In *Proceedings of the AAAI-08 Workshop on Transfer Learning For Complex Tasks,*. Chicago, IL, July 2008.

# Transfer Learning by Mapping with Minimal Target Data

**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

This paper introduces the single-entity-centered setting for transfer across two relational domains. In this setting, target domain data contains information about only a single entity. We present the SR2LR algorithm that finds an effective mapping of the source model to the target domain in this setting and demonstsrate its effectiveness in three relational domains. Our experiments additionally show that the most accurate model for the source domain is not always the best model to use for transfer.

## Introduction

Much of the work in transfer learning assumes that the source and target domains are described in terms of identical feature sets. However, sometimes knowledge needs to be transferred across domains that use different representations. In such cases, it is necessary to first map, or translate, the source model to the representation used in the target domain, before it can be further revised to improve its accuracy.

Addressing the mapping problem is especially important when using statistical relational learning (SRL) (Getoor & Taskar 2007) to perform transfer across relational domains. In a relational domain, there are a set of entities that participate in a variety of relationships. For example, in a domain describing an academic institution, e.g. (Richardson & Domingos 2006), the entities may be people, publications, and courses, whereas the relations may be `advised-by`, `taught-by`, `written-by`. In addition to relations of several entities, there may be unary relations that describe a single entity, such as `is-student` or `is-professor`. Several aspects of learning from relational data contribute to the complexity of the problem. In particular, a training instance is usually very large because the relations among entities may make it impossible to break an instance into smaller independent pieces. To emphasize this fact, we will call relational training instances *mega-examples*. Thus, in an academic domain, a mega-example may describe an entire area of study, such as AI. Moreover, individual mega-examples usually vary in size, e.g. the number of people, publications and courses and the relations among them in an AI program is unlikely to be identical to that in a Systems program.



The problem of finding a mapping, or a correspondence, between the relations in the source and target domains arises when these domains contain different sets of relations, such as when transferring knowledge learned in an academic domain to a domain about movies. We addressed this problem in our previous work (Mihalkova, Huynh, & Mooney 2007) where we presented TAMAR, an algorithm that transfers relational knowledge by first mapping and then revising it. The best way of mapping the source model is found by evaluating all valid mappings to the target data. It is therefore essential that TAMAR be provided with a full relational instance about the target domain. In cases when only partial data is available, TAMAR is likely to perform suboptimally.

In contrast to our previous work, here we present an approach to mapping source knowledge when minimal target-domain data is available. In particular, our approach addresses the *single-entity-centered* setting in which the learner is provided with information concerning only a single entity. To develop an intuition for how scarce this information is, consider the example in Fig. 1. It is centered around one entity in the domain, Bob, and thus all facts involve Bob. We assume that all true facts about Bob are listed; any fact about Bob that is not listed is false, e.g. `is-professor(Bob)` is false. However, no information about the other entities in the domain is provided or assumed, e.g. we do not know anything more about Ann apart from her relation to Bob. Fig. 2 contrasts the present paper to our previous work in terms of the amount of target domain data assumed to be available. Naturally, when target domain data is that limited, the source and target tasks need to be sufficiently related for transfer to be effective.

Like in our previous work, we consider transferring a Markov logic network (MLN) (Richardson & Domingos 2006) from a source domain to a target domain. An MLN consists of a set of first-order clauses, each of which has an attached weight. Roughly speaking, the weight of a clause determines how much more likely is a situation in which the clause is satisfied over a situation in which it is not satisfied.

Learning an MLN from scratch in the single-entity-centered setting is infeasible. For example, when learning to predict the `advisedBy` relation, one useful clause may be that if a student co-wrote a paper with a professor, then the student is advised by the professor. Our target data, however, does not contain examples of professors or of other people

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

Figure 1: Example target data provided to the learner.

