Transfer Learning from Minimal Target Data by Mapping across Relational Domains

Lilyana Mihalkova and **Raymond J. Mooney**
Department of Computer Sciences
The University of Texas at Austin
1 University Station C0500
Austin, TX 78712-0233, USA
{lilyanam,mooney}@cs.utexas.edu

Abstract

A central goal of transfer learning is to enable learning when training data from the domain of interest is limited. Yet, work on transfer across relational domains has so far focused on the case where there is a significant amount of target data. This paper bridges this gap by studying transfer when the amount of target data is minimal and consists of information about just a handful of entities. In the extreme case, only a single entity is known. We present the SR2LR algorithm that finds an effective mapping of predicates from a source model to the target domain in this setting and thus renders preexisting knowledge useful to the target task. We demonstrate SR2LR's effectiveness in three benchmark relational domains on social interactions and study its behavior as information about an increasing number of entities becomes available.

1 Introduction

Machine learning algorithms have traditionally been designed assuming that an adequate amount of training data for the task of interest is available. Although numerous successful approaches for this case have been developed, their accuracy will suffer when training data is very limited. One of the most effective techniques for enabling learning in such situations is *transfer learning*, i.e. transferring a *source* model learned in a domain that is related to the *target* domain at hand [Silver *et al.*, 2005; Banerjee *et al.*, 2006; Taylor *et al.*, 2008]. Transfer learning has been successful in a variety of learning problems, e.g., [Raina *et al.*, 2006; Niculescu-Mizil and Caruana, 2007; Torrey *et al.*, 2007].

One area in which transfer learning has proven particularly effective is statistical relational learning (SRL). In SRL, general probabilistic models are learned from multi-relational data, in which a set of entities are engaged in a variety of complex relations [Getoor and Taskar, 2007]. For example, in a domain describing an academic institution, e.g., [Richardson and Domingos, 2006], the entities are people, publications, and courses, whereas the relations are `advised-by`, `taught-by`, and `written-by`. In addition, each entity has attributes (i.e, relations of arity 1), such as `is-student` and `is-professor`. As a result of the rich connections among the entities, individual training examples are typically very large, containing hundreds of entities, have varying lengths, and cannot be broken down into smaller disconnected components. To emphasize this, we call relational training instances *mega-examples*. In an academic domain, a mega-example may describe an entire area of study, such as AI. Because of these characteristics of multi-relational data, SRL algorithms have long training times and are often susceptible to local maxima and plateaus. Effective transfer learning approaches have been developed to combat these problems, leading to improvements both in the speed and the accuracy of learning [Mihalkova *et al.*, 2007; Davis and Domingos, 2008].

However, to the best of our knowledge, all existing algorithms for transfer in SRL assume that an adequate amount of target domain data, i.e., at least one full mega-example, is available. There currently are no techniques for the case of limited target data, in which transfer learning could have the greatest impact. This paper bridges this gap by addressing the setting of minimal target domain data that consists of just a handful of entities. In the extreme case, a single entity is known. Fig. 1 contrasts the amount of data assumed by previous work to that assumed here.

This setting may arise in a variety of situations. For instance, when a new social networking site is launched, data is available on only a few initial registrants. The popularity of the site depends on its ability to make meaningful predictions that would allow it to suggest promising friendships to users. However, the sparcity of available data and the fact that data from other social networking sites is usually proprietary make learning of an effective model from scratch infeasible.

Frequently, two domains differ in their representations, but the underlying regularities that govern the dynamics in each domain are similar. So, when transferring a model learned from an academic data set to a movie business domain, one may discover that students and professors are similar to actors and directors respectively, which makes writing an academic paper analogous to directing or participating in a movie. Likewise, because human interactions bear a certain degree of similarity across settings, the social networking site can learn strong models from data on the professional relations among its employees and map them for the task of interest based on its very limited supply of data from the new site.

