Joint Inference in Information Extraction

Hoifung Poon
University of Washington

(Joint work with Pedro Domingos)
Problems of Pipeline Inference

- AI systems typically use pipeline architecture
  - Inference is carried out in stages
  - E.g., information extraction, natural language processing, speech recognition, vision, robotics
- Easy to assemble & low computational cost, but …
  - Errors accumulate along the pipeline
  - No feedback from later stages to earlier ones
- Worse: Often process one object at a time
We Need Joint Inference


We Need Joint Inference

- A topic of keen interest
  - Statistical relational learning [Getoor & Taskar, 2007]
  - Natural language processing [Sutton et. al., 2006]
  - Etc.

- However …
  - Often very complex to set up
  - Computational cost can be prohibitive
  - Joint inference may hurt accuracy

- State of the art:
  - Fully joint approaches are still rare
  - Even partly-joint ones require much engineering
Joint Inference in Information Extraction: State of the Art

- **Information Extraction (IE):**
  Segmentation + Entity Resolution (ER)

- **Previous approaches are not fully joint:**
  - [Pasula et. al., 2003]: Citation matching
    Perform segmentation in pre-processing
  - [Wellner et. al., 2004]: Citation matching
    Only one-time feedback from ER to segmentation

- Both Pasula and Wellner require much engineering
Joint Inference in Information Extraction: Our Approach

- **We develop the first fully joint approach to IE**
  - Based on **Markov logic**
  - Performs segmentation and entity resolution in a single inference process
  - Applied to citation matching domains
    - Outperform previous approaches
    - Joint inference improves accuracy
- **Requires far less engineering effort**
Outline

● **Background: Markov logic**
● MLNs for joint citation matching
● Experiments
● Conclusion
Markov Logic

- A general language capturing logic and uncertainty
- A Markov Logic Network (MLN) is a set of pairs \((F, w)\) where
  - \(F\) is a formula in first-order logic
  - \(w\) is a real number
- Together with constants, it defines a Markov network with
  - One node for each ground predicate
  - One feature for each ground formula \(F\), with the corresponding weight \(w\)

\[
P(x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i(x)\right)
\]
Markov Logic

- Open-source package: Alchemy
  alchemy.cs.washington.edu

- Inference: MC-SAT [Poon & Domingos, 2006]

- Weight Learning: Voted perceptron
  [Singla & Domingos, 2005]
Outline

- Background: Markov logic
- **MLNs for joint citation matching**
- Experiments
- Conclusion
Citation Matching

- Extract bibliographic records from citation lists
- Merge co-referent records
- Standard application of information extraction
- Here: Extract authors, titles, venues
Types and Predicates

token = {Minton, Hoifung, and, Pedro, ...}
field = {Author, Title, Venue}
citation = {C1, C2, ...}
position = {0, 1, 2, ...}

Token(token, position, citation)
HasPunc(citation, position)
......

InField(position, field, citation)
SameCitation(citation, citation)
MLNs for Citation Matching

- Isolated segmentation
- Entity resolution
- Joint segmentation: Jnt-Seg
- Joint segmentation: Jnt-Seg-ER
Isolated Segmentation

- Essentially a hidden Markov model (HMM)
  \[ \text{Token}(+t, p, c) \Rightarrow \text{InField}(p, +f, c) \]
  \[ \text{InField}(p, +f, c) \Rightarrow \text{InField}(p+1, +f, c) \]

- Add boundary detection:
  \[ \text{InField}(p, +f, c) \land \neg \text{HasPunc}(c, p) \]
  \[ \Rightarrow \text{InField}(p+1, +f, c) \]

- Other rules: Start with author, comma, etc.
Entity Resolution

- Obvious starting point: [Singla & Domingos, 2006]
  - MLN for citation entity resolution
  - Assume pre-segmented citations
- However ...
  - Poor results from merging this and segmentation
  - Entity resolution misleads segmentation
    \[ \text{InField}(p, f, c) \land \text{Token}(t, p, c) \]
    \[ \land \text{InField}(p', f, c') \land \text{Token}(t, p', c') \]
    \[ \Rightarrow \text{SameCitation}(c, c') \]
- Example of how joint inference can hurt
Entity Resolution

- Devise specific predicate and rule to pass information between stages
  
  - `SimilarTitle(citation, position, position, citation, position, position)`
  - Same beginning trigram and ending unigram
  - No punctuation inside
  - Meet certain boundary conditions