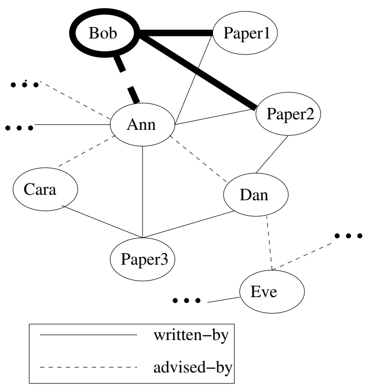


Figure 2: Target data available in previous versus current work. The nodes in this graph represent the entities in the domain and the edges represent the relations in which these entities participate. In our previous work we assumed that the information from the entire graph is available to the learner. In the present paper, we assume that only the bold relations are known.

who wrote papers and thus this clause cannot be learned. Therefore, applying transfer learning to this setting is a natural approach to take. In fact, it seems to be the approach taken by people when they relate their previous experiences to a new environment. Consider, for instance, a situation in which an actor A, who is intimately familiar with the workings of the movie business but has never been to college, first encounters the academic world by meeting a single representative from academia, professor P. Professor P talks about the graduate students she advises, the publications she has written, and the courses she has taught. This information, which is entirely centered on a single individual from academia, will be sufficient for actor A to form a fairly accurate idea of how the academic world works. He may imagine that advising a student is similar to instructing an actor when directing a movie, thus drawing an analogy between professors and movie directors, between actors and students, and between publications and movies. From this, actor A may further conclude that professor P's students are also listed in the "credits" of some of her publications, thus drawing conclusions about entities (i.e. the students) of which he has never seen examples.

Our algorithm 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 translations of the short-range clauses to the target domain can be evaluated on the available target-domain data, the key is to use the short-range clauses in order to find meaningful correspondences between the relations in the two domains. These correspondences are then used to translate the long-range clauses, thus boosting the performance of the model in the target domain. We call our new method Short-Range to Long-Range (SR2LR ).

## Background

In first-order logic, a *predicate* represents a relation in the domain, such as `advised-by`. Predicates can be viewed as functions that return true or false in which each argument has a *type*. For example, `advised-by` takes two arguments of type person, whereas `written-by` takes an argument of type paper and an argument of type person. An *atom* is a predicate applied to terms, where the terms can be variables or constants.[1] Here, we will call constants *entities*. A (negative/positive) *literal* is an atom that (is/is not) negated. A literal in which all terms are entities is *ground*. A *fact* is a ground positive literal. A *clause* is a disjunction of literals. 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 & 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 $P(\mathbf{X}=\mathbf{x}) = \frac{1}{Z}\exp\left(\sum_{f_i\in\mathcal{F}} w_i \sum_{g\in\mathcal{G}_{f_i}} g(\mathbf{x})\right)$.

In our previous work (Mihalkova, Huynh, & Mooney 2007) we presented TAMAR, an algorithm that transfers MLNs by mapping and revising them. We will review MTAMAR, the mapping portion of TAMAR, to which we will compare SR2LR . MTAMAR uses the concept of a *type-consistent* mapping. In order to map a source clause to the target domain, one needs to map each predicate in the clause to a predicate in the target domain. When mapping a source predicate to a target predicate, one implicitly defines a mapping between the types of the arguments of the two predicates. A mapping is type-consistent if a type in the source domain is mapped to at most one type in the target domain over all predicate mappings within a clause. MTAMAR maps each source clause independently of the others by evaluating all possible type-consistent mappings with the weighted pseudo log-likelihood score introduced by Kok and Domingos (2005). This measure assumes that at least one full target relational 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 only single-entity-centered target data is infeasible. Therefore, we will not use the revision portion of TAMAR, and SR2LR does not perform revision.