When target data is so limited, effective transfer depends

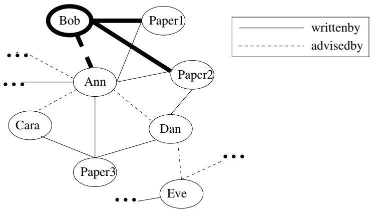

Figure 1: Target data available in previous versus current work. The nodes in this graph represent the entities in the domain and the edges represent the relations in which these entities participate. Previous work assumes that the information from the entire graph is provided. The present paper assumes that just the bold relations are known.

on the ability to map the representation of a source model learned in a closely related domain to that of the target task. The main challenge addressed in this work is, therefore, to harness the small amount of data in the target domain in order to find useful mappings between the source and target representations. We present an efficient algorithm for this task, SR2LR (which stands for Short-Range To Long-Range), that is based on the observation that a good model for the source domain contains two types of clauses—short-range ones that concern the properties of a single entity and long-range ones that relate the properties of several entities. Because possible mappings of the short-range clauses to the target domain can be directly evaluated on the available target data, the key is to use the short-range clauses in order to find mappings between the relations in the two domains, which are then used to translate the long-range clauses.

As in previous work on transfer in SRL [Mihalkova *et al.*, 2007; Davis and Domingos, 2008], we transfer a Markov logic network (MLN) [Richardson and Domingos, 2006]. We provide a detailed description of MLNs in Section 2. SR2LR is not limited to MLNs, and after describing the algorithm in Section 3, we discuss what other representations can be used. Then, in Section 4, we demonstrate the effectiveness of SR2LR in three benchmark relational domains.

2 Background

In first-order logic, a *predicate* represents a relation in the domain, such as `advised-by`. Predicates are like functions that return true or false in which arguments have *type*s. For example `written-by` takes one argument of type `paper` and one of type `person`. An *atom* is a predicate applied to terms, where the terms can be variables or constants.[1] Constants represent the *entities*. A (negative/positive) *literal* is an atom that (is/is not) negated. A literal whose terms are constants is *ground*. A clause is ground if all of its literals are ground. The word *grounding* refers to a ground literal or clause.

An MLN consists of a set of weighted formulae and provides a way of softening first-order logic by making situations in which not all clauses are satisfied less likely but not impossible [Richardson and Domingos, 2006]. Let $\mathbf{X}$ be the set of all propositions describing a world (i.e., all ground literals in the domain), $\mathcal{F}$ be the set of all clauses in the MLN, w_i be the weight of clause f_i, $\mathcal{G}_{f_i}$ be the set of all possible groundings of clause f_i, and Z be the normalizing partition function. Then the probability of a particular truth assignment $\mathbf{x}$ to $\mathbf{X}$ is given by the formula $P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{f_i \in \mathcal{F}} w_i \sum_{g \in \mathcal{G}_{f_i}} g(\mathbf{x})\right)$. The value of $g(\mathbf{x})$ is 1 if ground clause g is satisfied and 0 otherwise.

The TAMAR algorithm [Mihalkova *et al.*, 2007], which transfers MLNs by mapping and revising them, is the most closely related to this work. We will review the mapping portion of TAMAR, which we call MTAMAR and to which we compare SR2LR. MTAMAR uses the concept of a *type-consistent* mapping. A mapping of a source clause to the target domain implies a correspondence from the source predicates in the clause to a subset of the target predicates. Such a correspondence between a source predicate and a target predicate implicitly defines a mapping between the types of the arguments of the two predicates. A mapping is *type-consistent* if, within a clause, a type in the source domain is mapped to at most one type in the target domain. MTAMAR maps each source clause independently of the others by evaluating all possible type-consistent mappings with the weighted pseudo log-likelihood score from [Kok and Domingos, 2005]. This measure assumes that at least one full target mega-example is provided and uses the closed-world assumption to conclude that ground facts not listed as true in the data are false. Revising an MLN given the limited target data assumed in our setting is infeasible. Thus, we will not use the revision portion of TAMAR, and SR2LR does not perform revision.

3 The SR2LR Algorithm

We first describe the algorithm for the extreme *single-entity-centered* setting, in which information about only one entity is available. Then we generalize to more than one entity. More precisely, for now we assume that the data lists all true atoms concerning a **central** entity E, and only those atoms. Atoms that involve E but are not listed are assumed to be false. Atoms that do not involve E have unknown values.

SR2LR starts by producing all type-consistent mappings (defined in Section 2) of the source clauses. The key idea of SR2LR is to find valid source-to-target predicate correspondences by directly evaluating only the mapped clauses whose performance can be measured on the available target data and then to use these correspondences to map clauses whose accuracy cannot be directly evaluated. Mapped clauses that can be directly evaluated are short-range; the rest are long-range.

