Learning Relational Features with Backward Random Walks

Ni Lao
Google Inc.
nlao@google.com

Einat Minkov
University of Haifa
einatm@is.haifa.ac.il

William W. Cohen
Carnegie Mellon University
wcohen@cs.cmu.edu

Abstract

The path ranking algorithm (PRA) has been recently proposed to address relational classification and retrieval tasks at large scale. We describe Cor-PRA, an enhanced system that can model a larger space of relational rules, including longer relational rules and a class of first order rules with constants, while maintaining scalability. We describe and test faster algorithms for searching for these features. A key contribution is to leverage backward random walks to efficiently discover these types of rules. An empirical study is conducted on the tasks of graph-based knowledge base inference, and person named entity extraction from parsed text. Our results show that learning paths with constants improves performance on both tasks, and that modeling longer paths dramatically improves performance for the named entity extraction task.

1 Introduction

Structured knowledge about entities and the relationships between them can be represented as an edge-typed graph, and relational learning methods often base predictions on connectivity patterns in this graph. One such method is the Path Ranking Algorithm (PRA), a random-walk based relational learning and inference framework due to Lao and Cohen (2010b). PRA is highly scalable compared with other statistical relational learning approaches, and can therefore be applied to perform inference in large knowledge bases (KBs). Several recent works have applied PRA to link prediction in semantic KBs, as well as to learning syntactic relational patterns used in information extraction from the Web (Lao et al., 2012; Gardner et al., 2013; Gardner et al., 2014; Dong et al., 2014).

A typical relational inference problem is illustrated in Figure 1. Having relational knowledge represented as a graph, it is desired to infer additional relations of interest between entity pairs. For example, one may wish to infer whether an *AthletePlaysInLeague* relation holds between nodes *HinesWard* and *NFL*. More generally, link prediction involves queries of the form: which entities are linked to a source node s (*HinesWard*) over a relation of interest r (e.g., r is *AlthletePlaysInLeague*)?

PRA gauges the relevance of a target node t with respect to the source node s and relation r based on a set of relation paths (i.e., sequences of edge labels) that connect the node pair. Each path π_i is considered as feature, and the value of feature π_i for an instance (s, t) is the probability of reaching t from s following path π_i. A classifier is learned in this feature space, using logistic regression.

PRA's candidate paths correspond closely to a certain class of Horn clauses: for instance, the path $\pi = \langle \textit{AthletePlaysForTeam}, \textit{TeamPlaysInLeague} \rangle$, when used as a feature for the relation $r = \textit{AthletePlaysForLeague}$, corresponds to the Horn clause

$$\textit{AthletePlaysForTeam}(s, z) \wedge \textit{TeamPlaysInLeague}(z, t) \rightarrow \textit{AthletePlaysForLeague}(s, t)$$

One difference between PRA's features and more traditional logical inference is that random-walk weighting means that not all inferences instantiated by a clause will be given the same weight. Another difference is that PRA is very limited in terms of expressiveness. In particular, inductive logic programming

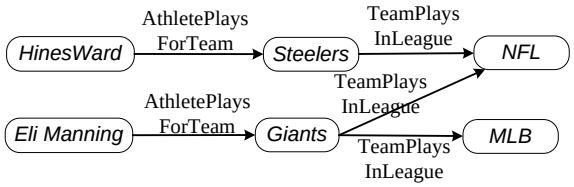

Figure 1: An example knowledge graph

(ILP) methods such as FOIL (Quinlan and Cameron-Jones, 1993) learn first-order Horn rules that may involve constants. Consider the following rules as motivating examples.

$$\textit{EmployeedByAgent}(s,t) \wedge \textit{IsA}(t, \textit{SportsTeam}) \rightarrow \textit{AthletePlaysForTeam}(s,t)$$

$$t = \textit{NFL} \rightarrow \textit{AthletePlaysForTeam}(s,t)$$

The first rule includes *SportsTeam* as a constant, corresponding to a particular graph node, which is a the semantic class (hypernym) of the target node t. The second rule simply assigns *NFL* as the target node for the *AthletePlaysForTeam* relation; if used probabilistically, this rule can serve as a prior. Neither feature can be expressed in PRA, as PRA features are restricted to edge type sequences.

We are interested in extending the range of relational rules that can be represented within the PRA framework, including rules with constants. A key challenge is that this greatly increases the space of candidate rules. Knowledge bases such as Freebase (Bollacker et al., 2008), YAGO (Suchanek et al., 2007), or NELL (Carlson et al., 2010a), may contain thousands of predicates and millions of concepts. The number of features involving concepts as constants (even if limited to simple structures such as the example rules above) will thus be prohibitively large. Therefore, it is necessary to search the space of candidate paths π very efficiently. More efficient candidate generation is also necessary if one attempts to use a looser bound on the length of candidate paths.