## Our Approach

This section describes the SR2LR algorithm. SR2LR is designed for the case when target-domain data is available only

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

| | | |
|---|---|---|
| 1.1 | advised-by$(a, b) \Rightarrow \neg$is-professor$(a)$ | worked-for $\rightarrow$ advised-by, is-director $\rightarrow$ is-professor |
| 1.2 | advised-by$(a, b) \Rightarrow \neg$is-student$(a)$ | worked-for $\rightarrow$ advised-by, is-director $\rightarrow$ is-student |
| 2.1 | written-by$(m, a) \wedge$ written-by$(m, b) \wedge$ is-professor$(b)$ $\Rightarrow$ advised-by$(a, b)$ | worked-for $\rightarrow$ advised-by, is-director $\rightarrow$ is-professor, in-movie $\rightarrow$ written-by |
| 2.2 | written-by$(m, a) \wedge$ written-by$(m, b) \wedge$ is-student$(b)$ $\Rightarrow$ advised-by$(a, b)$ | worked-for $\rightarrow$ advised-by, is-director $\rightarrow$ is-student, in-movie $\rightarrow$ written-by |

Table 1: Example mapped clauses with predicate correspondences

in the form of a single-entity-centered example.

**Definition 1** *A training example is* **single-entity-centered** *with respect to an entity $E$ if it lists all the true facts involving $E$, and only those facts. Facts that involve $E$ but are not listed are assumed to be false. Facts that do not involve $E$ have unknown truth values. $E$ is called the* **central** *entity.*

SR2LR starts by producing all possible type-consistent mappings of the clauses in the source model. Type-consistent mappings are defined in the Background section. As we discussed in that section, a mapping from a source clause to the target domain implies a correspondence from the source predicates that appear in the source clause to a subset of the target predicates. MTAMAR evaluates the usefulness of this correspondence by measuring the performance of the mapped clause in the target domain based on a probabilistic score. In the single-entity-centered setting, however, not all mapped clauses can be evaluated in this way because the target data is incomplete. The key idea of SR2LR is to find valid source-to-target predicate correspondences by directly evaluating only the clauses whose performance can be measured on the available data and then to use these correspondences to produce mappings for clauses whose performance cannot be directly measured in the target domain. Mapped clauses that can be directly evaluated are called short-range; the rest are called long-range.

**Definition 2** *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 $t$. We call any such variable $v$* **pivotal**.

**Definition 3** *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. 3 using the data in Fig. 1, i.e. transfer from a toy movie domain to a toy academic domain. Let us consider one possible type-consistent mapping of the first clause, given in line 1.1 of Table 1. Note that if we ground this clause using the substitution $a = Bob, b = Ann$, we obtain a ground clause whose literals are all known from our data, thus the clause can be evaluated and hence, it is short-range. 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 grounding is verifiable, whereas the second is not.

**Definition 4** *A ground short-range clause is* **verifiable** *on a single-entity-centered example if a pivotal variable is replaced by the central entity.*

Now consider one possible mapping of the second clause in Fig. 3, given in line 2.1 of Table 1. According to definitions 2 and 3, this clause is long-range. An intuitive interpretation is that this clause concerns relations that go beyond just a single entity, e.g. about papers written by other people.

$$
\begin{aligned}
0.7: &\ \text{worked-for}(a, b) \Rightarrow \neg\text{is-director}(a) \\
0.8: &\ \text{in-movie}(m, a) \wedge \text{in-movie}(m, b) \wedge \text{is-director}(b) \\
&\qquad \Rightarrow \text{worked-for}(a, b)
\end{aligned}
$$

Figure 3: Source MLN from toy movie domain

**Algorithm 1** SR2LR algorithm

**Input:**
1: SM: Source Markov logic network
2: $T_E$: Single-entity-centered target data
3: E: Central entity
4: $\mathcal{P}$: Set of predicates in the target domain
5: $\Theta$: Truth threshold for accepting a short-range clause

**Output:**
6: `Result`: Markov logic network for the target domain

**Procedure:**
7: Generate TM, the set of all possible type-consistent mappings of the clauses in SM. Each mapped clause is given the weight of its corresponding source clause.
8: Split the clauses in TM into sets $\mathcal{S}$ of short-range clauses and $\mathcal{L}$ of long-range clauses.
9: $\mathcal{S}'$ = filter-short-range($\mathcal{S}, \Theta$)
10: Add all clauses from $\mathcal{S}'$ to `Result`
11: $\mathcal{L}'$ = filter-long-range($\mathcal{L}, \mathcal{S}'$)
12: Add all clauses from $\mathcal{L}'$ to `Result`
13: Let $\mathcal{A}_C$ be the set of all clauses in `Result` mapped from source clause $C$ with weight $w_C$.
14: Set the weight of each $a \in \mathcal{A}_C$ to $w_C/|\mathcal{A}_C|$.

