Regularizing Relation Representations by First-order Implications

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 for automated knowledge base construction often rely on trained fixed-length vector representations of relations and entities to predict facts. Recent work showed that such representations can be regularized to inject first-order logic formulae. This enables to incorporate domain-knowledge for improved prediction of facts, especially for uncommon relations. However, current approaches rely on propositionalization of formulae and thus do not scale to large sets of formulae or knowledge bases with many facts. Here we propose a method that imposes first-order constraints directly on relation representations, avoiding costly grounding of formulae. We show that our approach works well for implications between pairs of relations on artificial datasets.

1 Introduction

Many methods for automated knowledge base (KB) construction rely on learned relation and entity vector representations (Nickel et al., 2015). Such representations are hard to learn for relations with only few supporting facts in KBs. Moreover, inference on KBs such as Freebase (Bollacker et al., 2008) could still benefit from common-sense knowledge contained in ontologies like WordNet (Miller, 1995) or PPDB (Ganitkevitch et al., 2013). It is thus desirable to be able to use various kinds of domain or ontological knowledge, for instance in the form of first-order logic formulae, to help knowledge base inference. Furthermore, such formulae make use of learned representations as well as help to learn better representations.

One way to incorporate logical formulae is to regularize relation and entity-pair representations (Rocktäschel et al., 2015). However, in their method first-order formulae need to be grounded for all entity pairs in the KB. As a result of this propositionalization, the method does not scale to large KBs or many formulae. Another recent method is based on imposing rules as constraints in an integer linear program (Wang et al., 2015). This approach suffers from a similar scalability problem, since every rule is imposed for all occurrences of facts in the training data.

To alleviate this computational bottleneck, we propose a method to incorporate first-order implications directly (and only) into relation representations. The idea is to map relation and entity-pair representations into a well-chosen subspace in which formulae can be expressed as direct regularizers of relation representations without imposing them on entity representations too. As such, the proposed method is suited for problems with large numbers of rules and facts.

Our approach is based on the concept of order-embeddings, introduced by Vendrov et al. (2016). Order-embeddings capture partial orderings, such as textual entailment, directly in vector representations. This idea can be extended towards relation representations in KBs. In particular, we show how to construct order-embeddings for capturing implications between relations, such that these implications hold for any possible entity-pair.

The model presented here is also related to Kruszewski et al. (2015). They demonstrate that textual entailment can be captured by mapping real-valued vectors into (approximate) Boolean valued vectors. This is achieved by requiring that Boolean vector representations of more specific words or sentences are included in the representation of more general ones. Furthermore, these representations may be useful for modeling other types of logical relationships, such as negation or conjunction. It is our goal to extend the approach towards arbitrary first-order formulae between relations. Therefore, as a first step we investigate whether restricting the relation embedding space to approximate Boolean vectors still allows us to reconstruct training facts and imposed implications.

The rest of the paper is organized as follows. We first revisit matrix factorization for KB construction (§2), before introducing a factorization model that regularizes approximately Boolean relation representations to incorporate first-order implications (§3). Finally, we show empirical results on synthetic knowledge bases. We explore how enforcing restrictions on representations influences the ability to model the observed data, analyze the learned relation representations qualitatively, and investigate the impact of injecting implications (§4).

2 Model

Before introducing first-order regularization of relation representations, we revisit one possible model that uses relation (and entity-pair) representations to estimate the probability of a fact: the universal schema matrix factorization proposed by Riedel et al. (2013). Let $\mathcal{R}$ be a set of relations r and $\mathcal{P}$ a set of entity pairs (e_i, e_j) (which we will shortly write as e from now on). We can represent facts, *i.e.*, possible combinations of entity pairs and relations, as a binary matrix of size $|\mathcal{P}| \times |\mathcal{R}|$. The probability that a particular relation and entity pair combination is a valid fact can be modeled by the sigmoid of the dot product of the relation's vector representation $\boldsymbol{v}(r)$ and the entity-pair's vector representation $\boldsymbol{v}(e)$:

$$p\big(z = 1|\boldsymbol{v}(r), \boldsymbol{v}(e)\big) = \sigma\big(\boldsymbol{v}(r)^T \boldsymbol{v}(e)\big), \quad (1)$$

with the binary target variable z indicating validity of the considered fact and $\boldsymbol{v}(r), \boldsymbol{v}(e) \in \mathbb{R}^k$. The representations $\boldsymbol{v}(r)$ and $\boldsymbol{v}(e)$ can be found by minimizing the negative log-likelihood of true given training facts (together with a set of negative facts) using stochastic gradient descent. The contribution to this loss from relation r and entity pair e takes the following form

$$\mathcal{L}_F(r, e) = -z \log(p) - (1 - z) \log(1 - p) \quad (2)$$

with p short-hand for the probability in eq. (1).

