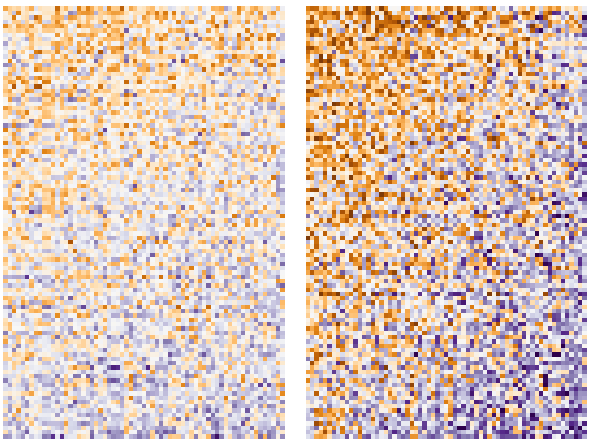
Figure 1: Weighted MAP for injecting hand-picked rules as a function of the fraction of Freebase training facts. Comparison between model F (lowest, in blue), R15-Joint (middle, in green) and model FSL (highest, in red).

terns and increased fractions of Freebase training facts. Although FSL starts at a lower performance than R15-Joint when no Freebase training facts are present, it outperforms R15-Joint and a plain matrix factorization model by a substantial margin when provided with more than 7.5% of Freebase training facts. This indicates that, in addition to being much faster than R15-Joint, it can make better use of provided rules and few training facts. We attribute this to the Bayesian personalized ranking loss instead of the logistic loss used in Rocktäschel et al. (2015). The former is compatible with our rule-injection method, but not with the approach of maximizing the expectation of propositional rules used by R15-Joint.

5.3 Injecting Knowledge from WordNet

The main purpose of this work is to be able to incorporate rules from external resources for aiding relation extraction. We use WordNet hypernyms to generate rules for the NYT dataset. To this end we iterate over all surface form patterns in the dataset and attempt to replace words in the pattern by their hypernyms. If the resulting pattern is contained in the dataset, we generate the corresponding rule. For instance, we generate a rule `appos->diplomat->amod` ⇒ `appos->official->amod` since both patterns are contained in the NYT dataset and we know from WordNet that a diplomat is an official. This leads to 427 rules from WordNet that we subsequently annotate manually to obtain 36 high-quality rules. Note that none of these rules directly imply a Freebase relation. Although the test relations all originate from Freebase, we still hope to see improvements by transitive effects, *i.e.*, better surface form representations that in turn help to predict Freebase facts.

We show results obtained by injecting these WordNet rules in Table 1 (column FSL). The weighted MAP measure increases by 2% with respect to model FS, and 4% compared to our reimplementation of the matrix factorization model F. This demonstrates that imposing a partial ordering based on implication rules can be used to incorporate logical commonsense knowledge and increase the quality of information extraction systems. Note that our evaluation setting guarantees that only indirect effects of the rules are measured, *i.e.*, we do not use any rules directly implying test relations. This shows that injecting such rules influences the relation embedding space beyond only the relations explicitly stated in the rules. For example, injecting the rule `appos<-father->appos` ⇒ `poss<-parent->appos` can contribute to improved predictions for the test relation `parent/child`.



(a) (b)

Figure 2: Visualization of embeddings (columns) for the relations that appear in the high-quality WordNet rules, (a) without and (b) with injection of these rules. Values range from -1 (orange) via 0 (white) to 1 (purple). Best viewed in color.

5.4 Visualizing Relation Embeddings

We provide a visual inspection of how the structure of the relation embedding space changes when rules are imposed. We select all relations involved in the WordNet rules, and gather them as columns in a single matrix, sorted by increasing ℓ_1 norm (values in the 100 dimensions are similarly sorted). Figures 2a and 2b show the difference between model F (without injected rules) and FSL (with rules). The values of the embeddings in model FSL are more polarized, *i.e.*, we observe stronger negative or positive components than for model F. Furthermore, FSL also reveals a clearer difference between the left-most (mostly negative, more specific) and right-most (predominantly positive, more general) embeddings (*i.e.*, a clearer separation between positive and negative values in the plot), which results from imposing the order relation in eq. (11) when injecting implications.

5.5 Efficiency of Lifted Injection of Rules

In order to get an idea of the time efficiency of injecting rules, we measure the time per epoch when restricting the program execution to a single 2.4GHz CPU core. We measure on average 6.33s per epoch without rules (model FS), against 6.76s and 6.97s when injecting the 36 high-quality WordNet rules and the unfiltered 427 rules (model FSL), respectively. Increasing the amount of injected rules from 36 to 427 leads to an increase of only 3% in computation time, even though in our setup all rule losses are used in every training batch. This confirms the high efficiency of our lifted rule injection method.