To achieve this, we propose using *backward random walks*. Given target nodes that are known to be relevant for relation r, we perform backward random walks (up to finite length ℓ) originating at these target nodes, where every graph node c reachable in this random walk process is considered as a potentially useful constant. Consequently, the relational paths that connect nodes c and t are evaluated as possible random walk features. As we will show, such paths provide informative class priors for relational classification tasks.

Concretely, this paper makes the following contributions. First, we outline and discuss a new and larger family of relational features that may be represented in terms of random walks within the PRA framework. These features represent paths with constants, expanding the expressiveness of PRA. In addition, we propose to encode bi-directional random walk probabilities as features; we will show that accounting for this sort of directionality provides useful information about graph structure.

Second, we describe the learning of this extended set of paths by means of backward walks from relevant target nodes. Importantly, the search and computation of the extended set of features is performed efficiently, maintaining high scalability of the framework. Concretely, using backward walks, one can compute random walk probabilities in a bi-directional fashion; this means that for paths of length $2M$, the time complexity of path finding is reduced from $O(|V|^{2M})$ to $O(|V|^{M})$, where $|V|$ is the number of edge types in graph.

Finally, we report experimental results for relational inference tasks in two different domains, including knowledge base link prediction and person named entity extraction from parsed text (Minkov and Cohen, 2008). It is shown that the proposed extensions allow one to effectively explore a larger feature space, significantly improving model quality over previously published results in both domains. In particular, incorporating paths with constants significantly improves model quality on both tasks. Bi-directional walk probability computation also enables the learning of longer predicate chains, and the modeling of long paths is shown to substantially improve performance on the person name extraction task. Importantly, learning and inference remain highly efficient in both these settings.

2 Related Work

ILP complexity stems from two main sources—the complexity of searching for clauses, and of evaluating them. First-order learning systems (e.g. FOIL, FOCL (Pazzani et al., 1991)) mostly rely on hill-climbing search,

i.e., incrementally expanding existing patterns to explore the combinatorial model space, and are thus often vulnerable to local maxima. PRA takes another approach, generating features using efficient random graph walks, and selecting a subset of those features which pass precision and frequency thresholds. In this respect, it resembles a stochastic approach to ILP used in earlier work (Sebag and Rouveirol, 1997).The idea of sampling-based inference and induction has been further explored by later systems (Kuželka and Železný, 2008; Kuželka and Železný, 2009).

Compared with conventional ILP or relational learning systems, PRA is limited to learning from binary predicates, and applies random-walk semantics to its clauses. Using sampling strategies (Lao and Cohen, 2010a), the computation of clause probabilities can be done in time that is independent of the knowledge base size, with bounded error rate (Wang et al., 2013). Unlike in FORTE and similar systems, in PRA, sampling is also applied to the induction path-finding stage. The *relational feature construction* problem (or propositionalization) has previously been addressed in the ILP community—*e.g.*, the RSD system (Železný and Lavrač, 2006) performs explicit first-order feature construction guided by an precision heuristic function. In comparison, PRA uses precision and recall measures, which can be readily read off from random walk results.

Bi-directional search is a popular strategy in AI, and in the ILP literature. The *Aleph* algorithm (Srinivasan, 2001) combines top-down with bottom-up search of the refinement graph, an approach inherited from Progol. FORTE (Richards and Mooney, 1991) was another early ILP system which enumerated paths via a bi-directional seach. Computing backward random walks for PRA can be seen as a particular way of bi-directional search, which is also assigned a random walk probability semantics. Unlike in prior work, we will use this probability semantics directly for feature selection.

3 Background

We first review the Path Ranking Algorithm (PRA) as introduced by (Lao and Cohen, 2010b), paying special attention to its random walk feature estimation and selection components.

3.1 Path Ranking Algorithm

Given a directed graph G, with nodes N, edges E and edge types R, we assume that all edges can be traversed in both directions, and use r^{-1} to denote the reverse of edge type $r \in R$. A *path type* π is defined as a sequence of edge types $r_1 \dots r_\ell$. Such path types may be indicative of an extended relational meaning between graph nodes that are linked over these paths; for example, the path $\langle \textit{AtheletePlaysForTeam}, \textit{TeamPlaysInLeague} \rangle$ implies the relationship "the league a certain player plays for". PRA encodes $P(s \to t; \pi_j)$, the probability of reaching target node t starting from source node s and following path π_j, as a feature that describes the semantic relation between s and t. Specifically, provided with a set of selected path types up to length ℓ, $\mathcal{P}_\ell = \{\pi_1, \dots, \pi_m\}$, the relevancy of target nodes t with respect to the query node s and the relationship of interest is evaluated using the following scoring function

$$score(s,t) = \sum_{\pi_j \in \mathcal{P}_\ell} \theta_j P(s \to t; \pi_j), \qquad (1)$$

where θ are appropriate weights for the features, estimated in the following fashion.