- `SimilarTitle(c, p1, p2, c', p1', p2') \land SimilarVenue(c, c') \Rightarrow SameCitation(c, c')`

- Suffices to outperform the state of the art
Joint Segmentation: Jnt-Seg

- Leverage well-delimited citations to help more ambiguous ones
- \texttt{JointInferenceCandidate(citation, position, citation)}


Joint Segmentation: Jnt-Seg

- Augment the transition rule:

  \[ \text{InField}(p, +f, c) \land \neg\text{HasPunc}(c, p) \]
  \[ \land \neg(\exists c' \ \text{JointInferenceCandidate}(c, p, c')) \]
  \[ \Rightarrow \text{InField}(p+1, +f, c) \]

- Introduces virtual boundary based on similar citations
Joint Segmentation: Jnt-Seg-ER

- Jnt-Seg could introduce wrong boundary in “dense” citation lists
- Entity resolution can help:
  Only consider co-referent pairs

\[
\text{InField}(p, +f, c) \land \neg \text{HasPunc}(c, p) \\
\land \neg (\exists c' \text{ JointInferenceCandidate}(c, p, c') \\
\land \neg \text{SameCitation}(c, c')) \\
\Rightarrow \text{InField}(p+1, +f, c)
\]
MLNs for Joint Citation Matching

- Jnt-Seg + ER or Jnt-Seg-ER + ER
- 18 predicates, 22 rules
- State-of-the-art system in citation matching:
  - Best entity resolution results to date
  - Both MLNs outperform isolated segmentation
Outline

- Background: Markov logic
- MLNs for joint citation matching
- Experiments
- Conclusion
Datasets

- **CiteSeer**
  - 1563 citations, 906 clusters
  - Four-fold cross-validation
  - Largest fold: 0.3 million atoms, 0.4 million clauses

- **Cora**
  - 1295 citations, 134 clusters
  - Three-fold cross-validation
  - Largest fold: 0.26 million atoms, 0.38 million clauses
Pre-Processing

- **Standard normalizations**: Case, stemming, etc.

- **Token matching**:
  - Two tokens match ⇔ Differ by at most one character
  - Bigram matches unigram formed by concatenation
Entity Resolution: CiteSeer (Cluster Recall)
## Entity Resolution: Cora

<table>
<thead>
<tr>
<th>Approach</th>
<th>Pairwise Recall</th>
<th>Pairwise Precision</th>
<th>Cluster Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fellegi-Sunter</td>
<td>78.0</td>
<td>97.7</td>
<td>62.7</td>
</tr>
<tr>
<td>Joint MLN</td>
<td>94.3</td>
<td>97.0</td>
<td>78.1</td>
</tr>
</tbody>
</table>
Segmentation: CiteSeer (F1)

Potential Records

<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isolated</td>
<td>Jnt-Seg</td>
<td>Jnt-Seg-ER</td>
</tr>
</tbody>
</table>

- **Author**: 94%
- **Title**: 82%
- **Venue**: 96%
Segmentation: CiteSeer (F1)

All Records

- Isolated
- Jnt-Seg
- Jnt-Seg-ER

Author vs Title vs Venue
Segmentation: Cora (F1)

Potential Records

- Author
- Title
- Venue

- Isolated
- Jnt-Seg
- Jnt-Seg-ER
Segmentation: Cora (F1)

All Records

Author  Title  Venue

- Isolated
- Jnt-Seg
- Jnt-Seg-ER
### Segmentation: Error Reduction by Joint Inference

<table>
<thead>
<tr>
<th></th>
<th>Author</th>
<th>Title</th>
<th>Venue</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CiteSeer</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>6%</td>
<td>3%</td>
<td>0%</td>
</tr>
<tr>
<td>Potential</td>
<td>35%</td>
<td>44%</td>
<td>8%</td>
</tr>
</tbody>
</table>

| **Cora**       |        |       |       |
| All            | 24%    | 7%    | 6%    |
| Potential      | 67%    | 56%   | 17%   |
Conclusion

- Joint inference attracts great interest today
- Yet very difficult to perform effectively
- We propose the first fully joint approach to information extraction, using Markov logic
- Successfully applied to citation matching
  - Best results to date on CiteSeer and Cora
  - Joint inference consistently improves accuracy
  - Use Markov logic & its general efficient algorithms
    ⇒ Far less engineering than Pasula and Wellner