Scalable Knowledge Harvesting with High Precision and High Recall

Ndapa Nakashole
Martin Theobald
Gerhard Weikum
Web Knowledge Harvesting

- **Goal:** To organise text into precise facts

  - To be a backbone for advanced applications
    - Semantic search, QA, entity disambiguation, …

  - Lots of ongoing efforts towards this goal
    - Cyc, YAGO-NAGA, TextRunner, ReadTheWeb, …

---

Alex Rodriguez


<table>
<thead>
<tr>
<th>A.Rodriguez</th>
<th>PlaysInLeague</th>
<th>Major League Baseball</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.Rodriguez</td>
<td>PlaysForTeam</td>
<td>New York Yankees</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Approaches

- Prevalent approaches roughly fall into two categories

**Solely pattern-based approaches**

Seed facts:
playsFor (Michael_Jordan, Chicago_Bulls)

Patterns in texts:
X scored for Y
X in the match against Y

New facts:
playsFor(Kobe_Bryant, LA_Lakers)
playsFor(Tom Brady, NE_Patriots)

**Patterns + constraint-aware approaches**

Constraint rule:
playsForNationalTeam(X, T1) \(\land\) playsForNationalTeam(X, T2) \(\Rightarrow\) (T1 == T2)

Inference rule:
teamMates(X, Y) \(\land\) playsFor(X, T) \(\Rightarrow\) playsFor(Y, T)
Use canonical entities in facts

LA Lakers
Los Angeles Lakers
Lakers

All refer to the same entity
Los_Angeles_Lakers
Goal

• To reconcile precision, recall and scalability
  – With facts referring to canonical entities

• PROSPERA contributes towards this goal:
  – Robust patterns
  – Streamlined constraint-aware reasoning
  – Deployed on a distributed architecture
Outline

• Introduction

• PROSPERA Components
  – Generalised patterns
  – Clause weighting in Max-SAT reasoning
  – Distributed architecture

• Large-scale Experiments

• Conclusions
(Beckham, LA_Galaxy)
(Rodriguez, NY_Yankees)
!(Brady, LA_Galaxy)

Seed examples and counter examples

Text corpus

Pattern Gathering

X plays for Y
X hit his first home run for Y
...

Pattern Analysis

X hit PRP ADJ home run for Y
...

Reasoning (MAX-SAT)

PlaysFor(Brady, NE_Patriots)
!PlaysFor(Beckham, Bulls)
...

PROSPERA 3-phase Architecture
Consider a sentence in the corpus:

Alex Rodriguez *hit his 600th career home run helping the New York Yankees* …

**Issue: pattern is overly specific**
- Pattern: *hit his 600th career home run, helping the* …
- Not a match for: *hit a game-changing home run, helping the* …

**Possible solutions**
- Ellipses, regular expressions, or dependency parsing
Generalized Patterns

• Our approach: feed all patterns into a frequent n-gram itemset mining algorithm
  – Identifies frequently co-occurring sub-patterns
  – N-gram pattern: {hit his, home run helping the}
  – A match: {hit a game-changing, home run helping the}

• To counter variations in patterns
  – Replacing certain words with part-of-speech tags
  – E.g. adjectives, nominative pronouns, etc

  { hit PRP, home run helping the}
  { scored the ADJ goal for}
Noisy patterns

• Possible misleading patterns
  – X is a fan of Y
  – X was drafted by Y

• Make use of counter-examples
  – Noisy patterns appear with various seed pairs stand in widely different relations
  – If the pattern appears with many counter examples, then it’s a poor quality pattern
Pattern quality measures

• Pattern occurrences with seed and counter-examples used to compute quality measures

• **Support:** pattern frequency in text

• **Confidence:** how often pattern was found with seed instances relative to counter

• Each seed pattern has a confidence weight
  – **Weight** = support x confidence
Aggregated Candidates

Matching newly found patterns to seed patterns produces a set of weighted fact candidates

- PlaysFor(Brady, New_England_Patriots)?[0.93]
- PlaysFor(LeBron,Chicago_Bulls)?[0.23]
- PlaysFor(LeBron,Miami_Heat)?[0.78]