Given a relation of interest r and a set of annotated node pairs $\{(s,t)\}$, for which it is known whether $r(s,t)$ holds or not, a training data set $\mathcal{D} = \{(\mathbf{x}, y)\}$ is constructed, where $\mathbf{x}$ is a vector of all the path features for the pair (s,t)—i.e., the j-th component of $\mathbf{x}$ is $P(s \to t; \pi_j)$, and y is a boolean variable indicating whether $r(s,t)$ is true. We adopt the closed-world assumption—a set of relevant target nodes G_i is specified for every example source node s_i and relation r, and all other nodes are treated as negative target nodes. A biased sampling procedure selects only a small subset of negative samples to be included in the objective function (Lao and Cohen, 2010b). The parameters θ are estimated from both positive and negative examples using a regularized logistic regression model.

3.2 PRA Features–Generation and Selection

PRA features are of the form $P(s \to t; \pi_j)$, denoting the probability of reaching target node t, originating random walk at node s and following edge type sequence π_j. These path probabilities need to be estimated for every node pair, as part of both training and inference. High scalability

is achieved due to efficient path probability estimation. In addition, feature selection is applied so as to allow efficient learning and avoid overfitting.

Concretely, the probability of reaching t from s following path type π can be recursively defined as

$$P(s \rightarrow t; \pi) = \sum_z P(s \rightarrow z; \pi') P(z \rightarrow t; r), \quad (2)$$

where r is the last edge type in path π, and π' is its prefix, such that adding r to π' gives π. In the terminal case that π' is the empty path ϕ, $P(s \rightarrow z; \phi)$ is defined to be 1 if $s = z$, and 0 otherwise. The probability $P(z \rightarrow t; r)$ is defined as $1/|r(z)|$ if $r(z, t)$, and 0 otherwise, where $r(z)$ is the set of nodes linked to node z over edge type r. It has been shown that $P(s \rightarrow t; \pi)$ can be effectively estimated using random walk sampling techniques, with bounded complexity and bounded error, for all graph nodes that can be reached from s over path type π (Lao and Cohen, 2010a).

Due to the exponentially large feature space in relational domains, candidate path features are first generated using a dedicated particle filtering path-finding procedure (Lao et al., 2011), which is informed by training signals. Meaningful features are then selected using the following *goodness measures*, considering path precision and coverage:

$$precision(\pi) = \frac{1}{n} \sum_i P(s_i \rightarrow G_i; \pi), \quad (3)$$

$$coverage(\pi) = \sum_i I(P(s_i \rightarrow G_i; \pi) > 0). \quad (4)$$

where $P(s_i \rightarrow G_i; \pi) \equiv \sum_{t \in G_i} P(s_i \rightarrow t; \pi)$. The first measure prefers paths that lead to correct nodes with high average probability. The second measure reflects the number of queries for which some correct node is reached over path π. In order for a path type π to be included in the PRA model, it is required that the respective scores pass thresholds, $precision(\pi) \geq a$ and $coverage(\pi) \geq h$, where the thresholds a and h are tuned empirically using training data.

4 Cor-PRA

We will now describe the enhanced system, which we call Cor-PRA, for the Constant and Reversed Path Ranking Algorithm. Our goal is to enrich the space of relational rules that can be represented using PRA, while maintaining the scalability of this framework.

4.1 Backward random walks

We first introduce *backward* random walks, which are useful for generating and evaluating the set of proposed relational path types, including paths with constants. As discussed in Sec.4.4, the use of backward random walks also enables the modeling of long relational paths within Cor-PRA.

A key observation is that the path probability $P(s \rightarrow t; \pi)$ may be computed using forward random walks (Eq. (2)), or alternatively, it can be recursively defined in a *backward* fashion:

$$P(t \leftarrow s; \pi) = \sum_z P(t \leftarrow z; \pi'^{-1}) P(z \leftarrow s; r^{-1}) \quad (5)$$

where π'^{-1} is the path that results from removing the last edge type r in π'. Here, in the terminal condition that $\pi'^{-1} = \phi$, $P(t \leftarrow z; \pi'^{-1})$ is defined to be 1 for $z = t$, and 0 otherwise. In what follows, the starting point of the random walk calculation is indicated at the left side of the arrow symbol; i.e., $P(s \rightarrow t; \pi)$ denotes the probability of reaching t from s computed using forward random walks, and $P(t \leftarrow s; \pi)$ denotes the same probability, computed in a backward fashion.

4.2 Relational paths with constants

As stated before, we wish to model relational rules that may include constants, denoting related entities or concepts. Main questions are, how can relational rules with constants be represented as path probability features? and, how can meaningful rules with constants be generated and selected efficiently?