5.6 Asymmetric Character of Implications

In order to demonstrate that injecting implications conserves their asymmetric nature, we perform the following experiment. After incorporating high-quality Wordnet rules $r_p \Rightarrow r_q$ into model FSL we select all of the tuples t_p that occur with relation r_p in a training fact $\langle r_p, t_p \rangle$. Matching these with relation r_q should result in high values for the scores $\boldsymbol{r_q}^\top \boldsymbol{t_p}$, if the implication holds. If however the tuples t_q are selected from the training facts $\langle r_q, t_q \rangle$, and matched with relation r_p, the scores $\boldsymbol{r_p}^\top \boldsymbol{t_q}$ should be much lower if the inverse implication does not hold (in other words, if r_q and r_p are not equivalent). Table 3 lists the averaged results for 5 example rules, and the average over all relations in WordNet rules, both for the case with injected rules (model FSL), and without rules (model FS). For easier comparison, the scores are mapped to the unit interval via the sigmoid function. This quantity $\sigma(\boldsymbol{r}^\top \boldsymbol{t})$ is often interpreted as the probability that the corresponding fact holds (Riedel et al., 2013), but because of the BPR-based training, only differences between scores play a role here. After injecting rules, the average scores of facts inferred by these rules (*i.e.*, column $\sigma(\boldsymbol{r_q}^\top \boldsymbol{t_p})$ for model FSL) are always higher than for facts (incorrectly) inferred by the inverse rules (column $\sigma(\boldsymbol{r_p}^\top \boldsymbol{t_q})$ for model FSL). In the fourth example, the inverse rule leads to high scores as well (on average 0.79, vs. 0.98 for the actual rule). This is due to the fact that the `daily` and `newspaper` relations are more or less equivalent, such that the components of $\boldsymbol{r_p}$ are not much below those of $\boldsymbol{r_q}$. For the last example (the `ambassador` $\Rightarrow$ `diplomat` rule), the asymmetry in the implication is maintained, although the absolute scores are rather low for these two relations.

The results for model FS reflect how strongly the implications in either direction are latently present in the training data. We can only conclude that model FS manages to capture the similarity be-

rule			model FSL		model FS	
r_p	$\Rightarrow$	r_q	$\sigma(\boldsymbol{r_q^\top t_p})$	$\sigma(\boldsymbol{r_p^\top t_q})$	$\sigma(\boldsymbol{r_q^\top t_p})$	$\sigma(\boldsymbol{r_p^\top t_q})$
appos->party->amod	$\Rightarrow$	appos->organization->amod	0.99	0.22	0.70	0.86
poss<-father->appos	$\Rightarrow$	poss<-parent->appos	0.96	0.00	0.72	0.89
appos->prosecutor->nn	$\Rightarrow$	appos->lawyer->nn	0.99	0.01	0.87	0.80
appos->daily->amod	$\Rightarrow$	appos->newspaper->amod	0.98	0.79	0.90	0.86
appos->ambassador->amod	$\Rightarrow$	appos->diplomat->amod	0.31	0.05	0.93	0.84
average over 36 high-quality Wordnet rules			0.95	0.28	0.74	0.70

Table 3: Average of $\sigma(\boldsymbol{r_q^\top t})$ over all inferred facts $\langle r_q, t_p \rangle$ for tuples $\boldsymbol{t_p}$ from training items for relation r_p, and vice versa, for Wordnet implications $\boldsymbol{r_p} \Rightarrow \boldsymbol{r_q}$, and model FSL (injected rules) vs. model FS (no rules).

tween relations, but not the asymmetric character of implications. For example, purely based on the training data, it appears to be more likely that the `parent` relation implies the `father` relation, than vice versa. This again demonstrates the importance and added value of injecting external rules capturing commonsense knowledge.

6 Conclusions

We presented a novel, fast approach for incorporating first-order implication rules into distributed representations of relations. We termed our approach 'lifted rule injection', as it avoids the costly grounding of first-order implication rules and is thus independent of the size of the domain of entities. By construction, these rules are satisfied for any observed or unobserved fact. The presented approach requires a restriction on the entity-tuple embedding space. However, experiments on a real-world dataset show that this does not impair the expressiveness of the learned representations. On the contrary, it appears to have a beneficial regularization effect.