Definition 1 *A clause C is* **short-range** *with respect to an entity of type t iff there exists a variable v that appears in every literal of C and v represents arguments of type t. A clause is* **long-range** *with respect to E iff it is not short-range.*

As an example, suppose we would like to transfer the MLN in Fig. 2 using the data in Fig. 3, i.e., transfer from a movie domain to an academic domain. Let us consider one possible type-consistent mapping of the first clause in Fig. 2, which is given in line 1.1 of Fig. 4. Note that variable a appears in both literals of this clause. Therefore, the clause is short-range. The truth value of any grounding such that $a = Bob$ can be directly evaluated from the data. However, if we use the substitution $a = Ann, b = Bob$, the resulting grounding cannot be evaluated because the truth-value of `is-professor`(Ann) is unknown. We say that the first

[1] We assume the domains contain no logical functions.

1	0.7 : worked-for(a, b) ⇒ ¬is-director(a)
2	0.8 : in-movie(m, a) ∧ in-movie(m, b) ∧is-director(b) ⇒ worked-for(a, b)

Figure 2: Source MLN

is-student(Bob), written-by(Paper1, Bob),
written-by(Paper2, Bob) , advised-by(Bob, Ann)

Figure 3: Target domain data centered around Bob. All listed atoms are true; atoms about Bob that are not listed are false; the remaining atoms have unknown values.

grounding is **verifiable,** whereas the second is not. Now consider one possible mapping of the second clause in Fig. 2, given in line 2.1 of Fig. 4. This clause concerns relations that go beyond just a single entity, e.g., about papers written by other people and is therefore long-range.

Algorithm 1 formally describes SR2LR. In line 1, the weight of a mapped clause is set to the weight of the source clause from which it was mapped. Because of limited target data, we do not attempt to re-learn weights or to revise the mapped clauses.[2] In line 3, the short-range mapped clauses are evaluated, as described in Algorithm 2, which checks whether the verifiable groundings of short-range clauses are satisfied in the target data. Clauses that are satisfied at least Θ proportion of the time are accepted; the rest are rejected. This procedure automatically rejects clauses that are not informative. A short-range clause is **informative** with respect to a single-entity-centered example if it has a verifiable grounding in which at least one ground literal is false. Intuitively, a clause is uninformative if, in every possible re-writing of the clause as an implication, the premises are never satisfied, and so the clause is always *trivially* true. For example, consider the clause `is-student`$(a) \vee \neg$`advised-by`(b, a), which has two verifiable groundings corresponding to the substitutions a=Bob, b=Ann, and a=Bob, b=Bob. It is not informative because all the literals in its verifiable groundings are true. To develop intuition for the significance of this, consider one of the groundings: `is-student`$(Bob) \vee \neg$`advised-by`(Ann, Bob). We can re-write it as $\neg$`is-student`$(Bob) \Rightarrow \neg$`advised-by`(Ann, Bob) or equivalently as `advised-by` $(Ann, Bob) \Rightarrow$ `is-student`(Bob). In both cases, the premises, or antecedents, of these clauses do not hold, and thus the clauses cannot be used to draw inferences that can be tested. So, judgements about mappings based on such clauses are likely to be misleading.

Once the short-range clauses are evaluated, in line 5 of Algorithm 1, SR2LR evaluates the long-range ones, based on the mappings found to be useful for short-range clauses. A long-range clause is accepted if all source-to-target predicate mappings implied by it either led to accepted short-range clauses (**support by evaluation**) or were never considered by Algorithm 2 (**support by exclusion**). More precisely, let C_S and C_L be short-range and long-range mapped clauses respectively. If the set of source-to-target predicate correspondences implied by C_S is a subset of those implied by C_L, we say that the literals of C_L that appear in C_S are **supported by evaluation**. A correspondence between source predicate P_S and target predicate P_T is **supported by exclusion** with respect to a set of mapped short-range clauses $\mathcal{S}$ if P_S and P_T do not appear in any of the source-to-target predicate correspondences implied by the clauses in $\mathcal{S}$. The goal of support by exclusion is to allow for predicates that do not appear in the short-range clauses to be mapped. Although support by exclusion may seem too risky, i.e., if a pair of completely unrelated source and target predicates are mapped to each other, in our experience the type consistency constraint and the requirement that neither of the predicates was mapped to any other predicate were strong enough to safeguard against this.