In order to address the first question, let us assume that a set of *constant nodes* $\{c\}$, which are known to be useful with respect to relation r, has been already identified. The relationship between each constant c and target node t may be represented in terms of path probability features, $P(c \rightarrow t; \pi)$. For example, the rule $IsA(t, SportsTeam)$ corresponds to a path originating at constant *SportsTeam*, and reaching target node t over a direct edge typed IsA^{-1}. Such paths, which are independent of the source node s, readily represent the semantic type, or other

characteristic attributes of relevant target nodes. Similarly, a feature (c, ϕ), designating a constant and an empty path, forms a prior for the target node identity.

The remaining question is how to identify meaningful constant features. Apriori, candidate constants range over all of the graph nodes, and searching for useful paths that originate at arbitrary constants is generally intractable. Provided with labeled examples, we apply the path-finding procedure for this purpose, where rather than search for high-probability paths from source node s to target t, paths are explored in a backward fashion, initiating path search at the known relevant target nodes $t \in G_i$ per each labeled query. This process identifies candidate (c, π) tuples, which give high $P(c \leftarrow t; \pi^{-1})$ values, at bounded computation cost. As a second step, $P(c \rightarrow t; \pi)$ feature values are calculated, where useful path features are selected using the *precision* and *coverage* criteria. Further details are discussed in Section 4.4.

4.3 Bi-directional Random Walk Features

The PRA algorithm only uses features of the form $P(s \rightarrow t; \pi)$. In this study we also consider graph walk features in the inverse direction of the form $P(s \leftarrow t; \pi^{-1})$. Similarly, we consider both $P(c \rightarrow t; \pi)$ and $P(c \leftarrow t; \pi^{-1})$. While these path feature pairs represent the same logical expressions, the directional random walk probabilities may greatly differ. For example, it may be highly likely for a random walker to reach a target node representing a sports team t from node s denoting a player over a path π that describes the functional *AthletePlaysForTeam* relation, but unlikely to reach a particular player node s from the multiplayer team t via the reversed path π^{-1}.

In general, there are six types of random walk probabilities that may be modeled as relational features following the introduction of constant paths and inverse path probabilities. The random walk probabilities between s and constant nodes c, $P(s \rightarrow c; \pi)$ and $P(s \leftarrow c; \pi)$, do not directly affect the ranking of candidate target nodes, so we do not use them in this study. It is possible, however, to generate random walk features that combine these probabilities with random walks starting or ending with t through conjunction.

Algorithm 1 Cor-PRA Feature Induction[1]

Input training queries $\{(s_i, G_i)\}, i = 1...n$
for each query (s, G) **do**
1. Path exploration
(i). Apply *path-finding* to generate paths $\mathcal{P}_s$ up to length ℓ that originate at s_i.
(ii). Apply *path-finding* to generate paths $\mathcal{P}_t$ up to length ℓ that originate at every $t_i \in G_i$.
2. Calculate random walk probabilities:
for each $\pi_s \in \mathcal{P}_s$: **do**
compute $P(s \rightarrow x; \pi_s)$ and $P(s \leftarrow x; \pi_s^{-1})$
end for
for each $\pi_t \in \mathcal{P}_t$: **do**
compute $P(G \rightarrow x; \pi_t)$ and $P(G \leftarrow x; \pi_t^{-1})$
end for
3. Generate constant paths candidates:
for each $(x \in N, \pi \in \mathcal{P}_t)$ with $P(G \rightarrow x|\pi_t) > 0$ **do**
propose path feature $P(c \leftarrow t; \pi_t^{-1})$ setting $c = x$, and update its statistics by $coverage \mathrel{+}= 1$.
end for
for each $(x \in N, \pi \in \mathcal{P}_t)$ with $P(G \leftarrow x|\pi_t^{-1}) > 0$ **do**
propose $P(c \rightarrow t; \pi_t)$ setting $c = x$ and update its statistics by $coverage \mathrel{+}= 1$
end for
4. Generate long (concatenated) path candidates:
for each $(x \in N, \pi_s \in \mathcal{P}_s, \pi_t \in \mathcal{P}_t)$ with $P(s \rightarrow x|\pi_s) > 0$ and $P(G \leftarrow x|\pi_t^{-1}) > 0$ **do**
propose long path $P(s \rightarrow t; \pi_s.\pi_t^{-1})$ and update its statistics by $coverage \mathrel{+}= 1$, and $precision \mathrel{+}= P(s \rightarrow x|\pi_s)P(G \leftarrow x|\pi_t^{-1})/n$.
end for
for each $(x \in N, \pi_s \in \mathcal{P}_s, \pi_t \in \mathcal{P}_t)$ with $P(s \leftarrow x|\pi_s^{-1}) > 0$ and $P(G \rightarrow x|\pi_t) > 0$ **do**
propose long path $P(s \leftarrow t; \pi_t.\pi_s^{-1})$ and update its statistics by $coverage \mathrel{+}= 1$, and $precision \mathrel{+}= P(s \leftarrow x|\pi_s^{-1})P(G \rightarrow x|\pi_t)/n$.
end for
end for