Algorithm 1 formally lists the steps SR2LR takes. In line 9, the short-range mapped clauses are evaluated, as described in Algorithm 2. Because of the restricted nature of the available target data, rather than using a probabilistic measure, we simply check whether the verifiable groundings of short-range clauses are satisfied on the target data. Clauses that are satisfied at least $\Theta$ proportion of time are accepted; the rest are rejected. This procedure automatically rejects clauses that are not informative, as defined next.

**Definition 5** *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.*

For example, consider the clause `is-student`$(a) \vee \neg$`advised-by`$(b, a)$. This clause 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 ground-

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

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

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

$\mathcal{L}' = \emptyset$
**for each** $LR \in \mathcal{L}$ **do**
  **for each** $SR \in \mathcal{S}'$ **do**
    **if** $SR$ supports $LR$ **then**
      Mark the corresponding literals in $LR$ as "supported"
    **end if**
  **end for**
  **if** All literals in $C$ are supported **then**
    Add $C$ to $\mathcal{L}'$
  **else if** All unsupported literals are supported by exclusion wrt $\mathcal{S}'$ **then**
    Add $C$ to $\mathcal{L}'$
  **end if**
**end for**
Return $\mathcal{L}'$

ings, i.e. is-student$(Bob)$, $\neg$advised-by$(Ann, Bob)$ and $\neg$advised-by$(Bob, Bob)$, are true. To develop an intuition for the significance of definition 5, consider one of the verifiable 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 as advised-by$(Ann, Bob) \Rightarrow$ is-student$(Bob)$. Thus clauses that are not informative cannot be used to draw inferences, i.e. they are always *trivially* satisfied because their preconditions do not hold. Therefore, judgements about predicate mappings based on clauses that are not informative are likely to be misleading.

Once the short-range clauses are evaluated, in line 11 of the main algorithm, SR2LR evaluates the long-range ones, based on the source-to-target predicate correspondences found to be useful for short-range clauses. Algorithm 3 uses the following definitions.

**Definition 6** *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 $C_S$* **supports** *$C_L$. The literals of predicates in $C_L$ that also appear in $C_S$ are said to be* **supported**.

**Definition 7** *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 source clauses to be mapped. Even though support by exclusion may seem too risky in the sense that a pair of completely unrelated source and target predicates may be 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.

We now illustrate the operation of Algorithm 1 up to line 13. Table 1 lists some of the possible ways of mapping the clauses in Fig. 3, along with the source-to-target predicate correspondences implied by them. Clauses 1.* in are (informative) short-range, while 2.* are long-range. Let $\Theta = 1$. All verifiable groundings of clause 1.1 are satisfied by the target data. Thus, this clause is accepted in the resulting model and the predicate correspondences found by it are useful. Clause 1.2 is rejected because not all of its groundings are satisfied by the target data. Thus the set $\mathcal{S}'$ 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 cannot be accepted because there is no support for is-student(b).

Finally, in lines 13-14 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. We found that performing the weight adjustment led to better performance. These experimental results are omitted because of space considerations.

## Experiments

We compared SR2LR to MTAMAR on the three domains we used previously (Mihalkova, Huynh, & Mooney 2007): IMDB, UW-CSE, and WebKb,[2] treating each possible ordered pair of them as a source and a target. WebKb and UW-CSE are about academics. IMDB is about movies. UW-CSE contains the predicates taughtBy, courseLevel, position, advisedBy, projectMember, phase, tempAdvisedBy, yearsInProgram, TA ,student, professor, samePerson, sameCourse, sameProject, publication; IMDB the predicates director, actor, movie, gender, workedUnder, genre, samePerson, sameMovie, sameGenre, sameGender; and WebKB the predicates student, faculty, project, courseTA, courseProf. Although some predicates have the same names in different domains, predicate names are not used by either system.