1.1	advised-by(a, b) ⇒ ¬is-professor(a)
worked-for → advised-by, is-director → is-professor	
1.2	advised-by(a, b) ⇒ ¬is-student(a)
worked-for → advised-by, is-director → is-student	
2.1	written-by(m, a) ∧ written-by(m, b)∧ is-professor(b) ⇒ advised-by(a, b)
worked-for → advised-by, is-director → is-professor, in-movie → written-by	
2.2	written-by(m, a) ∧ written-by(m, b)∧ is-student(b) ⇒ advised-by(a, b)
worked-for → advised-by, is-director → is-student, in-movie → written-by	

Figure 4: Example mapped clauses. The predicate correspondences used to map each clause are listed under it.

Algorithm 1 SR2LR algorithm

```
Input: SrcMLN: Source Markov logic network
    T_E: Target data centered on the entity $E$
    $\mathcal{P}$: Set of predicates in the target domain
    $\Theta$: Truth threshold for accepting a short-range clause
Procedure:
1: Generate TarMap, the set of all possible type-consistent mappings of the clauses in
   SrcMLN. Each mapped clause gets the weight of its corresponding source clause.
2: Split the clauses in TarMap into sets of short-range clauses, $\mathcal{S}$, and long-range
   clauses, $\mathcal{L}$.
3: $\mathcal{S}'$ = filter-short-range($\mathcal{S}$, $\Theta$) (Algorithm 2)
4: Add all clauses from $\mathcal{S}'$ to Result
5: $\mathcal{L}'$ = filter-long-range($\mathcal{L}$, $\mathcal{S}'$) (Algorithm 3)
6: Add all clauses from $\mathcal{L}'$ to Result
7: Let $\mathcal{A}_C$ be the set of all clauses in Result mapped from source clause $C$ with
   weight $w_C$.
8: Set the weight of each $a \in \mathcal{A}_C$ to $w_C/|\mathcal{A}_C|$.
```

We now illustrate Algorithm 1 up to line 7. Fig. 4 lists some mappings of the clauses in Fig. 2, along with the source-to-target predicate correspondences implied by them. Clauses 1.1 and 1.2 are (informative) short-range, and 2.1 and 2.2 are long-range. Let $\Theta = 1$. All verifiable groundings of clause 1.1 are satisfied by the target data (given in Fig. 3). Thus, this clause is accepted and the predicate correspondences found by it are useful. Clause 1.2 is rejected because not all of its verifiable groundings are satisfied by the target data. Thus $\mathcal{S}'$ contains only clause 1.1. Moving on to the long-range clauses, we see that predicates `advised-by` and `is-professor` in clause 2.1 are supported by clause 1.1; `written-by` is supported by exclusion, so clause 2.1 is accepted. Clause 2.2 is not accepted because there is no support for `is-student(b)`.

Finally, in lines 7-8 of Algorithm 1 the weight of each mapped clause M_C is divided by the number of mapped clauses that originated from the same source clause as M_C in order to ensure that none of the source clauses dominates the resulting model. In preliminary experiments this led to slightly better performance. The experiments supporting this conclusion are omitted because of space considerations.

The generalization to more than one entity is easy. The only difference is that now we have *a set* of single-entity-

[2] MTAMAR also directly copies the weights.

Algorithm 2 filter-short-range($\mathcal{S}, \Theta$)

```
$\mathcal{S}' = \emptyset$
for each $C \in \mathcal{S}$ do
    if $C$ is informative and the proportion of verifiable groundings of $C$ that are true is $\geq \Theta$ then
        Add $C$ to $\mathcal{S}'$
Return $\mathcal{S}'$
```

Algorithm 3 filter-long-range($\mathcal{L}, \mathcal{S}'$)

```
$\mathcal{L}' = \emptyset$
for each $LR \in \mathcal{L}$ do
    if All literals in $LR$ are supported either by evaluation based on the clauses in $\mathcal{S}'$ or by exclusion then
        Add $LR$ to $\mathcal{L}'$
Return $\mathcal{L}'$
```

centered training examples, and Algorithm 2 checks the validity of each short-range clause on each of the examples, accepting a clause if it holds more than Θ proportion of the time over all examples. As more entities become known, some of the long-range clauses become directly verifiable. However, in preliminary experiments (not presented because of space), we found that directly evaluating long-range clauses in this way does not significantly help performance, i.e., additional entities lead to improved accuracy mostly because they allow for more reliable evaluation of the short-range clauses.