4.4 Cor-PRA feature induction and selection

The proposed feature induction procedure is outlined in Alg. 1. Given labeled node pairs, the particle-filtering path-finding procedure is first applied to identify edge type sequences up to length ℓ that originate at either source nodes s_i or relevant target nodes t_i (step 1). Bi-directional path probabilities are then calculated over these paths, recording the terminal graph nodes x (step 2). Note that since the set of nodes x may be large, path probabilities are all computed with respect to s or t as starting points. As a result of the induction process, candidate relational paths involving constants are identified, and are associated with precision and coverage statistics (step 3). Further, long paths up to length 2ℓ are formed between the source and target nodes as the combination of paths π_s from the source side and path π_t from the target side, updating accuracy and coverage statistics for the concatenated paths $\pi_s\pi_t$

(step 4).

Following feature induction, feature selection is applied. First, random walks are performed for all the training queries, so as to obtain complete (rather than sampled) precision and coverage statistics per path. Then relational paths, which pass respective tuned thresholds are added to the model. We found, however, that applying this strategy for paths with constants often leads to over-fitting. We therefore select only the top K constant features in terms of F_1[2], where K is tuned using training examples.

Finally, at test time, random walk probabilities are calculated for the selected paths, starting from either s or c nodes per query–since the identity of relevant targets t is unknown, but rather has to be revealed.

5 Experiments

In this section, we report the results of applying Cor-PRA to the tasks of knowledge base inference and person named entity extraction from parsed text.

We performed 3-fold cross validation experiments, given datasets of labeled queries. For each query node in the evaluation set, a list of graph nodes ranked by their estimated relevancy to the query node s and relation r is generated. Ideally, relevant nodes should be ranked at the top of these lists. Since the number of correct answers is large for some queries, we report results in terms of mean average precision (MAP), a measure that reflects both precision and recall (Turpin and Scholer, 2006).

The *coverage* and *precision* thresholds of Cor-PRA were set to $h = 2$ and $a = 0.001$ in all of the experiments, following empirical tuning using a small subset of the training data. The particle filtering path-finding algorithm was applied using the parameter setting $w_g = 10^6$, so as to find useful paths with high probability and yet constrain the computational cost.

Our results are compared against the FOIL algorithm[3], which learns first-order horn clauses. In order to evaluate FOIL using MAP, its candidate beliefs are first ranked by the number of FOIL rules they match. We further report results using *Random Walks with Restart* (RWR), also known as personalized PageRank (Haveliwala, 2002), a popular random walk based graph similarity measure, that has been shown to be fairly successful for many types of tasks (e.g., (Agirre and Soroa, 2009; Moro et al., 2014)). Finally, we compare against PRA, which models relational paths in the form of edge-sequences (no constants), using only uni-directional path probabilities, $P(s \rightarrow t; \pi)$.

[2] F_1 is the harmonic mean of precision and recall, where the latter is defined as $\frac{coverage}{total\_number\_targets\_in\_training\_queries}$

[3] http://www.rulequest.com/Personal/

Table 1: MAP and training time [sec] on KB inference and NE extraction tasks. $const_i$ denotes constant paths up to length i.

	KB inference		NE extraction	
	Time	MAP	Time	MAP
RWR	25.6	0.429	7,375	0.017
FOIL	18918.1	0.358	366,558	0.167
PRA	10.2	0.477	277	0.107
CoR-PRA-$no\text{-}const$	16.7	0.479	449	0.167
CoR-PRA-$const_2$	23.3	0.524	556	0.186
CoR-PRA-$const_3$	27.1	**0.530**	643	**0.316**

All experiments were run on a machine with a 16 core Intel Xeon 2.33GHz CPU and 24Gb of memory. All methods are trained and tested with the same data splits. We report the total training time of each method, measuring the efficiency of inference and induction as a whole.

5.1 Knowledge Base Inference

We first consider relational inference in the context of NELL, a semantic knowledge base constructed by continually extracting facts from the Web (Carlson et al., 2010b). This work uses a snapshot of the NELL knowledge base graph, which consists of ∼1.6M edges comprised of 353 edge types, and ∼750K nodes. Following Lao et al. (2011), we test our approach on 16 link prediction tasks, targeting relations such as *Athlete-plays-in-league*, *Team-plays-in-league* and *Competes-with*.

