Injecting Logical Background Knowledge into Embeddings for Relation Extraction

Tim Rocktäschel
University College London
London, UK

Sameer Singh
University of Washington
Seattle, WA

Sebastian Riedel
University College London
London, UK

Abstract

Matrix factorization approaches to relation extraction provide several attractive features: they support distant supervision, handle open schemas, and leverage unlabeled data. Unfortunately, these methods share a shortcoming with all other distantly supervised approaches: they cannot learn to extract target relations without existing data in the knowledge base, and likewise, these models are inaccurate for relations with sparse data. Rule-based extractors, on the other hand, can be easily extended to novel relations and improved for existing but inaccurate relations, through first-order formulae that capture auxiliary domain knowledge. However, usually a large set of such formulae is necessary to achieve generalization.

In this paper, we introduce a paradigm for learning low-dimensional embeddings of entity-pairs and relations that combine the advantages of matrix factorization with first-order logic domain knowledge. We introduce simple approaches for estimating such embeddings, as well as a novel training algorithm to jointly optimize over factual and first-order logic information. Our results show that this method is able to learn accurate extractors with little or no distant supervision alignments, while at the same time generalizing to textual patterns that do not appear in the formulae.

1 Introduction

Relation extraction, the task of identifying relations between named entities, is a crucial component for information extraction. A recent successful approach (Riedel et al., 2013) relies on two ideas: (a) unifying traditional canonical relations, such as those of the Freebase schema, with OpenIE surface form patterns in a universal schema, and (b) completing a knowledge base of such a schema using matrix factorization. This approach has several attractive properties. First, for canonical relations it effectively performs distant supervision (Bunescu and Mooney, 2007; Mintz et al., 2009; Yao et al., 2011; Hoffmann et al., 2011; Surdeanu et al., 2012) and hence requires no textual annotations. Second, in the spirit of OpenIE, a universal schema can use textual patterns as novel relations and hence increases the coverage of traditional schemas (Riedel et al., 2013; Fan et al., 2014). Third, matrix factorization learns better embeddings for entity-pairs for which only surface form patterns are observed, and these can also lead to better extractions of canonical relations.

Unfortunately, populating a universal schema knowledge base using matrix factorization suffers from a problem all distantly-supervised techniques share: you can only reliably learn relations that appear frequently enough in the knowledge base. In particular, for relations that do not appear in the knowledge base or for which no facts are known we cannot learn a predictor at all. One way to overcome this problem is to incorporate additional domain knowledge, either specified manually or bootstrapped from auxiliary sources. In fact, domain knowledge encoded as simple logic formulae over patterns and relations has been used in practice to directly specify relation extractors (Reiss et al., 2008; Chiticariu et al., 2013; Akbik et al., 2014). However, these extractors can be brittle and obtain poor recall, since they are unable to generalize to textual patterns that are not
In this paper, we propose a paradigm for learning universal schema extractors by combining matrix factorization based relation extraction with additional information in the form of first-order logic knowledge. Our contributions are threefold: (i) We introduce simple baselines that enforce logic constraints through deterministic inference before and after matrix factorization (§3.1). (ii) We propose a novel joint training algorithm that learns vector embeddings of relations and entity-pairs using both distant supervision and first-order logic formulae such that the factorization captures these formulae (§3.2). (iii) We present an empirical evaluation using automatically mined rules that demonstrates the benefits of incorporating logical knowledge in relation extraction, in particular that joint factorization of distant and logic supervision is efficient, accurate, and robust to noise (§5).

2 Matrix Factorization and Logic

In this section we provide background on matrix factorization for universal schema relation extraction, and describe its connections to first-order logic.

2.1 Notation

In order to later unify observed facts and logical background knowledge, we first represent given factual data in terms of first-order logic. We have a set $E$ of constants that refer to entities, and a set of predicates $R$ that refer to relations between these entities. In the following we will focus on binary relations in a universal schema that contains both structured relations from one (or more) knowledge bases, and surface-form relations. Further, with $P \subseteq E \times E$ we denote the domain over entity-pairs of interest.

In function-free first-order logic a term is defined as a constant or a variable, and the most basic form of a formula is an atom such as $\text{professorAt}(x, y)$ that applies a predicate to a pair of terms. More complex formulae such as $\forall x, y : \text{professorAt}(x, y) \Rightarrow \text{employeeAt}(x, y)$ can be constructed by combining atoms with logical connectives (such as $\neg$ and $\land$) and quantifiers ($\exists x, \forall x$).

