Scalable Inference over Large Knowledge Bases

**Path Ranking Algorithm (PRA)**

- **Model** (Lao & Cohen, ECML 2010)
  - Scores source-target pairs by a linear function of their path features
  
  \[
  \text{score}(s, t) = \sum_{P \in \text{P}} \text{Prob}(s \rightarrow t | P) \theta_P
  \]

  \(P\) is the set of all relation paths with length \(\leq l\)

- **Training**
  - For a relation \(R\) and a set of node pairs \([(s, t), \ldots]\), create dataset \(D\)
  - e.g., \(s \rightarrow \text{Charlotte}, t \rightarrow \text{painter/writer}\)
  
  \(\chi_s\) is a vector of all the path features for \((s, t)\), and \(\gamma_t\) indicates whether \(R(s, t)\) is true or not
  
  \(\theta\) is estimated using L1,L2-regularized logistic regression

- **Path Finding & Feature Selection** (L.M.C., EMNLP 2011)
  - Impractical to enumerate all possible edge sequences \(O(|V|^l)\)
    - Constraint 1: paths to instantiate in at least \(K(=5)\) training queries
    - Constraint 2: \(\text{Prob}(s \rightarrow t| \text{path}, s \rightarrow \text{any node}) > \alpha (=0.2)\)

  Depth first search up to length \(l\) starts from a set of training queries, expand a node if the instantiation constraint is satisfied

- **Efficient Inference** (Lao & Cohen, KDD 2010)
  - Exact calculation of random walk distributions results in non-zero probabilities for many internal nodes in the graph
  - A few random walkers (or particles) are enough to distinguish good target nodes from bad ones

- **Random Walk Inference**
  - Features generated by Path Constrained Random Walks
  - \(\text{Prob}(\text{Charlotte} \rightarrow \text{Writer} | \text{HasFather})\)
  - \(\text{Prob}(\text{Charlotte} \rightarrow \text{Writer} | \text{Write}, \text{isa}, \text{isa}^-1, \text{Write}, \text{isa})\)
  - \(\text{Prob}(\text{Charlotte} \rightarrow \text{Writer} | \text{InSentence})\)

- **Evaluation by Mechanical Turk**
  - Evaluate the top ranked result for each query. Sort the predictions for each predicate by scores, and evaluate precisions at top 10, 100 and 1000 queries

<table>
<thead>
<tr>
<th>Functional Predicates</th>
<th>Non-functional Predicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Rules</td>
<td>p@10 p@100 p@1000 #Rules</td>
</tr>
<tr>
<td>N-FOIL</td>
<td>2.1(+37) 0.76 0.380 0.007</td>
</tr>
<tr>
<td>PRA</td>
<td>43 0.79 0.668 0.615 0.692 0.650 0.620 0.615</td>
</tr>
</tbody>
</table>

- **Future Work**
  - Efficiently discover long paths
  - Discover lexicalized paths (contains constant nodes)
  - Generalize relation paths to trees/networks

---

**The NELL Case Study**

- **Never-Ending Language Learning**
  - Combines multiple strategies: morphological patterns, textual context, html patterns, logical inference, etc.

- **Our task**
  - Learn to infer new instances of 96 relations for which NELL database has more than 100 instances. Closed world assumption for training: the actual nodes known to satisfy \(R(x, y)\) are treated as labeled positive examples, and all other nodes are treated as negative examples

Example features:

- \(\text{athletePlaysSport}\)
- \(\text{hasCity}\)
- \(\text{hasPlayer}\)
- \(\text{memberOfTeam}\)
- \(\text{painter/writer}\)

---

**Relational Learning**

Prediction with rich meta-data has great potentials and challenges

The world consists of objects and relations among them (e.g. family/friends, behavior, literature), but statistical learning tools (e.g. SVM) expect samples and their feature values

- **Random walk with restart**
  - One-parameter-per-edge label is limited because the context of an edge label is ignored, e.g. \(\text{Prob}(\text{Charlotte} \rightarrow \text{Writer})\)

- **First Order Inductive Learner**
  - Learn Horn clauses in first order logic (FOIL, 1993), e.g.
    - \(\text{HasFather}(a, b) \land \text{isa}(b, y) \Rightarrow \text{isa}(a; y)\)
    - \(\text{InSentence}(a, j) \land \text{isa}(b, y) \Rightarrow \text{isa}(a; y)\)

  Lexicalized rules: \(\text{HasFather}(x, a) \land \text{isa}(a, \text{writer}) \Rightarrow \text{isa}(x; \text{writer})\)

  Quantifier: \(\exists i, \text{Write}(x, i) \Rightarrow \text{isa}(x; \text{writer})\)

  Horn clauses are costly to discover, inference is generally slow

  Combine rules with disjunctions, therefore cannot leverage low accuracy rules

- **Random Walk Inference**
  - Features generated by Path Constrained Random Walks
  - \(\text{Prob}(\text{Charlotte} \rightarrow \text{Writer} | \text{HasFather}, \text{isa})\)
  - \(\text{Prob}(\text{Charlotte} \rightarrow \text{Writer} | \text{Write}, \text{isa}, \text{isa}^-1, \text{Write}, \text{isa})\)
  - \(\text{Prob}(\text{Charlotte} \rightarrow \text{Writer} | \text{InSentence}, \text{InSentence}^-1, \text{isa})\)