Table 1 reports MAP results and training times for all of the evaluated methods. The maximum path length of RWR, PRA, and CoR-PRA are set to 3 since longer path lengths do not result in better MAPs. As shown, RWR performance is inferior to PRA; unlike the other approaches, RWR is merely associative and does not involve path learning. PRA is significantly faster than FOIL due to its particle filtering approach in feature induction and inference. It also results in a better MAP performance due to its ability to combine random walk features in a discriminative model.

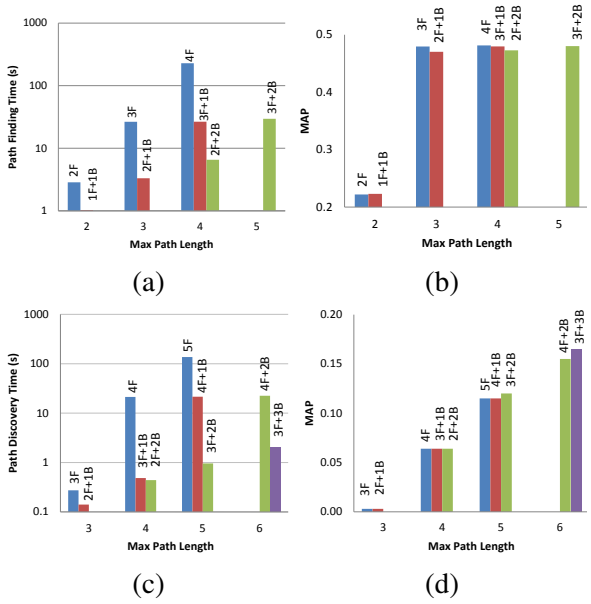

Figure 2: Path finding time (a) and MAP (b) for the KB inference (top) and name extraction (bottom) tasks. A marker $iF + jB$ indicates the maximum path exploration depth i from query node s and j from target node t–so that the combined path length is up to $i + j$. No paths with constants were used.

Table 1 further displays the evaluation results of several variants of CoR-PRA. As shown, modeling features that encode random walk probabilities in both directions (CoR-PRA-*no-const*), yet no paths with constants, requires longer training times, but results in slightly better performance compared with PRA. Note that for a fixed path length, CoR-PRA has "forward" features of the form $P(s \rightarrow t; \pi)$, the probability of reaching target node t from source node s over path π (similarly to PRA), as well as backward features of the form $P(s \leftarrow t; \pi^{-1})$, the probability of reaching s from t over the backward path π^{-1}. As mentioned earlier these probabilities are not the same; for example, a player usually plays for one team, whereas a team is linked to many players.

Performance improves significantly, however, when paths with constants are further added. The table includes our results using constant paths up to length $\ell = 2$ and $\ell = 3$ (denoted as CoR-PRA-$const_\ell$). Based on tuning experiments on one fold of the data, $K = 20$ top-rated constant paths were included in the models.[4] We found that these paths provide informative class priors; example paths and their interpretation are included in Table 2.

[4]MAP performance peaked at roughly $K = 20$, and gradually decayed as K increased.

Table 2: Example paths with constants learnt for the knowledge base inference tasks. (ϕ denotes empty paths.)

Constant path	Interpretation
r=athletePlaysInLeague	
$P(mlb \rightarrow t; \phi)$	Bias toward *MLB*.
$P(boston\_braves \rightarrow t;$ $\langle athletePlaysForTeam^{-1},$ $athletePlaysInLeague\rangle)$	The leagues played by *Boston Braves* university team members.
r=competesWith	
$P(google \rightarrow t; \phi)$	Bias toward *Google*.
$P(google \rightarrow t;$ $\langle competesWith, competesWith\rangle)$	Companies which compete with Google's competitors.
r=teamPlaysInLeague	
$P(ncaa \rightarrow t; \phi)$	Bias toward *NCAA*.
$P(boise\_state \rightarrow t;$ $\langle teamPlaysInLeague\rangle)$	The leagues played by *Boise State* university teams.

Figure 2(a) shows the effect of increasing the maximal path length on path finding and selection time. The leftmost (blue) bars show baseline performance of PRA, where only forward random walks are applied. It is clearly demonstrated that the time spent on path finding grows exponentially with ℓ. Due to memory limitations, we were able to execute forward-walk models only up to 4 steps. The bars denoted by $iF + jB$ show the results of combining forward walks up to length i with backward walks of up to $j = 1$ or $j = 2$ steps. Time complexity using bidirectional random walks is dominated by the longest path segment (either forward or backward)—e.g., the settings $3F$, $3F + 1B$, $3F + 2B$ have similar time complexity. Using bidirectional search, we were able to consider relational paths up to length 5. Figure 2(b) presents MAP performance, where it is shown that extending the maximal explored path length did not improve performance in this case. This result indicates that meaningful paths in this domain are mostly short. Accordingly, path length was set to 3 in the respective main experiments.