The simplest form of first-order formulae are ground atoms: predicates applied to constants, such as $\text{directorOf}(\text{NOLAN}, \text{INTERSTELLAR})$. A possible world is a set of ground atoms. Ground literals are either ground atoms or negated ground atoms such as $\neg \text{bornIn}(\text{NOLAN}, \text{BERLIN})$, and correspond to positive or negative facts. Training data for distant supervision can now be viewed as a knowledge base of such ground literals. Our goal is to extend the class offormulae from such facts to rules such as the first-order formula above.

2.2 Matrix Factorization with Ground Atoms

Given the notation presented above, matrix factorization can now be seen as a learning task in which low-dimensional embeddings are estimated for all constant pairs in $P$ and predicates (relations) in $R$, given a collection of ground atoms (facts) as supervision. We represent constant-pairs as rows and predicates as columns of a $|P| \times |R|$ binary matrix, and each atom in the training data represents an observed cell in this matrix. As introduced in Riedel et al. (2013), we seek to find a low-rank factorization into a $|P| \times k$ matrix of embeddings of constant-pairs and a $k \times |R|$ matrix of predicate embeddings such that they approximate the observed matrix.

More precisely, let $v(\cdot)$ denote the mapping from constant-pairs and predicates to their corresponding embedding. That is, $v_{r_m}$ is the embedding for predicate $r_m$, and $v(e_i, e_j)$ is the embedding for the pair of constants $(e_i, e_j)$. Let $w$ be a possible world (i.e. a set of ground atoms), and $V$ be the set of all entity-pair and relation embeddings. Further, let $\pi_{m, e_i, e_j} = \sigma(v_{r_m} \cdot v(e_i, e_j))$ where $\sigma$ is the sigmoid function and $v_{r_m}, v(e_i, e_j)$ denotes the vector dot-product between the embeddings of relation $r_m$ and entity-pair $(e_i, e_j)$. We define the conditional probability of a possible world $w$ given embeddings $V$ as

$$p(w|V) = \prod_{r_m(e_i, e_j) \in w} \pi_{m, e_i, e_j} \prod_{r_m(e_i, e_j) \notin w} (1 - \pi_{m, e_i, e_j}).$$

The embeddings can be estimated by maximizing the likelihood of a set of observed ground atoms with $\ell_2$ regularization (Collins et al., 2001), optimized using stochastic gradient descent. In summary, with atomic formulae (i.e. factual knowledge) we learn entity-pair and relation embeddings that reconstruct known facts and are able to generalize to unknown facts.
3 Injecting Logic Into Factorization

Matrix factorization is capable of learning complex dependencies between relations, but requires observed facts as training signal. However, often we either do not have this signal because the relations of interest do not have pre-existing facts, or this signal is noisy due to alignment errors or mismatches when linking knowledge base entities to mentions in text.

To overcome this problem we investigate the use of first-order logic background knowledge (e.g. implications) to aid relation extraction. One option is to rely on a fully symbolic approach that exclusively uses first-order logic (Bos and Markert, 2005; Baader et al., 2007; Bos, 2008). In this case incorporating additional background knowledge is trivial. However, it is difficult to generalize and deal with noise and uncertainty in language when relying only on manual rules. In contrast, matrix factorization methods can overcome these shortcomings, but it is not clear how they can be combined with logic formulae.

In this section, we propose to inject formulae into the embeddings of relations and entity-pairs, i.e., estimate the embeddings such that predictions based on them conform to given logic formulae (see Figure 1 for an overview). We refer to such embeddings as low-rank logic embeddings. Akin to matrix factorization, inference of a fact at test time still amounts to an efficient dot product of the corresponding relation and entity-pair embeddings, and logical inference is not needed. We present two techniques for injecting logical background knowledge, pre-factorization inference (§3.1) and joint optimization (§3.2), and demonstrate in subsequent sections that they generalize better than direct logical inference, even if such inference is performed on the predictions of the matrix factorization model.

3.1 Pre-Factorization Inference

Background knowledge in form of first-order formulae can be seen as hints that can be used to generate additional training data (Abu-Mostafa, 1990). For pre-factorization inference we first perform logical inference on the training data and add inferred facts as additional training data. For example, for a formula $F = \forall x,y : r_s(x, y) \Rightarrow r_t(x, y)$, we add an additional observed cell $r_t(x, y)$ for any $(x, y)$ for which $r_s(x, y)$ is observed in the distant supervision training data. This is repeated until no further facts can be inferred. Subsequently, we run matrix factorization on the extended set of observed cells.