We used two sets of source MLNs. The first set was learned using BUSL (Mihalkova & Mooney 2007); the second set was learned using a *relaxed* version of BUSL that learns models that are less accurate in the source domain but contain more clauses. The relaxed set of sources was motivated by the hypothesis that the best-performing model in the source domain is not necessarily the best model to use for transfer. We also experimented with using the manually developed knowledge base provided with UW-CSE as a source (we call this *hand* in our experiments). We removed

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

| | from UW-CSE busl | | from UW-CSE relaxed | | from UW-CSE hand | |
|---|---|---|---|---|---|---|
| | mTamar | SR2LR | mTamar | SR2LR | mTamar | SR2LR |
| director | 0.95 | **0.95** | 1.00 | 1.00 | 0.59 | 0.61 |
| actor | 0.53 | **0.91** | 1.00 | 1.00 | 0.98 | 0.95 |
| movie | 0.12 | **0.15** | 0.22 | **0.28** | 0.23 | **0.31** |
| gender | 0.24 | **0.46** | 0.33 | **0.45** | 0.37 | **0.54** |
| workedUnder | 0.03 | 0.01 | 0.04 | **0.39** | 0.51 | **0.94** |
| genre | 0.02 | 0.02 | 0.03 | 0.01 | 0.44 | **0.54** |
| sameMovie | 0.08 | **0.12** | 0.09 | **0.12** | 0.08 | **0.12** |
| sameGenre | 0.05 | **0.07** | 0.06 | **0.08** | 0.06 | **0.07** |
| sameGender | 0.11 | **0.25** | 0.14 | **0.58** | 0.11 | **0.25** |

| | from Webkb busl | | from Webkb relaxed | |
|---|---|---|---|---|
| | mTamar | SR2LR | mTamar | SR2LR |
| director | 0.95 | **0.99** | 0.63 | **0.74** |
| movie | 0.22 | **0.28** | 0.22 | **0.28** |
| gender | 0.17 | **0.26** | 0.17 | **0.26** |
| samePerson | 0.90 | 0.90 | 0.90 | **1.00** |
| sameMovie | 0.08 | **0.12** | 0.08 | **0.12** |
| sameGenre | 0.05 | **0.07** | 0.05 | **0.07** |
| sameGender | 0.11 | **0.25** | 0.11 | **0.25** |

Table 2: AUC for transfer to IMDB. Bold results are significantly better than the performance of MTAMAR given the same source.

from this knowledge base all clauses that refer to specific entities in UW-CSE because SR2LR and MTAMAR do not map entities. We used generative weight-training to learn weights for the clauses in the knowledge base in the source domain (Richardson & Domingos 2006). Target training data in each case consisted of a single-entity-centered example, where the central entity was one of the entities of type person in the domain. In IMDB, the average size of a training example was 6.26 true ground facts; in UW-CSE it was 10.40, and in Webkb it was 2.66. The size of the full mega-example in IMDB was 320 true ground facts; in UW-CSE 766 true ground facts, and in WebKb 519 true ground facts. We report average performances over all entities of type person in each domain.

We evaluated the performance of the mapped MLNs by performing leave-one-predicate-out inference over them in the full mega-example, as done in previous work (Kok & Domingos 2005; Mihalkova, Huynh, & Mooney 2007). For inference we used MC-SAT (Poon & Domingos 2006). The results are reported in terms of the area under the precision-recall curve (AUC). This measure 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 in the data.

We implemented SR2LR as part of the Alchemy package (Kok *et al.* 2005), and used the implementation of MTAMAR available at `http://www.cs.utexas.edu/users/ml/mlns/`. We set $\Theta = 1$.