In this paper we propose various forms of $\boldsymbol{v}(r)$ and $\boldsymbol{v}(e)$. However, when the representations are chosen to be unrestricted real-valued vectors, i.e., $\boldsymbol{v}(r) = \boldsymbol{\rho} \in \mathbb{R}^k$ and $\boldsymbol{v}(e) = \boldsymbol{e} \in \mathbb{R}^k$ for some fixed embedding length k, we get the latent feature **Model F** by Riedel et al. (2013).

Note that often no explicit negative instances are available for training, in which case unobserved facts can be randomly sampled and assumed to be negative.

2.1 Non-Negative Embedding Space

With the model described above we do not have any control over the learned representations. However, the embeddings can gain useful properties once we restrict them in an appropriate way. We propose the following restrictions, motivated below: we require all components of $\boldsymbol{v}(e)$ to be non-negative, and we confine relation representations $\boldsymbol{v}(r)$ to lie within the unit hypercube $(0, 1)^k$.

We want to be able to model implications between relations by defining an order relation on their vector representations. An in-depth description of order-embeddings is given in Vendrov et al. (2016), but the main idea applied to relation representations is as follows. Consider a pair of relations r_p and r_q such that r_p implies r_q for any entity pair for which r_p holds (which we shortly write as '$r_p \Rightarrow r_q$'). For their vector representations we require that the component-wise inequality $v_i(r_p) \leq v_i(r_q)$ holds $(i = 1, \ldots, k)$. Note that enforcing this locally for every relation pair will also lead to globally consistent relation representations (*e.g.* imposing $r_s \Rightarrow r_t$ and $r_t \Rightarrow r_u$ will satisfy $r_s \Rightarrow r_u$ by construction). Relations that hold true more often will have larger entries, whereas relation vectors with the overall lowest values will represent the most specific relations (such as leaf nodes in an ontology).

If $r_p \Rightarrow r_q$ holds, it needs to hold for any entity pair e. Thus, we require that $\forall e \in \mathcal{P}$:

$$p\big(z = 1|\boldsymbol{v}(r_p), \boldsymbol{v}(e)\big) \leq p\big(z = 1|\boldsymbol{v}(r_q), \boldsymbol{v}(e)\big).$$

If $v_i(r_p) \leq v_i(r_q)$ $(i = 1, \ldots, k)$, and we restrict all components of $\boldsymbol{v}(e)$ to be non-negative, then by construction $\boldsymbol{v}(r_p)^T\boldsymbol{v}(e) \leq \boldsymbol{v}(r_q)^T\boldsymbol{v}(e)$, and with eq. (1), the above requirement is satisfied.

Besides the ability to capture pairwise implications, we also want to incorporate more complex first-order formulae and need to be able to express these as a function of the relation and entity-pair representations. Approximate Boolean vectors discussed in Kruszewski et al. (2015) provide an attractive direction, but studying how they can be adapted to suit the relation extraction use case is out of scope of the current work. To pave the way for future work on incorporating arbitrary first-order constraints, we will however investigate whether constraining relation representations to the unit hypercube $\boldsymbol{v}(r) \in (0,1)^k$ still allows us to reliably encode observed facts and impose implications.