The intuition is that the additional training data generated by the formulae provide evidence of the logical dependencies between relations to the matrix factorization model, while at the same time allowing the factorization to generalize to unobserved facts and to deal with ambiguity and noise in the data. No further logical inference is performed during or after training of the factorization model as we expect that the learned embeddings encode the given formulae.

3.2 Joint Optimization

One drawback of pre-factorization inference is that the formulae are enforced only on observed atoms,
i.e., first-order dependencies on predicted facts are ignored. Instead we would like to include a loss term for the logical formulae directly in the matrix factorization objective, thus jointly optimizing embeddings to reconstruct factual training data as well as obeying to first-order logical background knowledge.

3.2.1 Training Objective

Here we first present a learning objective that unifies ground atoms (facts) and logical background knowledge by treating both as logic formulae (atomic or complex), and define a loss function over this general representation. We then define the loss function for ground atoms and simple implications, along with a brief sketch of how the loss can be defined for arbitrarily complex logic formulae.

As introduced in §2.1, let \( \mathcal{R} \) be the set of all relations/predicates and \( \mathcal{P} \) be the set of all entity-pairs/constants. Furthermore, let \( \mathfrak{F} \) be a training set of logic formulae \( \mathcal{F} \), and \( L \) a loss function. The training objective (omitting \( \ell_2 \) regularization on \( \mathbf{v}(i) \) for simplicity) is

\[
\min_{\mathbf{v}} \sum_{\mathcal{F} \in \mathfrak{F}} L([\mathcal{F}])
\]

(1)

where \( \mathbf{v} \) is the set of all relation and entity-pair embeddings, and \( [\mathcal{F}] \) is the marginal probability \( p(w|\mathbf{v}) \) that the formula \( \mathcal{F} \) is true under the model. In this paper we use the logistic loss: \( L([\mathcal{F}]) := -\log([\mathcal{F}]) \). The objective thus prefers embeddings that assign formulae a high marginal probability.

To optimize this function we need the marginal probabilities \( [\mathcal{F}] \), and the gradients of the losses \( L([\mathcal{F}]) \) for every \( \mathcal{F} \in \mathfrak{F} \) with respect to entity-pair and relation embeddings, i.e., \( \partial L([\mathcal{F}]) / \partial \mathbf{v}_{r_m} \) and \( \partial L([\mathcal{F}]) / \partial \mathbf{v}_{(e_i,e_j)} \). Below we discuss how these quantities can be computed or approximated for arbitrary first-order logic formulae, with details provided for ground atoms and implications.

Ground Atoms  Due to the conditional independence of ground atoms in the distribution \( p(w|\mathbf{v}) \), the marginal probability of a ground atom \( \mathcal{F} = r_m(e_i,e_j) \) is \( \mathcal{F} = \pi_{m,e_i,e_j} = \sigma(\mathbf{v}_{r_m} \cdot \mathbf{v}_{e_i,e_j}) \). Hence when only ground atoms (or literals) are used, objective (1) reduces to the standard log-likelihood loss. The gradients of the loss for the entity-pair embedding \( \mathbf{v}_{(e_i,e_j)} \) and relation embedding \( \mathbf{v}_{r_m} \) are

\[
\partial L([\mathcal{F}]) / \partial \mathbf{v}_{(e_i,e_j)} = [\mathcal{F}](1 - [\mathcal{F}]) \mathbf{v}_{r_m}
\]

(2)

\[
\partial L([\mathcal{F}]) / \partial \mathbf{v}_{r_m} = [\mathcal{F}](1 - [\mathcal{F}]) \mathbf{v}_{(e_i,e_j)}
\]

(3)

\[
\partial [\mathcal{F}] / \partial \mathbf{v}_{(e_i,e_j)} = [\mathcal{F}]^{-1} \partial [\mathcal{F}] / \partial \mathbf{v}_{(e_i,e_j)}
\]

(4)

\[
\partial [\mathcal{F}] / \partial \mathbf{v}_{r_m} = [\mathcal{F}]^{-1} \partial [\mathcal{F}] / \partial \mathbf{v}_{r_m}
\]