## Results

Our experiments address the following questions:

1. In which cases does SR2LR outperform MTAMAR?
2. How does the choice of source model affect performance?

To answer the first question, in Tables 2-4, we report the performance of the systems for predicting each predicate separately. We only show the predicates for which MTAMAR performs better than SR2LR or for which SR2LR is significantly better than MTAMAR. Statistical significance was measured using a paired t-test at the $95\%$ confidence level. Predicates for which SR2LR performs at least as well as MTAMAR but the difference is not significant are omitted to save space.

SR2LR achieves the greatest performance gains in the UW-CSE $\rightarrow$ IMDB experiment. This is not surprising because each predicate in IMDB has a corresponding predicate in UW-CSE. Thus, this demonstrates that SR2LR is capable of discovering useful predicate correspondences. When transferring from a domain with a richer predicate set to one with fewer predicates, there invariably will be target predicates to which no useful correspondences were found. This explains the more modest performance gains observed when transferring to UW-CSE or from WebKb. Because WebKb has the smallest number of predicates, which seem to have obvious correspondences to predicates in UW-CSE, it may seem surprising that we do not observe bigger performance gains in transfer to WebKb. This behavior can be explained if we look more closely at the WebKb predicates where performance is especially poor: courseTA(coursename, person) and courseProf(coursename,person). Their obvious counterparts in UW-CSE are TA(course, person, semester) and taughtBy(course, person, semester). As we see, the UW-CSE predicates have arity 3, whereas the WebKb ones have arity 2. At present, SR2LR maps only predicates with the same arity to each other; in the future we plan to experiment with less restrictive clause-mapping techniques.

To answer the second question, we observe that transfer from the "relaxed" sources can give better results, even though the "busl" sources perform much better in the source domains. This demonstrates that the best model in the source domain is not necessarily the best one for transfer. In the single-entity-centered scenario, sources that have more short-range clauses lead to better results.

## Related Work

The problem of mapping a source model to a target domain has also been addressed by others. For example, the Structure-Mapping Engine (Falkenhainer, Forbus, & Gentner 1989) uses reasoning by analogy to find correspondences between source and target predicates. In the area of reinforcement learning, Taylor *et al.* (2008) present a method for mapping the feature spaces across tasks by using samples from runs in the two tasks.

Our work is also related to the problem of transfer learning within the SRL setting. Guestrin *et al.* (2003) use a relational representation as a vehicle to achieving transfer by learning general rules about the ways objects interact. Recently, Deshpande *et al.* (2007) presented a hierarchical Bayesian approach to transferring knowledge about probabilistic planning rules. Torrey *et al.* (2007) use SRL tech-

| | from IMDB busl | | from IMDB relaxed | |
|---|---|---|---|---|
| | mTamar | SR2LR | mTamar | SR2LR |
| courseLevel | 0.33 | 0.25 | 0.24 | 0.25 |
| position | 0.20 | **0.36** | 0.05 | **0.31** |
| advisedBy | 0.02 | 0.02 | 0.01 | **0.03** |
| phase | 0.20 | **0.28** | 0.15 | **0.28** |
| yearsInProgram | 0.03 | **0.09** | 0.05 | **0.09** |
| student | 0.87 | 0.89 | 0.78 | **0.95** |
| professor | 0.41 | 0.30 | 0.32 | **0.92** |
| samePerson | 0.02 | 0.01 | 0.01 | 0.01 |
| sameCourse | 0.02 | 0.02 | 0.03 | 0.02 |
| publication | 0.02 | 0.02 | 0.03 | 0.02 |

| | from Webkb busl | | from Webkb relaxed | |
|---|---|---|---|---|
| | mTamar | SR2LR | mTamar | SR2LR |
| courseLevel | 0.16 | **0.25** | 0.16 | **0.25** |
| phase | 0.09 | **0.12** | 0.09 | **0.12** |
| student | 0.98 | 0.95 | 0.95 | 0.77 |
| professor | 0.93 | 0.79 | 0.64 | 0.41 |
| samePerson | 0.52 | **0.67** | 0.68 | **0.94** |
| publication | 0.02 | **0.06** | 0.04 | 0.03 |

