Experiments in Graph-based Semi-Supervised Learning for Class-Instance Acquisition

Partha Pratim Talukdar* (Search Labs, MSR)
Fernando Pereira (Google)

ACL, 2010 (Uppsala, Sweden)

*Work done while at the Univ. of Pennsylvania
Class-Instance Acquisition
Class-Instance Acquisition

• Given an entity, assign human readable descriptors to it
  – e.g., Toyota is a *car manufacturer, japanese company, multinational company*, ...
Class-Instance Acquisition

• Given an entity, assign human readable descriptors to it
  – e.g., Toyota is a car manufacturer, japanese company, multinational company, ...

• Large scale, open domain
Class-Instance Acquisition

• Given an entity, assign human readable descriptors to it
  – e.g., *Toyota* is a *car manufacturer, japanese company, multinational company*, ...

• Large scale, open domain

• Applications
  – Web search, Advertising, etc.
Graph-based Class-Instance Acquisition
# Graph-based Class-Instance Acquisition

<table>
<thead>
<tr>
<th>Set 1</th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Billy Joel (0.75)</strong></td>
<td><strong>Bob Dylan (0.95)</strong></td>
</tr>
<tr>
<td><strong>Johnny Cash (0.73)</strong></td>
<td><strong>Johnny Cash (0.87)</strong></td>
</tr>
<tr>
<td><strong>Billy Joel (0.82)</strong></td>
<td><strong>Billy Joel (0.82)</strong></td>
</tr>
</tbody>
</table>
### Graph-based Class-Instance Acquisition

<table>
<thead>
<tr>
<th>Set 1</th>
<th></th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Musician</strong></td>
<td><strong>Singer</strong></td>
<td></td>
</tr>
<tr>
<td>Billy Joel (0.75)</td>
<td>Bob Dylan (0.95)</td>
<td></td>
</tr>
<tr>
<td>Johnny Cash (0.73)</td>
<td>Johnny Cash (0.87)</td>
<td></td>
</tr>
<tr>
<td>Billy Joel (0.82)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Graph-based Class-Instance Acquisition**

<table>
<thead>
<tr>
<th>Set 1</th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Musician</strong></td>
<td><strong>Singer</strong></td>
</tr>
<tr>
<td><strong>Billy Joel</strong></td>
<td>Bob Dylan</td>
</tr>
<tr>
<td>(0.75)</td>
<td>(0.95)</td>
</tr>
<tr>
<td>Johnny Cash</td>
<td>Johnny Cash</td>
</tr>
<tr>
<td>(0.73)</td>
<td>(0.87)</td>
</tr>
<tr>
<td></td>
<td>Billy Joel</td>
</tr>
<tr>
<td></td>
<td>(0.82)</td>
</tr>
</tbody>
</table>

Clusters: Set 1, Set 2

Cluster ID: 3
Graph-based Class-Instance Acquisition

Table Mining

<table>
<thead>
<tr>
<th>Set 1</th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Musician</strong></td>
<td><strong>Singer</strong></td>
</tr>
<tr>
<td>Billy Joel (0.75)</td>
<td>Bob Dylan (0.95)</td>
</tr>
<tr>
<td>Johnny Cash (0.73)</td>
<td>Johnny Cash (0.87)</td>
</tr>
<tr>
<td></td>
<td>Billy Joel (0.82)</td>
</tr>
</tbody>
</table>

Cluster ID

Extraction Confidence

Pattern
Graph-based Class-Instance Acquisition

Set 1
- Billy Joel (0.75)
- Johnny Cash (0.73)

Set 2
- Bob Dylan (0.95)
- Johnny Cash (0.87)
- Billy Joel (0.82)
Graph-based Class-Instance Acquisition

Set 1
- Billy Joel (0.75)
- Johnny Cash (0.73)

Set 2
- Bob Dylan (0.95)
- Johnny Cash (0.87)
- Billy Joel (0.82)

Cluster ID
- Extraction Confidence
- Pattern
- Seed Classes

Musician
- Singer

Table Mining

Set 1
- 0.75
- 0.73

Set 2
- 0.95
- 0.87
- 0.82

Musician 1.0

Bob Dylan

Johnny Cash

Billy Joel
Can we infer that *Bob Dylan* is also a *Musician*, as that is missing in current extractions?
Graph-based Class-Instance Acquisition [Talukdar et al., EMNLP 2008]
Graph-based Class-Instance Acquisition [Talukdar et al., EMNLP 2008]