(5)

First-order Logic  Crucially, and in contrast to the log-likelihood loss for matrix factorization, we can inject more expressive logic formulae than just ground atoms. We briefly outline how to recursively compute the probability of the formula \( \mathcal{F} \) and the gradients of the loss \( L([\mathcal{F}]) \) for any first-order formula \( \mathcal{F} \). Again, note that the probabilities of ground atoms in our model are independent conditioned on embeddings. This means that for any two formulae \( \mathcal{A} \) and \( \mathcal{B} \), the marginal probability of \( \mathcal{A} \land \mathcal{B} \) can be computed as \( \mathcal{A} \land \mathcal{B} \) (known as product t-norm), provided both formula concern non-overlapping sets of ground atoms. In combination with \( [\neg \mathcal{A}] := 1 - [\mathcal{A}] \) and the \( [\ ] \) operator as defined for ground atoms earlier, we can compute the probability of any propositional formula recursively, e.g.,

\[
[\mathcal{A} \lor \mathcal{B}] = [\mathcal{A}] + [\mathcal{B}] - [\mathcal{A}][\mathcal{B}]
\]

\[
[\mathcal{A} \implies \mathcal{B}] = [\mathcal{A}]/([\mathcal{B}] - 1) + 1
\]

\[
[\mathcal{A} \land \neg \mathcal{B} \implies \mathcal{C}] = ([\mathcal{A}]/([\mathcal{B}] - 1))/([\mathcal{C}] - 1) + 1.
\]

Note that for statements \( [\mathcal{F}] \in \{0,1\} \), we directly recover logical semantics. First-order formulae in finite domains can be embedded through explicit grounding. For universal quantification we can get

\[
[\forall x,y : \mathcal{F}(x,y)] = \bigwedge_{x,y} [\mathcal{F}(x,y)].
\]

If we again assume non-overlapping ground atoms in each of the arguments of the conjunction, we can simplify this to

\[
[\forall x,y : \mathcal{F}(x,y)].
\]

When arguments do overlap we can think of this simplification as an approximation.

Since \( [\mathcal{F}(x,y)] \) is defined recursively, we can back-propagate the training signal through the structure of \( \mathcal{F} \) to compute \( \partial [\mathcal{F}(x,y)] / \partial \mathbf{v}_{r_m} \) and \( \partial [\mathcal{F}(x,y)] / \partial \mathbf{v}_{e_i,e_j} \) for any nested formula.

Implications  A particularly useful family of formulae for relation extraction are universally quantified first-order formula over a knowledge base such as \( \mathcal{F} = \forall x,y : r_s(x,y) \implies r_t(x,y) \). Assuming a finite domain, such a formula can be unrolled into a conjunction of propositional statements of the form \( \mathcal{F}_{ij} = r_s(e_i,e_j) \implies r_t(e_i,e_j) \), one for
each entity-pair \((e_i, e_j)\) in the domain. Specifically, 
\[
[F] = \prod_{(e_i, e_j) \in \mathcal{P}} [F_{ij}],
\]
and therefore 
\[
\mathcal{L}([F]) = \sum_{(e_i, e_j) \in \mathcal{P}} \mathcal{L}([F_{ij}]).
\]
The gradients are derived as:

\[
\frac{\partial \mathcal{L}([F_{ij}])}{\partial v_{r_{i}}} = -[F_{ij}]^{-1} (r_{t}(e_i, e_j) - 1) \frac{\partial [r_{s}(e_i, e_j)]}{\partial v_{r_{i}}}
\]
\[
\frac{\partial \mathcal{L}([F_{ij}])}{\partial v_{r_{i}}} = -[F_{ij}]^{-1} [r_{s}(e_i, e_j)] \frac{\partial [r_{s}(e_i, e_j)]}{\partial v_{r_{i}}}
\]
\[
\frac{\partial \mathcal{L}([F_{ij}])}{\partial v_{e_i, e_j}} = -[F_{ij}]^{-1} [r_{s}(e_i, e_j)] \frac{\partial [r_{s}(e_i, e_j)]}{\partial v_{e_i, e_j}}.
\]

Following such a derivation, one can obtain gradients for other first-order logic formulae as well.