Candidates are grouped with an aggregated weight

- By fact candidates: Weight(R,x,y) = \sum w(R,x,y)
- By new patterns: Weight(p,R) = \sum w(p,R)
Max-SAT Logical Reasoning

• Serves to prune false candidates
• We built on SOFIE (Suchanek et al., WWW2009)
  – Rule-based reasoning model

\[
\begin{align*}
\text{occurs}(p,x,y) \land R(x,y) &\Rightarrow \text{expresses}(p,R) \\
\text{occurs}(p,x,y) \land \text{expresses}(p,R) &\Rightarrow R(X,R)
\end{align*}
\]

// pattern-fact duality

\[
\begin{align*}
R(x,y) \land \text{diff}(y,z) &\Rightarrow \neg R(x,z)
\end{align*}
\]

// functional constraint

– Solves a max. satisfiability problem on clauses of fact candidates instantiated on the rules (weighted max-sat)

• Our improvements:
  – Pattern filtering reduces amount of input to reasoner improving efficiency
  – Candidate weights guide the reasoner to most likely answer
Distributed Architecture

• Developed MapReduce algorithms for all three phases

• Distributed pattern gathering
  – Trivially parallelizable
  – Each document processed independently

• Distributed pattern analysis
  – Expressed as a sequence of MapReduce jobs
  – N-gram-itemset pattern generation
  – Pattern weight computation
  – Fact candidate generation
Distributed Reasoning

• Distributed reasoning is non-trivial!
  – Constraints impose dependencies
  – Statements of grounded rules are vertices
  – An edge for each literal in common

• Min-cut two-phase algorithm
  – Randomized approximation (Karypis et al.’98, Karger’96)
  1. Coarsen graph
  2. Partition coarser graph

occurs (drafted by, Alex, Yankees) 
∧ playsFor(Alex, Yankees) 
⇒ expresses (drafted by, playsFor)
Experiments

• ClueWeb09 corpus ~500 mio. Pages

• Comparisons to recent NELL
  – Reported in Carlson et al., AAAI 2010)
  – A cluster-based implementation

• Two domains
  – Sports for comparison to NELL
  – Academia for reasoning over domain-specific constraints
### Precision and Recall (Sports relations)

<table>
<thead>
<tr>
<th>Relation</th>
<th># Extractions</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PROSPERA</td>
<td>NELL</td>
</tr>
<tr>
<td>PlaysforTeam</td>
<td>14,655</td>
<td>456</td>
</tr>
<tr>
<td>CoachesTeam</td>
<td>1,013</td>
<td>329</td>
</tr>
<tr>
<td>PlaysInLeague</td>
<td>1,920</td>
<td>288</td>
</tr>
<tr>
<td>TeamMate</td>
<td>19,666</td>
<td>n/a</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Prospera sports experiments completed in 2.5 days
- NELL results obtained over 66 days
Accumulated facts

- Number of extractions increases over iterations
  - Implied more iterations would lead to further extractions
Facts refer to disambiguated entities
- \texttt{AtheletePlaysForTeam(Ben\_Gordon, Chicago\_Bulls)}
- \texttt{TeamPlaysInLeague(Chicago\_Bulls, NBA)}

As opposed to
- \texttt{AtheletePlaysForTeam(Ben Gordon, Bulls)}
- \texttt{TeamPlaysInLeague(Chicago Bulls, NBA)}
Reasoning Parallelization Speedup

• Sports domain
  – No domain specific constraints
  – First iteration speedup of 2.2 (7.6 min vs. 3.5 min)

• Academic domain
  – Reasoner uses domain specific constraints
  – E.g Advisor’s life span should overlap with student's life span
  – First iteration speedup of 2.76 (7.1 hours vs. 2.5 hours)
Conclusions

• Contributed towards reconciling precision, recall and scalability in knowledge harvesting
  – Generalized patterns (Recall)
  – Streamlined reasoning (Precision)
  – Distributed implementation (Scalability)

– Experimental data available at:
  www.mpi-inf.mpg.de/yago-naga/prospera/
Thanks …

Questions

?