5.2 Named Entity Extraction

We further consider the task of named entity extraction from a corpus of parsed texts, following previous work by Minkov and Cohen (2008).

In this case, an entity-relation graph schema is used to represent a corpus of parsed sentences, as illustrated in Figure 3. Graph nodes denoting word mentions (in round edged boxes) are linked over edges typed with dependency relations. The

parsed sentence structures are connected via nodes that denote word lemmas, where every word lemma is linked to all of its mentions in the corpus via the special edge type W. We represent part-of-speech tags as another set of graph nodes, where word mentions are connected to the relevant tag over POS edge type.

In this graph, task-specific word similarity measures can be derived based on the lexico-syntactic paths that connect word types (Minkov and Cohen, 2014). The task defined in the experiments is to retrieve a ranked list of *person names* given a small set of seeds. This task is implemented in the graph as a query, where we let the query distribution be uniform over the given seeds (and zero elsewhere). That is, our goal is to find target nodes that are related to the query nodes over the relation r =*similar-to*, or, *coordinate-term*. We apply link prediction in this case with the expected result of generating a ranked list of graph nodes, which is populated with many additional person names. The named entity extraction task we consider is somewhat similar to the one adopted by FIGER (Ling and Weld, 2012), in that a finer-grain category is being assigned to proposed named entities. Our approach follows however set expansion settings (Wang and Cohen, 2007), where the goal is to find new instances of the specified type from parsed text.

In the experiments, we use the training set portion of the MUC-6 data set (MUC, 1995), represented as a graph of 153k nodes and 748K edges. We generated 30 labeled queries, each comprised of 4 person names selected randomly from the person names mentioned in the data set. The MUC corpus is fully annotated with entity names, so that relevant target nodes (other person names) were readily sampled. Extraction performance was evaluated considering the tagged *person names*, which were not included in the query, as the correct answer set. The maximum path length of RWR, PRA, and CoR-PRA are set to 6 due to memory limitation.

Table 1 shows that PRA is much faster than RWR or FOIL on this data set, giving competitive MAP performance to FOIL. RWR is generally ineffective on this task, because similarity in this domain is represented by a relatively small set of long paths, whereas RWR express local node associations in the

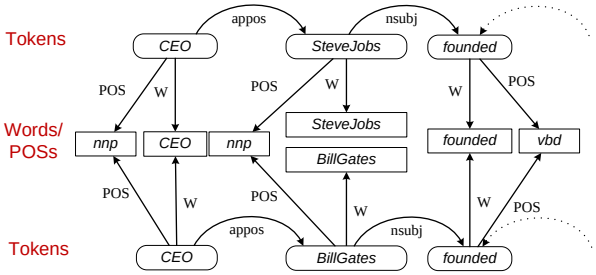

Figure 3: Part of a typed graph representing a corpus of parsed sentences.

Table 3: Highly weighted paths with constants learnt for the person name extraction task.

Constant path	Interpretation
$P(said \leftarrow t; W^{-1}, nsubj, W)$	The subjects of 'said' or 'say'
$P(says \leftarrow t; W^{-1}, nsubj, W)$	are likely to be a person name.
$P(vbg \leftarrow t; POS^{-1}, nsubj, W)$	Subjects, proper nouns, and
$P(nnp \leftarrow t; POS^{-1}, W)$	nouns with apposition or
$P(nn \leftarrow t; POS^{-1}, appos^{-1}, W)$	possessive constructions, are
$P(nn \leftarrow t; POS^{-1}, poss, W)$	likely to be person names.

graph (Minkov and Cohen, 2008). Modeling inverse path probabilities improves performance substantially, and adding relational features with constants boosts performance further. The constant paths learned encode lexical features, as well as provide useful priors, mainly over different part-of-speech tags. Example constant paths that were highly weighted in the learned models and their interpretation are given in Table 3.

Figure 2(c) shows the effect of modeling long relational paths using bidirectional random walks in the language domain. Here, forward path finding was applied to paths up to length 5 due to memory limitation. The figure displays the results of exploring paths up to a total length of 6 edges, performing backward search from the target nodes of up to $j = 1, 2, 3$ steps. MAP performance (Figure 2(d)) using paths of varying lengths shows significant improvements as the path length increases. Top weighted long features include:

$P(s \rightarrow t; W^{-1}, conj\_and^{-1}, W, W^{-1}, conj\_and, W)$

$P(s \rightarrow t; W^{-1}, nn, W, W^{-1}, appos^{-1}, W)$

$P(s \rightarrow t; W^{-1}, appos, W, W^{-1}, appos^{-1}, W)$

These paths are similar to the top ranked paths found in previous work (Minkov and Cohen, 2008). In comparison, their results on this dataset using paths of up to 6 steps measured 0.09 in MAP. Our results reach roughly 0.16 in MAP due to modeling of inverse paths; and, when constant

paths are incorporated, MAP reaches 0.32.