### 3.2.2 Learning

We learn the embeddings by minimizing Eq. 1 with \(\ell_2\)-regularization using AdaGrad (Duchi et al., 2011). Since we have no negative training facts, we follow Riedel et al. (2013) by sampling unobserved facts that we assume to be false. Specifically, in every epoch and for every true training fact \(r_{m}(e_i, e_j)\) we sample an \((e_p, e_q)\) such that \(r_{m}(e_p, e_q)\) is unobserved. Subsequently, we perform two kinds of updates: \(F = r_{m}(e_i, e_j)\) and \(F = -r_{m}(e_p, e_q)\). For every non-atomic first-order formula in \(\mathcal{F}\) we iterate over all entity-pairs for which at least one atom in the formula is observed (in addition to as many sampled entity-pairs for which \(none\) of the atoms have been observed) and add corresponding grounded propositional formulae to the training objective. At test time, predicting a score for any unobserved statement \(r_{m}(e_i, e_j)\) is done efficiently by calculating \([r_{m}(e_i, e_j)]\). Note that this does not involve any explicit logical inference, instead we expect that the predictions from the learned embeddings already respect the provided formulae.

### 4 Experimental Setup

There are two orthogonal question when evaluating the effectiveness of low-rank logic embeddings: a) does injection of logic formulae into the embeddings of entity-pairs and relations provide any benefits, and b) where do the background formulae come from? The latter is a well-studied problem (Hipp et al., 2000; Schoenmackers et al., 2010; Völker and Niepert, 2011). In this paper we focus the evaluation on the ability of various approaches to benefit from formulae that we directly extract from the training data using a simple method.

#### Distant Supervision Evaluation

We follow the procedure as used in Riedel et al. (2013) for evaluating knowledge base completion of Freebase (Bollacker et al., 2008) with textual data from the NYT-Matrix factorization model. We first run matrix factorization over the complete training data to learn accurate relation and entity-pair embeddings. After training, we iterate over all pairs of relations \((r_s, r_t)\) where \(r_t\) is a Freebase relation. For every relation-pair we iterate over all training atoms \(r_s(e_i, e_j)\), evaluate the score \([r_s(e_i, e_j) \Rightarrow r_t(e_i, e_j)]\) as described in §3.2.1, and calculate the average to arrive at a score for the formula. Finally, we rank all formulae by their score and manually filter the top 100 formulae, which resulted in 36 annotated high-quality formulae (see Table 1 for examples). Note that our formula extraction approach does not observe the relations for test entity-pairs. All models used in our experiments have access to these formulae, except for the matrix factorization baseline.

#### Methods

Our proposed methods for injecting logic into relation embeddings are **pre-factorization inference** (Pre; §3.1) which performs regular matrix factorization after propagating the logic formulae in a deterministic manner, and **joint optimization** (Joint; §3.2) which maximizes an objective that combines terms from factual and first-order logic knowledge. Additionally, we use the following three baselines. The **matrix factorization** (MF; §2.2) model uses only ground atoms to learn relation and entity-pair embed-
Table 1: **Sample Extracted Formulae**: Top implications of textual patterns to five different Freebase relations. These implications were extracted from the matrix factorization model and manually annotated. The premises of these implications are dependency paths, but we present a simplified version to make them more readable.

<table>
<thead>
<tr>
<th>Formula</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall x,y : #2$-unit-of-$#1(x,y) \Rightarrow org/parent/child(x,y)$</td>
<td>0.97</td>
</tr>
<tr>
<td>$\forall x,y : #2$-city-of-$#1(x,y) \Rightarrow location/containedby(y,x)$</td>
<td>0.97</td>
</tr>
<tr>
<td>$\forall x,y : #2$-minister-$#1(x,y) \Rightarrow person/nationality(y,x)$</td>
<td>0.97</td>
</tr>
<tr>
<td>$\forall x,y : #2$-co-founder-of-$#1(x,y) \Rightarrow company/founders(y,x)$</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Table 2: **Zero-shot Relation Learning**: Average and (weighted) mean average precisions with relations that do not appear in any of the annotated formulae omitted from the evaluation. The difference between “Pre” and “Joint” is significant according to the sign-test ($p < 0.05$).