2.2 Training Restricted Representations

There are different ways to impose the discussed restrictions on vector representations. In this work, we choose $\boldsymbol{v}(r) = \sigma(\boldsymbol{\rho})$, and $\boldsymbol{v}(e) = \mathrm{ReLU}(\boldsymbol{e})$ or $\exp(\boldsymbol{e})$, where $\mathrm{ReLU}(\boldsymbol{e}) = \log(1 + \exp \boldsymbol{e})$ is the component-wise smooth approximation of the rectified linear unit, and with again $\boldsymbol{\rho} \in \mathbb{R}^k$ and $\boldsymbol{e} \in \mathbb{R}^k$. The imposed restrictions constrain the set of usable loss functions for training. Indeed, the lowest value of $\sigma\big(\boldsymbol{v}(r)^T\boldsymbol{v}(e)\big)$ is 0.5, which makes training with the loss function in eq. (2) no longer practical. The problem can be avoided if the dot product $\boldsymbol{v}(r)^T\boldsymbol{v}(e)$ is first mapped from the positive real axis to entire $\mathbb{R}$. Among various options, we choose the logarithm because

$$\sigma\Big(\log\big(\boldsymbol{v}(r)^T\boldsymbol{v}(e)\big)\Big) = \frac{\boldsymbol{v}(r)^T\boldsymbol{v}(e)}{1 + \boldsymbol{v}(r)^T\boldsymbol{v}(e)}, \qquad (3)$$

such that the loss from eq. (2) simplifies to

$$\begin{aligned}\mathcal{L}_F(r,e) = &-z\log\big(\boldsymbol{v}(r)^T\boldsymbol{v}(e)\big)\\ &+ \log\big(1 + \boldsymbol{v}(r)^T\boldsymbol{v}(e)\big). \qquad (4)\end{aligned}$$

The expression on the right-hand side of eq. (3) represents an alternative form of the probability in eq. (1) for training and predicting the validity of facts using non-negative embeddings. Note that since log and exp are inverse functions, choosing $\boldsymbol{v}(e) = \exp(\boldsymbol{e})$ leads to values of $\log\big(\boldsymbol{v}(r)^T\boldsymbol{v}(e)\big)$ with the same order of magnitude as $\boldsymbol{e}$, unlike the choice $\boldsymbol{v}(e) = \mathrm{ReLU}(\boldsymbol{e})$. This may be the reason why the former seems to work better in practice (see § 3). Yet another option would be to construct an approximate Boolean factorization for both, entity pairs and relations, whereby $\boldsymbol{v}(r) = \sigma(\boldsymbol{\rho})$ and $\boldsymbol{v}(e) = \sigma(\boldsymbol{e})$. Finding a suitable loss function is less straightforward, but we tested the quadratic loss on $\boldsymbol{v}(r)^T\boldsymbol{v}(e)$. As shown in the following section, this additional restriction reduces the ability of the model to reconstruct facts.

2.3 Implication Regularization

We will refer to the loss term $\mathcal{L}_F$ introduced above as the *fact loss*, as it measures how well training facts are recovered with low-dimensional representations. To impose logical constraints, we add an additional loss term per rule which we will call the *implication loss* $\mathcal{L}_I$. As already described, the required order relation between two relations can be expressed by their representations as $\bigwedge_{i=1}^{k} v_i(r_p) \leq v_i(r_q)$. We thus propose the following loss term for every implication $r_p \Rightarrow r_q$,

$$\mathcal{L}_I^{r_p \Rightarrow r_q} = \sum_{i=1}^{k} \log\big(1 + \mathrm{ReLU}(\rho_{p,i} - \rho_{q,i})\big). \qquad (5)$$

As before, other choices are possible. It is however essential to ensure that only positive values of $\rho_{p,i} - \rho_{q,i}$ are penalized, which is obtained by applying the ReLU function (see § 2.2). The difficulty in choosing an appropriate loss function is that its behavior needs to be compatible with the fact loss. For instance, the simple loss $\mathrm{ReLU}(\rho_{p,i} - \rho_{q,i})$ seems not to work in practice as balancing both losses during optimization becomes difficult. The particular form of $\mathcal{L}_I$ in eq. (5) was obtained in a similar way to eq. (4), and originates from simplifying

$$-\sum_{i=1}^{k} \log\Big(1 - \sigma\big(\log \mathrm{ReLU}(\rho_{p,i} - \rho_{q,i})\big)\Big).$$

We empirically found that this loss works well in practice and behaves in an intuitive way. For example, injecting the formulae $r_p \Rightarrow r_q$ and $r_q \Rightarrow r_p$ leads to roughly identical representations for both relations.