Initialization

Set 2
- Bob Dylan: 0.95

Set 1
- Johnny Cash: 0.87
  - Musician 1.0: 0.82
- Billy Joel: 0.73
  - Musician 1.0: 0.75

Seed Labels: Musician 1.0
Predicted Labels: Musician 1.0
Graph-based Class-Instance Acquisition [Talukdar et al., EMNLP 2008]

Iteration 1

Set 1
- Musician 1.0

Set 2
- Musician 0.8

Bob Dylan
- 0.95
- 0.87

Johnny Cash
- 0.82
- 0.73
- Musician 1.0

Billy Joel
- 0.75
- Musician 1.0

Seed Labels

Predicted Labels
Graph-based Class-Instance Acquisition [Talukdar et al., EMNLP 2008]

Set 1
- Musician 1.0

Set 2
- Musician 0.8

Iteration 2

Bob Dylan
- Musician 0.6

Johnny Cash
- Musician 1.0

Billy Joel
- Musician 1.0
- Musician 1.0
- Musician 1.0

Seed Labels
Predicted Labels
Outline

• Graph-based Class-Instance Acquisition
  – Overview & previous work

• Review & comparison of three SSL algorithms
  – LP-ZGL (Zhu+, 2003)
  – Adsorption (Baluja+, 2008)
  – Modified Adsorption (MAD) (Talukdar and Crammer, 2009)

• Additional Semantic Constraints

• Summary
Outline

- Graph-based Class-Instance Acquisition
  - Overview & previous work

- Review & comparison of three SSL algorithms
  - LP-ZGL (Zhu+, 2003)
  - Adsorption (Baluja+, 2008)
  - Modified Adsorption (MAD) (Talukdar and Crammer, 2009)

- Additional Semantic Constraints

- Summary
Comments on the Constructed Graph

Set 2

Bob Dylan
0.95

Johnny Cash
0.87

Billy Joel
0.82

Set 1

Musician 1.0
0.73

Musician 1.0
0.75
Comments on the Constructed Graph

Set 2

0.95

Bob Dylan

0.87

Johnny Cash

0.82

Set 1

0.73

Billy Joel

0.75

Smoothness: Nodes connected by an edge should be assigned similar classes, as enforced by edge weight.
Comments on the Constructed Graph

**Initial Clusters**

- **Set 1**
  - **Bob Dylan**: 1.0
  - **Johnny Cash**: 1.0
  - **Billy Joel**: 1.0

- **Set 2**
  - **Bob Dylan**: 0.95
  - **Johnny Cash**: 0.87
  - **Billy Joel**: 0.82

**Smoothness:**
Nodes connected by an edge should be assigned similar classes, as enforced by edge weight.
Comments on the Constructed Graph

Initial Clusters

Set 1

Set 2

Bob Dylan

Johnny Cash

Billy Joel

Smoothness:
Nodes connected by an edge should be assigned similar classes, as enforced by edge weight.

Coupling Node:
Force (softly) all instance nodes connected to it to have similar class labels, exploiting the Smoothness requirement.
Comments on the Constructed Graph

Set 1

Set 2

Bob Dylan

Johnny Cash

Billy Joel

Seed classes can be different from the cluster IDs

Initial Clusters

Coupling Node:
Force (softly) all instance nodes connected to it to have similar class labels, exploiting the Smoothness requirement.

Smoothness:
Nodes connected by an edge should be assigned similar classes, as enforced by edge weight.
• $m$ labels

• $W$: edge weight matrix

• $\hat{Y}_{ul}$: weight of label $l$ on node $u$

• $Y_{ul}$: seed weight of label $l$ on node $u$

• $S$: diagonal matrix, non-zero for seed nodes
\[
\text{LP-ZGL}
\]

[Zhu et al., ICML 2003]

\[
\arg\min_{\hat{\mathbf{Y}}} \sum_{l=1}^{m} \sum_{u,v} W_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2; \quad \text{s.t. } S\hat{Y}_l = SY_l
\]