Choice of Representation The only characteristic of MLNs crucial to SR2LR is that MLNs use first-order clauses that are interpreted in the standard way for first-order logic, i.e. by evaluating their truth values. SR2LR would therefore be applicable to any relational model based on a traditional interpretation of first-order logic, such as purely logical representations, stochastic logic programs [Muggleton, 1996], and MACCENT [Dehaspe, 1997]. MLNs have properties which, while not crucial to SR2LR, contribute to its effectiveness. In particular, the ability of MLNs to handle uncertainty allows SR2LR to recover gracefully from an occasional incorrect predicate mapping.

4 Experiments

We first describe methodology common to all experiments and then discuss the empirical questions we asked.

4.1 Methodology

We compared SR2LR to MTAMAR and other baselines in three benchmark relational domains on social interactions: IMDB, UW-CSE, and WebKb.[3] IMDB is about relations in the movie business and contains predicates such as `director, actor, movie, workedUnder`. The goal is to predict the `workedUnder` relation, which takes two arguments of type `person` and indicates that the first one acted in a movie directed by the second. UW-CSE is about interactions in an academic environment and contains predicates such as `student, professor, advisedBy, publication`. The goal is to predict the `advisedBy` relation, which takes two arguments of type `person` and indicates that the second one is the research advisor of the first.

The IMDB and UW-CSE domains have closely related dynamics, which, however, are expressed in differing representations. For example, in IMDB an actor and a director are usually in a `workedUnder` relationship if they appear in the credits of the same movie. Analogously, in UW-CSE a student and a professor are typically in an `advisedBy` relationship if they appear in the author list of the same publication. Thus, an algorithm capable of discovering effective mappings from the predicates of one domain to those of the other, would be able to achieve good accuracy via transfer. This example also demonstrates why data centered around a single entity, or a handful of isolated entities, cannot support effective learning from scratch: one of the most useful clauses for predicting `advisedBy` involves knowledge about the publications of two *connected* entities, i.e., the advisor and the advisee.

We also used the WebKb domain, which contains predicates such as `student, faculty, project`. Although UW-CSE may seem more closely related to this domain than to IMDB, in fact, WebKb does not have a predicate analogous to `advisedBy`, which renders it much less useful for transfer. We note that although some of the predicates occur in more than one domain under the same name, the systems do not use the actual predicate names. As sources, we used MLNs learned with the BUSL algorithm, demonstrated to give good performance in the domains we consider [Mihalkova and Mooney, 2007]. We slightly modified BUSL to encourage it to learn larger models by removing the `minWeight` threshold and by treating the clauses learned for each predicate separately. We call these models **learned.** For transfer from UW-CSE, we also used the manually coded knowledge base provided with that data set. We call it **manual**[4].

The results are reported in terms of two metrics: AUC-PR and CLL, commonly used for evaluation of MLNs and in SRL, e.g., [Kok and Domingos, 2005]. AUC-PR is the area under the precision-recall curve. A high AUC-PR score signifies that the algorithm correctly assigns a higher probability to the true positives than to the true negatives. AUC-PR is particularly appropriate for relational domains because it focuses on how well the algorithm predicts the few true positives and is not misled by the large number of true negatives. CLL is the conditional log-likelihood. We report CLL for completeness; however, because we are unable to tune the weights of the MLN on the limited target data, the CLL may be misleading. This can happen when the predicted probabilities are correctly ordered, i.e., true ground atoms have higher probability than false ones (thus giving a high AUC-PR), but are not close to 0 or 1 (thus giving a low CLL). At the same time, because of the large number of true negatives, the CLL can be boosted by predicting near 0 for every ground atom; so a model that predicts very low probabilities has a relatively high CLL even when these probabilities are incorrectly ordered.

We implemented SR2LR and the baselines as part of the Alchemy system [Kok *et al.*, 2005], and used the implementation of MTAMAR available from `http://www.cs.utexas.edu/users/ml/mlns/`. Θ in Algorithm 1 was set to 1. Inference during testing was performed on the mega-examples other than the one supplying training data, iterating over the available test examples. Within the same experiment,

[3] UW-CSE is available from `http://alchemy.cs.washington.edu/`. IMDB and WebKb are available from `http://www.cs.utexas.edu/users/ml/mlns/`.