Interestingly, in this domain, FOIL generates fewer yet more complex rules, which are characterised with low recall and high precision, such as: $W(B, A) \wedge POS(B, nnp) \wedge nsubj(D, B) \wedge W(D, said) \wedge appos(B, F) \rightarrow person(A)$. Note that subsets of these rules, namely, $POS(B, nnp)$, $nsubj(D, B) \wedge W(D, said)$ and $appos(B, F)$ have been discovered by PRA as individual features assigned with high weights (Table 3). This indicates an interesting future work, where products of random walk features can be used to express their conjunctions.

6 Conclusion

We have introduced CoR-PRA, extending an existing random walk based relational learning paradigm to consider relational paths with constants, bi-directional path features, as well as long paths. Our experiments on knowledge base inference and person name extraction tasks show significant improvements over previously published results, while maintaining efficiency. An interesting future direction is to use products of these random walk features to express their conjunctions.

Acknowledgments

We thank the reviewers for their helpful feedback. This work was supported in part by BSF grant No. 2010090 and a grant from Google Research.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*.

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

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. Hruschka Jr., and T. Mitchell. 2010a. Toward an architecture for never-ending language learning. In *AAAI*.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010b. Toward an Architecture for Never-Ending Language Learning. In *AAAI*.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610.

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *EMNLP*.

Matt Gardner, Partha Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating Vector Space Similarity in Random Walk Inference over Knowledge Bases. In *EMNLP*.

Taher H. Haveliwala. 2002. Topic-sensitive pagerank. In *WWW*, pages 517–526.

Ondřej Kuželka and Filip Železný. 2008. A restarted strategy for efficient subsumption testing. *Fundam. Inf.*, 89(1):95–109, January.

Ondřej Kuželka and Filip Železný. 2009. Block-wise construction of acyclic relational features with monotone irreducibility and relevancy properties. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 569–576, New York, NY, USA. ACM.

Ni Lao and William W. Cohen. 2010a. Fast query execution for retrieval models based on path-constrained random walks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 881–888, New York, NY, USA. ACM.

Ni Lao and William W. Cohen. 2010b. Relational retrieval using a combination of path-constrained random walks. In *Machine Learning*, volume 81, pages 53–67, July.

Ni Lao, Tom M. Mitchell, and William W. Cohen. 2011. Random Walk Inference and Learning in A Large Scale Knowledge Base. In *EMNLP*, pages 529–539.

Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W. Cohen. 2012. Reading the web with learned syntactic-semantic inference rules. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1017–1026, Stroudsburg, PA, USA. Association for Computational Linguistics.

X. Ling and D.S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*.

Einat Minkov and William W Cohen. 2008. Learning Graph Walk Based Similarity Measures for Parsed Text. *EMNLP*.

Einat Minkov and William W. Cohen. 2014. Adaptive graph walk-based similarity measures for parsed text. *Natural Language Engineering*, 20(3).

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2.

1995. *MUC6 '95: Proceedings of the 6th Conference on Message Understanding*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michael Pazzani, Cliff Brunk, and Glenn Silverstein. 1991. A Knowledge-Intensive Approach to Learning Relational Concepts. In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 432–436. Morgan Kaufmann.

J. Ross Quinlan and R. Mike Cameron-Jones. 1993. FOIL: A Midterm Report. In *ECML*, pages 3–20.

B L Richards and R J Mooney. 1991. First-Order Theory Revision. In *Proceedings of the 8th International Workshop on Machine Learning*, pages 447–451. Morgan Kaufmann.

Michele Sebag and Celine Rouveirol. 1997. Tractable induction and classification in first order logic via stochastic matching. In *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence - Volume 2*, IJCAI'97, pages 888–893, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ashwin Srinivasan. 2001. The Aleph Manual. In *http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/*.

F. Suchanek, G. Kasneci, and G. Weikum. 2007. YAGO - A Core of Semantic Knowledge. In *WWW*.

Andrew Turpin and Falk Scholer. 2006. User performance versus precision measures for simple search tasks. In *PProceedings of the international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*.

Filip Železný and Nada Lavrač. 2006. Propositionalization-based relational subgroup discovery with rsd. *Mach. Learn.*, 62(1-2):33–63, February.

Richard C Wang and William W Cohen. 2007. Language-independent set expansion of named entities using the web. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*.

William Yang Wang, Kathryn Mazaitis, and William W Cohen. 2013. Programming with personalized pagerank: A locally groundable first-order probabilistic logic. *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM 2013)*.