Table 3: AUC for transfer to UW-CSE. Bold results are significantly better than the performance of MTAMAR given the same source.

| | from IMDB busl | | from IMDB relaxed | |
|---|---|---|---|---|
| | mTamar | SR2LR | mTamar | SR2LR |
| student | 0.46 | **0.67** | 0.46 | **0.67** |
| faculty | 0.21 | 0.15 | 0.21 | 0.15 |
| project | 0.02 | 0.01 | 0.03 | 0.01 |

| | from UW-CSE hand | | from UW-CSE busl | | from UW-CSE relaxed | |
|---|---|---|---|---|---|---|
| | mTamar | SR2LR | mTamar | SR2LR | mTamar | SR2LR |
| student | 0.91 | 0.80 | 0.38 | **0.76** | 0.98 | **0.99** |
| faculty | 0.35 | **0.66** | 0.12 | **0.24** | 0.96 | **0.98** |

Table 4: AUC for transfer to Webkb. Bold results are significantly better than the performance of MTAMAR given the same source.

niques to learn relational macros in the source task that then guide the initial stages of learning in the target task.

None of this previous work, however, addresses the single-entity-centered setting.

## Conclusion and Future Work

This paper introduces the single-entity-centered setting of transfer in which data about only one entity in the target domain is available. We presented the SR2LR algorithm which evaluates possible source-to-target predicate correspondences based on short-range clauses in order to also transfer the knowledge captured in long-range clauses.

The single-entity-centered setting can be viewed as one extreme on the spectrum of possible available relational data. On the other end of the spectrum is a full mega-example. In the future, we plan to study how the relative performance of SR2LR and MTAMAR changes as data about an increasing number of entities becomes available. We also plan to experiment with novel ways of mapping a source clause to the target domain. At present, we require that a source predicate be mapped to a single target predicate of the same arity, and we map the arguments of the predicates in order. We would like to consider mappings in which different arity predicates are mapped to each other by using techniques such as merging their arguments, as well as mappings in which the arguments are mapped 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

This work is partially supported by DARPA grant FA8750-05-2-0283 (managed by AFRL) and by a research gift from Microsoft.The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of DARPA, AFRL, Microsoft, or the United States Government. The experiments were run on the Mastodon Cluster, provided by NSF Grant EIA-0303609.

## References

Deshpande, A.; Milch, B.; Zettlemoyer, L. S.; and Kaelbling, L. P. 2007. Learning probabilistic relational dynamics for multiple tasks. UAI-07.

Falkenhainer, B.; Forbus, K. D.; and Gentner, D. 1989. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence* 41(1):1–63.

Getoor, L., and Taskar, B., eds. 2007. *Introduction to Statistical Relational Learning*. MIT Press.

Guestrin, C.; Koller, D.; Gearhart, C.; and Kanodia, N. 2003. Generalizing plans to new environments in relational MDPs. IJCAI-03.

Kok, S., and Domingos, P. 2005. Learning the structure of Markov logic networks. ICML-05.

Kok, S.; Singla, P.; Richardson, M.; and Domingos, P. 2005. The Alchemy system for statistical relational AI. Tech. report, Dept. of Comp. Sci. and Eng., University of Washington. `http://www.cs.washington.edu/ai/alchemy`.

Mihalkova, L., and Mooney, R. J. 2007. Bottom-up learning of Markov logic network structure. ICML-07.

Mihalkova, L.; Huynh, T.; and Mooney, R. J. 2007. Mapping and revising Markov logic networks for transfer learning. AAAI-07.

Poon, H., and Domingos, P. 2006. Sound and efficient inference with probabilistic and deterministic dependencies. AAAI-06,

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

Taylor, M. E.; Kuhlmann, G.; and Stone, P. 2008. Autonomous transfer for reinforcement learning. AAMAS-08.

Torrey, L.; Shavlik, J.; Walker, T.; and Maclin, R. 2007. Relational macros for transfer in reinforcement learning. ILP-07.