- \(m\) labels
- \(W\): edge weight matrix
- \(\hat{Y}_{ul}\): weight of label \(l\) on node \(u\)
- \(Y_{ul}\): seed weight of label \(l\) on node \(u\)
- \(S\): diagonal matrix, non-zero for seed nodes
LP-ZGL
[Zhu et al., ICML 2003]

$$\arg \min_{\hat{Y}} \sum_{l=1}^{m} \sum_{u,v} W_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2$$
\hspace{1cm} \text{s.t. } S\hat{Y}_l = SY_l$$

- $m$ labels
- $W$: edge weight matrix
- $\hat{Y}_{ul}$: weight of label $l$ on node $u$
- $Y_{ul}$: seed weight of label $l$ on node $u$
- $S$: diagonal matrix, non-zero for seed nodes
LP-ZGL
[Zhu et al., ICML 2003]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m} \sum_{u,v} W_{uv}(\hat{Y}_{ul} - \hat{Y}_{vl})^2 \quad \text{s.t.} \quad S\hat{Y}_l = SY_l
\]

- \(m\) labels
- \(W\): edge weight matrix
- \(\hat{Y}_{ul}\): weight of label \(l\) on node \(u\)
- \(Y_{ul}\): seed weight of label \(l\) on node \(u\)
- \(S\): diagonal matrix, non-zero for seed nodes
Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]
Modified Adsorption (MAD)
[Talukdar and Crammer, ECML 2009]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left[ \| S\hat{Y}_l - SY_l \|^2 + \mu_1 \sum_{u,v} M_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \| \hat{Y}_l - R_l \|^2 \right]
\]

- \( m \) labels, +1 dummy label
- \( M = W^\top + W \) is the symmetrized weight matrix
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)
Match Seeds (soft)

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left( \left\| S\hat{Y}_l - SY_l \right\|^2 \right) + \mu_1 \sum_{u,v} M_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \left\| \hat{Y}_l - R_l \right\|^2
\]

- \( m \) labels, +1 dummy label
- \( M = W^\top + W \) is the symmetrized weight matrix
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)
Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left( \| S\hat{Y}_l - SY_l \|^2 + \mu_1 \sum_{u,v} M_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \| \hat{Y}_l - R_l \|^2 \right)
\]

- \( m \) labels, +1 dummy label
- \( M = \mathbf{W}^\top + \mathbf{W} \) is the symmetrized weight matrix
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)

**Match Seeds (soft)**

**Smooth**
Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left[ \| S \hat{Y}_l - S Y_l \|^2 + \mu_1 \sum_{u,v} M_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \| \hat{Y}_l - R_l \|^2 \right]
\]

- \( m \) labels, +1 dummy label
- \( M = W^\top + W \) is the symmetrized weight matrix
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)
Modified Adsorption (MAD)  
[Talukdar and Crammer, ECML 2009]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left( \| S \hat{Y}_l - SY_l \|^2 \right) + \mu_1 \sum_{u,v} M_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \| \hat{Y}_l - R_l \|^2
\]

- \( m \) labels, +1 dummy label
- \( M = W^\top + W \) is the symmetrized weight matrix
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)

MAD has extra regularization compared to LP-ZGL
MAD (contd.)

Inputs $\mathbf{Y}, \mathbf{R} : |V| \times (|L| + 1)$, $\mathbf{W} : |V| \times |V|$, $\mathbf{S} : |V| \times |V|$ diagonal

$\hat{\mathbf{Y}} \leftarrow \mathbf{Y}$

$\mathbf{M} = \mathbf{W} + \mathbf{W}^\top$

$Z_v \leftarrow S_{vv} + \mu_1 \sum_{u \neq v} M_{vu} + \mu_2 \quad \forall v \in V$

repeat
  for all $v \in V$ do
    $\hat{Y}_v \leftarrow \frac{1}{Z_v} \left( (SY)_v + \mu_1 M_{v,v} \hat{Y} + \mu_2 R_v \right)$
  end for
until convergence
MAD (contd.)