By incorporating rules generated from WordNet hypernyms, our model improved over a matrix factorization baseline for knowledge base completion. Especially for domains where annotation is costly and only small amounts of training facts are available, our approach provides a way to leverage external knowledge sources for inferring facts.

In future work, we want to extend the proposed ideas beyond implications towards general first-order logic rules. We believe that supporting conjunctions, disjunctions and negations would enable to debug and improve representation learning based knowledge base completion. Furthermore, we want to integrate these ideas into neural methods beyond matrix factorization approaches.

Acknowledgments

This work was supported by the Research Foundation - Flanders (FWO), Ghent University - iMinds, Microsoft Research through its PhD Scholarship Programme, an Allen Distinguished Investigator Award, and a Marie Curie Career Integration Award.

References

[Abadi et al.2015] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Bollacker et al.2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

[Bowman et al.2015] Samuel R Bowman, Christopher Potts, and Christopher D Manning. 2015. Recursive neural networks can learn logical semantics. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*.

[Braz2007] Rodrigo De Salvo Braz. 2007. *Lifted First-order Probabilistic Inference*. Ph.D. thesis, Champaign, IL, USA. AAI3290183.

[Chang et al.2014] Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *EMNLP*, pages 1568–1579.

[Cohen2016] William. W. Cohen. 2016. TensorLog: A Differentiable Deductive Database. *ArXiv e-prints*, May.

[Demeester et al.2016] Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2016. Regularizing relation representations by first-order implications. In *NAACL Workshop on Automated Knowledge Base Construction (AKBC)*.

[Faruqui et al.2014] Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

[Ganitkevitch et al.2013] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 758–764.

[Hu et al.2016] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318*.

[Kingma and Ba2014] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kruszewski et al.2015] German Kruszewski, Denis Paperno, and Marco Baroni. 2015. Deriving boolean structures from distributional vectors. *Transactions of the Association for Computational Linguistics*, 3:375–388.

[Miller1995] George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

[Neelakantan et al.2015] Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. *arXiv preprint arXiv:1504.06662*.

[Nickel et al.2012] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280. ACM.

[Nickel et al.2015] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *arXiv preprint arXiv:1503.00759*.

[Poole2003] David Poole. 2003. First-order probabilistic inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 985–991, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Rendle et al.2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 452–461, Arlington, Virginia, United States. AUAI Press.

[Riedel et al.2013] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 74–84.

[Rocktäschel and Riedel2016] Tim Rocktäschel and Sebastian Riedel. 2016. Learning knowledge base inference with neural theorem provers. In *NAACL Workshop on Automated Knowledge Base Construction (AKBC)*.

[Rocktäschel et al.2014] Tim Rocktäschel, Matko Bosnjak, Sameer Singh, and Sebastian Riedel. 2014. Low-dimensional embeddings of logic. In *ACL Workshop on Semantic Parsing*.

[Rocktäschel et al.2015] Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. Injecting Logical Background Knowledge into Embeddings for Relation Extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

[Socher et al.2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems (NIPS)*.

[Toutanova et al.2015] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *EMNLP*.

[Vendrov et al.2016] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-embeddings of images and language. *arXiv preprint*, abs/1511.06361.

[Verga and McCallum2016] Patrick Verga and Andrew McCallum. 2016. Row-less universal schema. In *NAACL Workshop on Automated Knowledge Base Construction (AKBC)*.

[Verga et al.2016] Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016. Multilingual relation extraction using compositional universal schema. In *Annual Conference of the*

North American Chapter of the Association for Computational Linguistics (HLT-NAACL), pages 886–896. ACL.

[Wang and Cohen2016] William Yang Wang and William W. Cohen. 2016. Learning first-order logic embeddings via matrix factorization. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, New York, NY, July. AAAI.

[Wang et al.2014] William Yang Wang, Kathryn Mazaitis, and William W Cohen. 2014. Structure learning via parameter learning. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1199–1208. ACM.

[Wang et al.2015] Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI)*, pages 1859–1865. AAAI Press.

[Wei et al.2015] Zhuoyu Wei, Jun Zhao, Kang Liu, Zhenyu Qi, Zhengya Sun, and Guanhua Tian. 2015. Large-scale knowledge base completion: Inferring via grounding network sampling over selected instances. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*, pages 1331–1340. ACM.

[Wu et al.2015] Fei Wu, Jun Song, Yi Yang, Xi Li, Zhongfei Zhang, and Yueting Zhuang. 2015. Structured embedding via pairwise relations and long-range interactions in knowledge base. In *AAAI Conference on Artificial Intelligence*, pages 1663–1670.