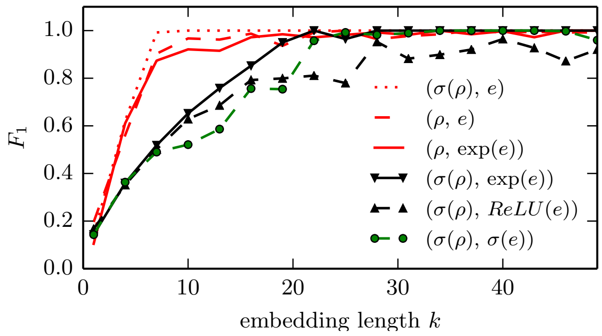

Figure 1: Ability of various methods $\big(\boldsymbol{v}(r), \boldsymbol{v}(e)\big)$ to reconstruct binary matrices, on a sampled KB with 50 entities (249 observed entity pairs) and 20 relations.

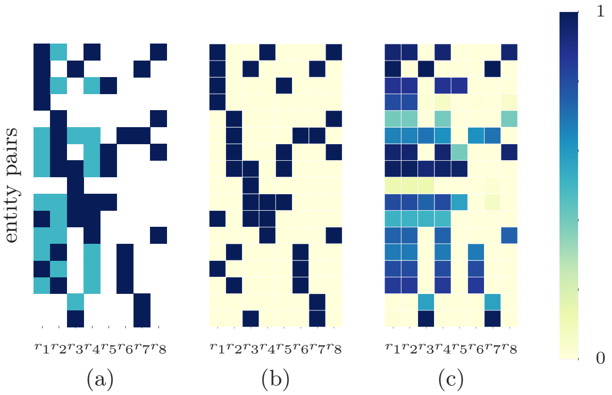

Figure 2: Toy example with 8 relations, 17 observed entity pairs, and 5 implication rules $(r_4 \Rightarrow r_1)$, $(r_7 \Rightarrow r_3)$, $(r_4 \Rightarrow r_2)$, $(r_6 \Rightarrow r_4)$, and $(r_5 \Rightarrow r_4)$. (a) Original knowledge base (dark blue: known facts; white: unknown facts; light blue: inferred facts from rules); (b) reconstructed with embedding size 15 with $\mathcal{L} = \mathcal{L}_F$, and (c) with $\mathcal{L} = \mathcal{L}_F + \mathcal{L}_I$.

3 Experiments

To gain insights into the proposed models, we investigate their behavior on small-scale artificial KB inference datasets that we can adapt to different possible scenarios. Concretely, we sample facts for a predefined number of entities and relations. Then, we generate implications for sampled pairs of relations and add a fraction of implied facts to the training data and the rest to a test set. This gives us control over how much an implication is visible for training representations of facts in the KB.

Fact Reconstruction in Non-Negative Space We first investigate whether restricting embedding spaces still allows to reconstruct observed facts. To this end, we consider a dataset with 20 relations and 50 entities, leading to observations for 249 entity pairs. We calculate the F_1 score for reconstructing all training facts, assuming that all unobserved facts are negative. Fig. 1 shows the result for different combinations of restricting the relation and entity pair embedding spaces. Every model maps a relation r and entity-pair e into vector space, denoted by $\big(\boldsymbol{v}(r), \boldsymbol{v}(e)\big)$ where $\boldsymbol{\rho}$ and $\boldsymbol{e}$ represent the learned real-valued (i.e., non-restricted) representations before mapping into a non-negative subspace. The results are shown as a function of the embedding size k. We found that from the two models that satisfy both the relation and the entity pair restriction, the one with $\boldsymbol{v}(e) = \exp(\boldsymbol{e})$ seems to work best and will be used in the remainder of the experiments. As expected, imposing restrictions leads to a reduced ability to fit the data exactly and hence requires higher-dimensional vector representations of relations and entity-pairs.

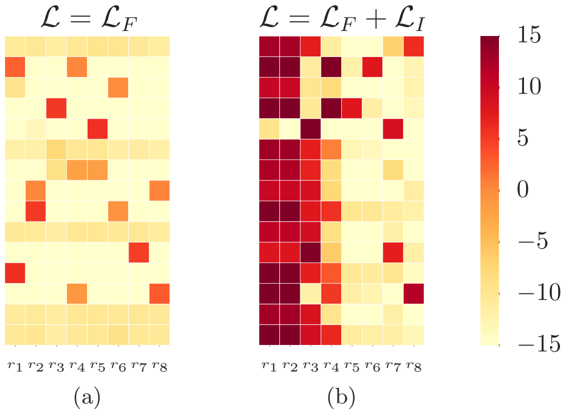

Figure 3: The columns are the 15-dimensional representations $\boldsymbol{\rho}_1$ to $\boldsymbol{\rho}_8$ for the relations r_1 to r_8 in the toy example of Fig. 2. (a) Only fact loss $\mathcal{L}_F$ applied; (b) including implication loss $\mathcal{L}_I$.