Inputs $Y, R: |V| \times (|L| + 1)$, $W: |V| \times |V|$, $S: |V| \times |V|$ diagonal

$\hat{Y} \leftarrow Y$
$M = W + W^\top$
$Z_v \leftarrow S_{vv} + \mu_1 \sum_{u \neq v} M_{vu} + \mu_2 \quad \forall v \in V$

repeat
  for all $v \in V$ do
    $\hat{Y}_v \leftarrow \frac{1}{Z_v} \left( (SY)_v + \mu_1 M_v \hat{Y} + \mu_2 R_v \right)$
  end for
until convergence

• Easily Parallelizable: Scalable
Inputs $Y, R : |V| \times (|L| + 1), W : |V| \times |V|, S : |V| \times |V|$ diagonal

$\hat{Y} \leftarrow Y$

$M = W + W^\top$

$Z_v \leftarrow S_{vv} + \mu_1 \sum_{u \neq v} M_{vu} + \mu_2 \quad \forall v \in V$

repeat

for all $v \in V$ do

$\hat{Y}_v \leftarrow \frac{1}{Z_v} \left( (SY)_v + \mu_1 M_v.\hat{Y} + \mu_2 R_v \right)$

end for

until convergence

- Easily Parallelizable: Scalable
- Importance of a node can be discounted
MAD (contd.)

Inputs $\mathbf{Y}, \mathbf{R} : |V| \times (|L| + 1), \mathbf{W} : |V| \times |V|, \mathbf{S} : |V| \times |V|$ diagonal

$
\hat{Y} \leftarrow \mathbf{Y} \\
\mathbf{M} = \mathbf{W} + \mathbf{W}^\top \\
Z_v \leftarrow \mathbf{S}_{vv} + \mu_1 \sum_{u \neq v} \mathbf{M}_{vu} + \mu_2 \quad \forall v \in V
$

repeat

for all $v \in V$ do

$\hat{Y}_v \leftarrow \frac{1}{Z_v} \left( (\mathbf{SY})_v + \mu_1 \mathbf{M}_v.\hat{Y} + \mu_2 \mathbf{R}_v \right)$

end for

until convergence

- Easily Parallelizable: Scalable
- Importance of a node can be discounted
- Convergence guarantee under mild conditions
Experiments with Public Datasets
Experiments with Public Datasets

• Previous research used proprietary datasets
  – difficult to reproduce and extend
Experiments with Public Datasets

• Previous research used proprietary datasets
  – difficult to reproduce and extend

• Public datasets are used in the paper
  – **Freebase**: relational tables from multiple sources
  – **YAGO** (Suchanek+, 2007): KB curated from Wikipedia and WordNet
  – Gold standard hypernyms: [Pantel+, 2009], WordNet
Experiments with Public Datasets

• Previous research used proprietary datasets
  – difficult to reproduce and extend

• Public datasets are used in the paper
  – Freebase: relational tables from multiple sources
  – YAGO (Suchanek+, 2007): KB curated from Wikipedia and WordNet
  – Gold standard hypernyms: [Pantel+, 2009], WordNet

• Available at: http://www.talukdar.net/datasets/class_inst/
Graph Stats

Statistics of Graphs used in Experiments

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Freebase-1 (Section 3.1)</td>
<td>32970</td>
<td>957076</td>
<td>29.03</td>
<td>1</td>
<td>13222</td>
</tr>
<tr>
<td>Freebase-2 (Section 3.2)</td>
<td>301638</td>
<td>2310002</td>
<td>7.66</td>
<td>1</td>
<td>137553</td>
</tr>
<tr>
<td>TextRunner (Section 3.3)</td>
<td>175818</td>
<td>529557</td>
<td>3.01</td>
<td>1</td>
<td>2738</td>
</tr>
<tr>
<td>YAGO (Section 3.6)</td>
<td>142704</td>
<td>777906</td>
<td>5.45</td>
<td>0</td>
<td>74389</td>
</tr>
<tr>
<td>TextRunner + YAGO (Section 3.6)</td>
<td>237967</td>
<td>1307463</td>
<td>5.49</td>
<td>1</td>
<td>74389</td>
</tr>
</tbody>
</table>
Evaluation Metric: Mean Reciprocal Rank (MRR)

\[
\text{MRR} = \frac{1}{|\text{test-set}|} \sum_{v \in \text{test-set}} \frac{1}{\text{rank}_v(\text{class}(v))}
\]
Evaluation Metric:
Mean Reciprocal Rank (MRR)

\[
\text{MRR} = \frac{1}{|\text{test-set}|} \sum_{v \in \text{test-set}} \frac{1}{\text{rank}_v(\text{class}(v))}
\]

Billy Joel

*Gold Label: Musician*

MRR: 0.5 (1/2)

Linguist 0.4
Musician 0.3
...