<table>
<thead>
<tr>
<th>Relation</th>
<th>PF</th>
<th>Inf</th>
<th>Post</th>
<th>Pre</th>
<th>Joint</th>
</tr>
</thead>
<tbody>
<tr>
<td>person/company</td>
<td>0.07</td>
<td>0.03</td>
<td>0.15</td>
<td>0.31</td>
<td><strong>0.35</strong></td>
</tr>
<tr>
<td>location/containedby</td>
<td>0.03</td>
<td>0.06</td>
<td>0.14</td>
<td>0.22</td>
<td><strong>0.31</strong></td>
</tr>
<tr>
<td>author/work/author</td>
<td>0.02</td>
<td>0.05</td>
<td>0.18</td>
<td><strong>0.31</strong></td>
<td>0.27</td>
</tr>
<tr>
<td>person/nationality</td>
<td>0.01</td>
<td>0.19</td>
<td>0.09</td>
<td>0.15</td>
<td>0.19</td>
</tr>
<tr>
<td>parent/child</td>
<td>0.01</td>
<td>0.01</td>
<td>0.48</td>
<td>0.66</td>
<td><strong>0.75</strong></td>
</tr>
<tr>
<td>person/place/of_birth</td>
<td>0.01</td>
<td>0.43</td>
<td>0.40</td>
<td>0.56</td>
<td><strong>0.59</strong></td>
</tr>
<tr>
<td>person/place/of_death</td>
<td>0.01</td>
<td>0.24</td>
<td>0.23</td>
<td><strong>0.27</strong></td>
<td>0.23</td>
</tr>
<tr>
<td>neighborhood/neighborhood_of</td>
<td>0.00</td>
<td>0.00</td>
<td>0.60</td>
<td>0.63</td>
<td><strong>0.65</strong></td>
</tr>
<tr>
<td>person/parents</td>
<td>0.00</td>
<td>0.17</td>
<td>0.19</td>
<td>0.37</td>
<td>0.65</td>
</tr>
<tr>
<td>company/founders</td>
<td>0.00</td>
<td>0.25</td>
<td>0.13</td>
<td>0.37</td>
<td><strong>0.77</strong></td>
</tr>
<tr>
<td>film/directed_by</td>
<td>0.00</td>
<td>0.50</td>
<td>0.50</td>
<td>0.36</td>
<td><strong>0.51</strong></td>
</tr>
<tr>
<td>film/produced_by</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

MAP: 0.01, 0.23, 0.34, 0.43, 0.52
Weighted MAP: 0.03, 0.10, 0.21, 0.33, 0.38

5.1 Zero-shot Relation Learning

We start with the scenario of learning extractors for relations that do not appear in the knowledge base schema, *i.e.*, those that do not have any textual alignments. Such a scenario occurs in practice when a new relation needs to be added to a knowledge base for which there are no facts that connect the new relation to existing relations or surface patterns. For accurate extractions of such relations, the model needs to rely primarily on background domain knowledge to identify relevant textual alignments, but at the same time it also needs to utilize correlations between textual patterns for generalization. To simulate this setup, we remove all alignments between all entity-pairs and Freebase relations from the distant supervision data, use the extracted logic formulae (§4) as background knowledge, and evaluate on the ability of the different methods to recover the lost alignments.

Table 2 provides detailed results. Unsurprisingly,
matrix factorization (MF) performs poorly since embeddings cannot be learned for the Freebase relations without any observed cells. Scores higher than zero for matrix factorization are caused by random predictions. Logical inference (Inf) is limited by the number of known facts that appear as premise in one of the implications. Although post-factorization inference (Post) is able to achieve a large improvement over logical inference, explicitly injecting logic formulae into the embeddings (i.e., learning low-rank logic embeddings) using pre-factorization inference (Pre) or joint optimization (Joint) gives superior results. Last, we observe that joint optimization is able to best combine logic formulae and textual patterns for accurate, zero-shot learning of relation extractors.

5.2 Relations with Few Distant Labels

In this section we study the scenario of learning relations that have only a few distant supervision alignments, in particular, we observe the behavior of the various methods as the amount of distant supervision is varied. We run all methods on training data that contains different fractions of Freebase training facts (and therefore different degrees of relation/text pattern alignment), but keep all textual patterns in addition to the set of extracted formulae.

Figure 2 summarizes the results. The performance of pure logical inference does not depend on the amount of distant supervision data, since it does not take advantage of the correlations in the data. Matrix factorization ignores logic formulae, and thus is the baseline performance when only using distant supervision. For the factorization based methods, only a small fraction (15%) of the training data is needed to achieve around 0.50 wMAP performance, thus demonstrating that they are efficiently exploiting correlations and generalizing to unobserved facts.