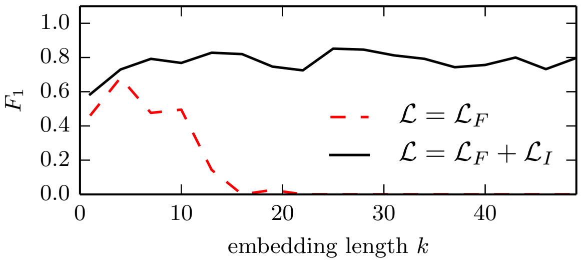

Figure 4: Ability of correctly predicting unseen facts implied by observed facts on a dataset with 20 relations, 249 observed entity pairs, and 10 pairwise implications with 50% evidence for the observed facts, with the model $\big(\sigma(\boldsymbol{\rho}),\, \exp(\boldsymbol{e})\big)$.

Implication Regularization To visualize what happens when regularizing relation representations based on given implications, we sample a small KB with 8 relations, 17 entity pair observations and the following five implications: $r_4 \Rightarrow r_1$, $r_7 \Rightarrow r_3$, $r_4 \Rightarrow r_2$, $r_6 \Rightarrow r_4$ and $r_5 \Rightarrow r_4$. We add 20% of the facts that can be inferred from these rules as training data and use the rest as test data.

Fig. 2(a) shows observed facts (dark blue), as well as test facts (light blue). With an embedding size of 15, $\mathcal{L}_F$ is able to perfectly reconstruct the training data, as shown in Fig. 2(b), but therefore overfits. In contrast, when imposing implications we can reconstruct training facts and predict test facts that could be inferred by these implications (Fig. 2(c)). Note that in Fig. 2(b) the predictions are made with high confidence, whereas in Fig. 2(c) the reconstruction is not perfect, with the predictions distributed between 0 and 1. This is due to the fact that during training the loss related to some of the facts is influenced both by the implication loss and by a conflicting contribution from the fact loss (due to the random sampling of negative examples among the unobserved ones). Although this effect is an artifact of the small scale of the example (where non-observed facts are sampled more often than in a large and sparse situations), it underlines the importance of properly weighting both loss terms, for which further research on large-scale data is needed.

The learned relation embeddings are visualized in Fig. 3. We can see that regularizing relation embeddings by implications leads to representations that satisfy the order imposed by the implications (see Fig. 3(b)).

For the final experiment, we again consider the dataset used for Fig. 1, but this time we inject 10 pairwise implications and add half of the additional facts that can be inferred from them to the training set. The others are added to the test set, together with as many sampled negative test facts. The F_1 value on the test facts for different embedding sizes is shown in Fig. 4. We found that the implication loss successfully acts as a regularizer, yielding F_1 scores of around 80% for predicting unobserved valid facts even with large embedding sizes where a model without this regularization drastically overfits.

4 Conclusion and Future Work

We have presented a scalable method to incorporate first-order implications into relation representations for knowledge base inference. It alleviates the need for propositionalization of such formulae and we plan to use it to improve large-scale knowledge base inference with many formulae extracted from ontologies. We discussed and illustrated the method in a matrix factorization setting, but it can be applied to any model that produces relation and entity (or entity-pair) representations that can be mapped into non-negative space. In future work, we will investigate ways to efficiently incorporate more complex formulae as well, involving conjunctions, disjunctions, and negations.

Acknowledgments

We thank Sameer Singh and Dirk Weissenborn for fruitful discussions, and the reviewers as well as Johannes Welbl for comments on drafts of this paper. 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

[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.

[Ganitkevitch et al.2013] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *HLT-NAACL*, pages 758–764.

[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.

[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*.

[Riedel et al.2013] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas.

[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 (NAACL)*.

[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.

[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'15, pages 1859–1865. AAAI Press.