[4] Source MLNs are available from `http://www.cs.utexas.edu/usrs/ml/mlns/` under SR2LR.

all systems used the same sequence of training and testing examples. The performance of a given predicate was evaluated by inferring probabilities for all of its groundings, given the truth values of all other predicates in the test mega-example as evidence. While training occurs on limited data, we test on a full mega-example. This is appropriate because the final goal of transfer is to obtain a model that gives effective predictions in the target domain as a whole and not just for an isolated entity. For inference, we used the Alchemy implementation of MC-SAT [Poon and Domingos, 2006] with the default parameter settings. Statistical significance was measured via a paired t-test at the 95% level. As a final note, all systems we compared ran extremely efficiently and found mappings in seconds on a standard workstation.

4.2 Overall Performance

The first set of experiments evaluates the relative accuracy of SR2LR over *all* predicates in each domain in the most challenging case when information about a single entity from the target domain is available. We formed single-entity-centered examples by randomly selecting as the central entity 10% of the entities of type person from each mega-example available in the target domain. This resulted in 29 entities in IMDB, 58 in UW-CSE, and 147 in WebKb. We compared against MTAMAR and a **Scratch** baseline that learns with no transfer as follows. For every ordered pair of known atoms in the available data, a clause is formed by having the first atom imply the second and variablizing consistently. All clauses obtained in this way are assigned a weight of 1. This baseline generates a set of informative clauses (in the terminology of Section 3) that are true in the given data.[5] Thus, it can be viewed as a variation of SR2LR that transfers only the short-range clauses of a source model that contains of all possible clauses of length 2.

Tables 1 and 2 list the accuracies for every possible target/source pair in terms of AUC-PR and CLL respectively. Significant improvement (degradation) over MTAMAR is indicated by a ↑ (↓), and significant improvement (degradation) over Scratch is indicated by ↗ (↙). In terms of AUC-PR, the more informative measure, transfer between UW-CSE and IMDB is always beneficial over learning from scratch, and SR2LR always has a significant advantage over MTAMAR. As expected, transfer to or from WebKb and the other two domains leads to less consistent gains and, in some cases, degradation. SR2LR is competitive also in terms of CLL, although in some cases, as discussed earlier, a model that gives significant advantages in AUC-PR is at a disadvantage in CLL.

Target	Source	MTAMAR	Scratch	SR2LR
IMDB	UW-CSE-learned	0.327	0.276	0.452 ↑ ↗
IMDB	UW-CSE-manual	0.414	0.276	0.577 ↑ ↗
IMDB	WebKb-learned	0.388	0.276	0.468 ↑ ↗
UW-CSE	IMDB-learned	0.115	0.108	0.188 ↑ ↗
UW-CSE	WebKb-learned	0.199	0.108	0.174 ↓ ↗
WebKb	IMDB-learned	0.164	0.287	0.168 ↑ ↙
WebKb	UW-CSE-learned	0.297	0.287	0.295
WebKb	UW-CSE-manual	0.276	0.287	0.178 ↓ ↙

Table 1: Average AUC-PR over all target domain predicates.

Target	Source	MTAMAR	Scratch	SR2LR
IMDB	UW-CSE-learned	-1.692	-4.575	-0.682 ↑ ↗
IMDB	UW-CSE-manual	-0.433	-4.575	-0.502 ↓ ↗
IMDB	WebKb-learned	-0.728	-4.575	-0.872 ↓ ↗
UW-CSE	IMDB-learned	-2.057	-5.708	-0.606 ↑ ↗
UW-CSE	WebKb-learned	-1.191	-5.708	-0.891 ↑ ↗
WebKb	IMDB-learned	-1.731	-3.440	-0.694 ↑ ↗
WebKb	UW-CSE-learned	-1.221	-3.440	-0.643 ↑ ↗
WebKb	UW-CSE-manual	-0.561	-3.440	-0.873 ↓ ↗

Table 2: Average CLL over all target domain predicates.