Pre-factorization inference, however, does not outperform post-factorization inference, and is on par with matrix factorization for most of the curve. This suggests that it is not an effective way of injecting logic into embeddings when ground facts are also available. In contrast, joint optimization leads to low-rank logic embeddings that outperform all other methods in the 0 to 30% Freebase training data interval. Beyond 30% there seem to be sufficient Freebase facts for matrix factorization to encode these formulae, thus yielding diminishing returns.

5.3 Comparison on Complete Data

Although the focus of this paper is injection of logical knowledge for relations without sufficient alignments to the knowledge base, we also present an evaluation on the complete distant supervision data as used by Riedel et al. (2013). Compared to the Riedel et al.’s “F” model, our matrix factorization implementation (“MF”) achieves a lower wMAP (64% vs 68%) and a higher MAP (66% vs 64%). We attribute this difference to the different loss function (logistic loss in our case vs. ranking loss). We show the PR curve
in Figure 3, demonstrating that joint optimization provides benefits over the existing factorization and distant supervision techniques even on the complete dataset, and obtains 66% wMAP and 69% MAP. This improvement over the matrix factorization model can be explained by reinforcement of high-quality annotated formulae via the joint model.

5.4 Analysis of Asymmetry in the Predictions
Since the injected formulae are of the form $\forall x, y : r_s(x, y) \implies r_t(x, y)$, it is worthwhile to study the extent to which these rules are captured, and which approaches are in fact capturing the asymmetric nature of the implication. To this end, we compute the probabilities that the formulae and their inverse hold, averaged over all annotated formulae and cells. The degree to which $r_s \implies r_t$ is captured is quite high for all models (0.94, 0.96, and 0.97 for matrix factorization, pre-factorization inference, and joint optimization respectively). On the other hand, the probability of $r_t \implies r_s$ is also relatively high for matrix factorization and pre-factorization inference (0.81 and 0.83 respectively), suggesting that these methods are primarily capturing symmetric similarity between relations. Joint optimization, however, produces much more asymmetric predictions (probability of $r_t \implies r_s$ is 0.49), demonstrating that it is appropriate for encoding logic in the embeddings.

6 Related Work

Embeddings for Knowledge Base Completion
Embedding predicates and constants (or pairs of constants) based on factual knowledge for knowledge base completion has for instance been investigated by Bordes et al. (2011), Nickel et al. (2012), Socher et al. (2013), Riedel et al. (2013) and Fan et al. (2014). Our work goes further in that we learn embeddings that follow not only factual but also first-order logic knowledge, and the ideas presented in this paper can be incorporated with any embedding-based method that uses a per-atom loss.

Logical Inference A common alternative that directly incorporates first-order logic knowledge is to perform logical inference (Bos and Markert, 2005; Baader et al., 2007; Bos, 2008), however such purely symbolic approaches cannot deal with the uncertainty inherent to natural language, and generalize poorly.

Probabilistic Inference To ameliorate some of the drawbacks of symbolic logical inference, probabilistic logic based approaches have been proposed (Schoenmackers et al., 2008; Garrette et al., 2011; Beltagy et al., 2013; Beltagy et al., 2014). Since logical connections between relations are modeled explicitly, such approaches are generally hard to scale. Specifically, approaches based on Markov Logic Networks (MLNs) (Richardson and Domingos, 2006) encode logical knowledge in dense, loopy graphical models, making structure learning, parameter estimation, and inference hard for the scale of our data. In contrast, in our model the logical knowledge is captured directly in the embeddings, leading to efficient inference. Furthermore, as our model is based on matrix factorization, we have a natural way to deal with linguistic ambiguities and label errors.

Weakly Supervised Learning Our work is also inspired by weakly supervised approaches (Ganchev et al., 2010) that use structural constraints as a source of indirect supervision, and have been used for several NLP tasks (Chang et al., 2007; Mann and McCallum, 2008; Druck et al., 2009; Singh et al., 2010). Carlson et al. (2010) in particular is similar since they use common sense constraints to jointly train multiple information extractors. In this work, however, we are training a matrix factorization model, and allowing for arbitrarily complex logic formulae.