4.3 Focus on Specific Predicates

We have shown that, over all predicates in a domain, SR2LR can lead to significant gains in accuracy. Next, we study in greater detail the performance on the workedUnder predicate in IMDB and advisedBy in UW-CSE, which, as argued earlier, require more data to be learned from scratch, and are best predicted by long-range clauses. We used the single-entity-centered instances from Section 4.2 and introduced an additional **SR-Only** baseline that uses SR2LR to transfer only the short-range clauses, ignoring the long-range ones. This baseline is used to verify that transferring the long-range clauses is beneficial. Significant improvement (degradation) of SR2LR over SR-Only is indicated by a ⇑ (⇓). As shown in Table 3, when transferring to IMDB from UW-CSE, SR2LR outperforms significantly all other methods. SR2LR also leads to significant gains in transfer from IMDB to UW-CSE, although in this case SR2LR is significantly better than SR-Only just on CLL, equalling its performance on AUC-PR. Transferring from IMDB to UW-CSE is less beneficial than going in the opposite direction, from UW-CSE to IMDB, because several predicates in UW-CSE do not have analogs in IMDB while most of IMDB's predicates have a matching predicate in UW-CSE. As before, transfer from the more distantly related WebKb domain produces mixed results.

4.4 Increasing Numbers of Entities

In our final set of experiments, we compared the accuracy of SR2LR versus that of MTAMAR on workedUnder and advisedBy, as information about more entities becomes available. To do this, we considered 5 distinct orderings of the constants of type person in each mega-example, and provided the first n to the systems, with n ranging from 2 to 40 in IMDB, where the smallest mega-example has 44 constants of type person and from 2 to 50 in UW-CSE, where the smallest mega-example has 56 such constants. Each point on the curves is the average over all training instances with that many known entities. The results in terms of AUC-PR are shown in Fig. 5. These curves mirror the CLL results, which are omitted for space. As can be seen, SR2LR maintains its effectiveness even as more data becomes available. Surprisingly, MTAMAR's performance actually decreases as more entities become known. This is due to the fact that a larger number of known entities translates to a larger number of possible relations among them. If the known entities are disconnected, MTAMAR does not observe any instances in which mappings of the long-range clauses are helpful and therefore rejects them. Instead, it accepts mappings of the short-range clauses for which there is more evidence of usefulness. SR2LR is not susceptible to this because it treats

[5] If a clause has groundings that are violated by the data, then our construction procedure guarantees that there will be another clause with the same weight of 1, which draws the opposite conclusion. Thus, clauses that are not always true in the data cancel each other in pairs during inference.

Source	MTAMAR	SR-only	Scratch	SR2LR
UW-CSE-manual	0.726	0.339	0.032	0.982 ↑ ⇑ ↗
UW-CSE-learned	0.024	0.215	0.032	0.239 ↑ ⇑ ↗
WebKb-learned	0.025	0.023	0.032	0.023 ↓ ↙

Source	MTAMAR	SR-only	Scratch	SR2LR
IMDB-learned	0.010	0.030	0.008	0.030 ↑ ↗
WebKb-learned	0.007	0.007	0.008	0.007 ↙

Table 3: AUC-PR for `workedUnder` in IMDB (top) and `advisedBy` in UW-CSE (bottom).

Source	MTAMAR	SR-only	Scratch	SR2LR
UW-CSE-manual	-0.084	-0.066	-6.488	-0.037 ↑ ⇑ ↗
UW-CSE-learned	-0.385	-0.695	-6.488	-0.727 ↓ ⇓ ↗
WebKb-learned	-0.728	-0.700	-6.488	-0.700 ↑ ↗

Source	MTAMAR	SR-only	Scratch	SR2LR
IMDB-learned	-1.767	-0.295	-5.542	-0.280 ↑ ⇑ ↗
WebKb-learned	-0.757	-0.696	-5.542	-0.696 ↑ ↗

Table 4: CLL for `workedUnder` in IMDB (top) and `advisedBy` in UW-CSE (bottom).

long-range and short-range clauses separately. This effect is not observed in the smaller IMDB domain where randomly chosen entities are much less likely to be disconnected.

5 Related Work

This paper is most closely related to [Mihalkova *et al.*, 2007] and [Davis and Domingos, 2008], in that it also considers transfer of MLNs. However, both of these earlier works assume at least one full target domain mega-example is provided. Mapping source knowledge to a target domain is also addressed by the structure-mapping engine (SME) [Forbus and Oblinger, 1990]. SME evaluates predicate mappings based on a syntactic, structural criterion called *systematicity* and does not consider the accuracy of the resulting inferences in the target data. By contrast, TAMAR and SR2LR evaluate mappings primarily based on whether they produce empirically adequate clauses in the target domain.

6 Conclusion and Future Work

We presented SR2LR, an effective algorithm for mapping knowledge when target domain data is extremely limited and consists of a handful of disconnected entities, in the extreme case just one. This setting has not been studied before despite the fact that successful transfer could have the greatest impact in it. Our experiments demonstrate SR2LR's significant improvements over MTAMAR, which, unlike SR2LR, does not have a mechanism for coping with the large amount of missing data, as well as over other baselines.

In the future, we plan to experiment with novel ways of mapping source knowledge, such as mapping different arity predicates to one another, as well as mapping the arguments in different orders. Finally, we would like to explore ways of mapping a conjunction of two source predicates to a single target predicate and vice versa, thus performing a sort of transfer-motivated predicate invention.

Acknowledgement

We thank Tuyen Huynh and the three anonymous reviewers for their comments, as well as the Alchemy team for their continued support of the Alchemy package. This work is partially supported by DARPA grant FA8750-05-2-0283 (managed by AFRL) and by a research gift from Microsoft. The experiments were run on the Mastodon Cluster, provided by NSF Grant EIA-0303609.

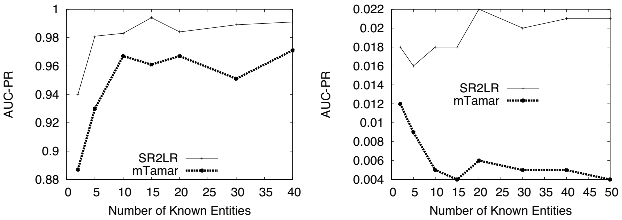

Figure 5: Accuracy on increasing amounts of data on `workedUnder` (left) and `advisedBy` (right).

References

[Banerjee *et al.*, 2006] B. Banerjee, Y. Liu, and G. M. Youngblood. ICML workshop on "Structural Knowledge Transfer for Machine Learning" 2006.

[Davis and Domingos, 2008] J. Davis and P. Domingos. Deep transfer via second-order markov logic. In *Proceedings of the AAAI Workshop on Transfer Learning For Complex Tasks*. 2008.

[Dehaspe, 1997] L. Dehaspe. Maximum entropy modeling with clausal constraints. *(ILP-97)*.

[Forbus and Oblinger, 1990] Kenneth D. Forbus and Dan Oblinger. Making SME greedy and pragmatic. (CogSci-90).

[Getoor and Taskar, 2007] L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning*. MIT Press, Cambridge, MA, 2007.

[Kok and Domingos, 2005] S. Kok and P. Domingos. Learning the structure of Markov logic networks. *(ICML-2005)*.

[Kok *et al.*, 2005] S. Kok, P. Singla, M. Richardson, and P. Domingos. The Alchemy system for statistical relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, 2005. `http://www.cs.washington.edu/ai/alchemy`.

[Mihalkova and Mooney, 2007] L. Mihalkova and R. J. Mooney. Bottom-up learning of Markov logic network structure. *(ICML-2007)*.

[Mihalkova *et al.*, 2007] L. Mihalkova, T. Huynh, and R. J. Mooney. Mapping and revising Markov logic networks for transfer learning. *(AAAI-07)*.

[Muggleton, 1996] S. Muggleton. Stochastic logic programs. *(ILP-96)*.

[Niculescu-Mizil and Caruana, 2007] A. Niculescu-Mizil and R. Caruana. Inductive transfer for Bayesian network structure learning. *(AISTATS-07)*.

[Poon and Domingos, 2006] H. Poon and P. Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. *(AAAI-06)*.

[Raina *et al.*, 2006] R. Raina, A. Y. Ng, and D. Koller. Constructing informative priors using transfer learning. *(ICML-2006)*.

[Richardson and Domingos, 2006] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[Silver *et al.*, 2005] D. Silver, G. Bakir, K. Bennett, R. Caruana, M. Pontil, S. Russell, and P. Tadepalli. NIPS workshop on "Inductive Transfer : 10 Years Later", 2005.

[Taylor *et al.*, 2008] M. E. Taylor, A. Fern, and K. Driessens. AAAI workshop on "Transfer Learning for Complex Tasks", 2008.

[Torrey *et al.*, 2007] L. Torrey, J. Shavlik, T. Walker, and R. Maclin. Relational macros for transfer in reinforcement learning. *(ILP-07)*.