Combining Symbolic and Distributed Representations There have been a number of recent approaches that combine distributed representations with symbolic knowledge. Grefenstette (2013) describes an isomorphism between first-order logic and tensor calculus, using full-rank matrices to exactly memorize facts. Based on this isomorphism, Rocktäschel et al. (2014) combine logic with matrix factorization for learning low-dimensional embeddings that approximately satisfy given formulae and generalize to unobserved facts on toy data. Our work extends this workshop paper by proposing a simpler formalism without tensor-based logical connectives, presenting results on a real-world task, and demonstrating the utility of this approach for learning relations with few textual alignments.

Chang et al. (2014) use Freebase entity types as hard constraints in a tensor factorization objective for universal schema relation extraction. In contrast, our
approach is imposing soft constraints that are formulated as universally quantified first-order formula.

de Lacalle and Lapata (2013) combine first-order logic knowledge with a topic model to improve surface pattern clustering for relation extraction. Since these formulae only specify which relations can be clustered and which not, they do not capture the variety of dependencies embeddings can model, such as asymmetry. Lewis and Steedman (2013) use distributed representations to cluster predicates before logical inference. Again, this approach is not as powerful as factorizing the relations, as it makes symmetry assumptions for the predicates.

Several studies have investigated the use of symbolic representations (such as dependency trees) to guide the composition of distributed representations (Clark and Pulman, 2007; Mitchell and Lapata, 2008; Coecke et al., 2010; Hermann and Blunsom, 2013). Instead we are using symbolic representations (first-order logic) as prior domain knowledge to directly learn better embeddings.

Combining symbolic information with neural networks has also been an active area of research. Towell and Shavlik (1994) introduce Knowledge-Based Artificial Neural Networks whose topology is isomorphic to a knowledge base of facts and inference formulae. There, facts are input units, intermediate conclusions hidden units, and final conclusions (inferred facts) output units. Unlike our work, there is no latent representation of predicates and constants. Hölldobler et al. (1999) and Hitzler et al. (2004) prove that for every logic program theoretically there exists a recurrent neural network that approximates the semantics of that program. Finally, Bowman (2014) recently demonstrated that a neural tensor network can accurately learn natural logic reasoning.

7 Conclusions

Inspired by the benefits of logical background knowledge that can lead to precise extractors, and of distant supervision based matrix factorization that can utilize dependencies between textual patterns to generalize, in this paper we introduced a novel training paradigm for learning embeddings that combine matrix factorization with logic formulae. Along with a deterministic approach to enforce the formulae a priori, we propose a joint objective that rewards predictions that satisfy given logical knowledge, thus learning embeddings that do not require logical inference at test time. Experiments show that the proposed approaches are able to learn extractors for relations with little to no observed textual alignments, while at the same time benefiting more common relations. The source code of the methods presented in this paper and the annotated formulae used for evaluation are available at github.com/uclmr/low-rank-logic.

This research has thrown up many questions in need of further investigation. As opposed to our approach that modifies both relation and entity-pair embeddings, further work needs to explore training methods that only modify relation embeddings in order to encode logical dependencies explicitly, and thus avoid memorization. Although we obtain significant gains by using implications, our approach facilitates the use of arbitrary formulae; it would be worthwhile to pursue this direction by following the steps outlined in §3.2.1. Furthermore, we are interested in combining relation extraction with models that learn entity type representations (e.g., tensor factorization or neural models) to allow for expressive logical statements such as $\forall x, y: \text{nationality}(x, y) \Rightarrow \text{country}(y)$. Since such common sense formulae are often not directly observed in distant supervision, they can go a long way in fixing common extraction errors. Finally, we will investigate methods to automatically mine common-sense knowledge for injection into embeddings from additional resources like Probate (Wu et al., 2012) or directly from text using a semantic parser (Zettlemoyer and Collins, 2005).

Acknowledgments

The authors want to thank Mathias Niepert for proposing pre-factorization inference as alternative to joint optimization. We thank Edward Grefenstette, Luke Zettlemoyer, and Guillaume Bouchard for comments on the manuscript, and the reviewers for very helpful feedback. This work was supported in part by Microsoft Research through its PhD Scholarship Programme, in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA, and in part by the Paul Allen Foundation through an Allen Distinguished Investigator grant.
References


Laura Chiticariu, Yunyao Li, and Frederick R. Reiss. 2013. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 827–832.


Dan Garrette, Katrin Erk, and Raymond Mooney. 2011. Integrating logical representations with probabilistic
information using markov logic. In Proc. of International Conference on Computational Semantics (IWCS), pages 105–114. ACL.