Graph-based SSL Comparisons (I)
Graph-based SSL Comparisons (I)

TextRunner Graph, 170 WordNet Classes

Mean Reciprocal Rank (MRR)

Amount of Supervision

LP-ZGL  Adsorption  MAD

Graph with 175k nodes, 529k edges.
Graph-based SSL Comparisons (2)

Freebase-2 Graph, 192 WordNet Classes

Mean Reciprocal Rank (MRR)

Amount of Supervision

LP-ZGL  Adsorption  MAD

Graph with 303k nodes, 2.3m edges.
When is MAD most effective?
When is MAD most effective?
MAD is most effective in dense graphs, where there is greater need for regularization.
Outline

• Graph-based Class-Instance Acquisition
  – Overview & previous work

• Review & comparison of three SSL algorithms
  – LP-ZGL (Zhu+, 2003)
  – Adsorption (Baluja+, 2008)
  – Modified Adsorption (MAD) (Talukdar and Crammer, 2009)

• Additional Semantic Constraints

• Summary
Outline

• Graph-based Class-Instance Acquisition
  – Overview & previous work

• Review & comparison of three SSL algorithms
  – LP-ZGL (Zhu+, 2003)
  – Adsorption (Baluja+, 2008)
  – Modified Adsorption (MAD) (Talukdar and Crammer, 2009)

• Additional Semantic Constraints

• Summary
Can Additional Semantic Constraints Help?
Can Additional Semantic Constraints Help?

Set 2

Isaac Newton

Johnny Cash

Bob Dylan

Set 1
Can Additional Semantic Constraints Help?

Instances with shared attributes are likely to be from the same class.
Can Additional Semantic Constraints Help?

Instances with shared attributes are likely to be from the same class.

Set 1

- Isaac Newton
- Johnny Cash
- Bob Dylan

Set 2

(has_attribute-albums)
Can Additional Semantic Constraints Help?

Graph-based representation makes it easy to incorporate such constraints!

Instances with shared attributes are likely to be from the same class.
Better Classes with YAGO Attributes
Better Classes with YAG0 Attributes

170 WordNet Classes, 10 Seeds per Class, using Adsorption

Mean Reciprocal Rank (MRR)

TextRunner Graph
YAGO Graph
TextRunner + YAGO Graph
Better Classes with YAGO Attributes

170 WordNet Classes, 10 Seeds per Class, using Adsorption

- TextRunner Graph
- YAGO Graph
- TextRunner + YAGO Graph

Graph constructed from TextRunner (UWash) output, 175k nodes, 529k edges
Better Classes with YAGO Attributes

170 WordNet Classes, 10 Seeds per Class, using Adsorption

Mean Reciprocal Rank (MRR)

- TextRunner Graph
- YAGO Graph
- TextRunner + YAGO Graph

Graph constructed from output of YAGO Knowledge Base, 142k nodes, 777k edges
Better Classes with YAGO Attributes

170 WordNet Classes, 10 Seeds per Class, using Adsorption

Mean Reciprocal Rank (MRR)

- TextRunner Graph
- YAGO Graph
- TextRunner + YAGO Graph

Combined graph, with 237k nodes, 1.3m edges
Better Classes with YAGO Attributes

170 WordNet Classes, 10 Seeds per Class, using Adsorption

Mean Reciprocal Rank (MRR)

- TextRunner Graph
- YAGO Graph
- TextRunner + YAGO Graph
Better Classes with YAGO Attributes

Additional semantic constraints help improve performance significantly.

170 WordNet Classes, 10 Seeds per Class, using Adsorption

Mean Reciprocal Rank (MRR)

TextRunner Graph
YAGO Graph
Classes for Attributes

• Classes assigned to attribute nodes

\textit{has_isbn}

\texttt{wordnet_book}
\texttt{wordnet_magazine}
# Classes for Attributes

- Classes assigned to attribute nodes

<table>
<thead>
<tr>
<th>YAGO Attribute</th>
<th>Top-2 WordNet Classes Assigned by MAD (example instances for each class are shown in brackets)</th>
</tr>
</thead>
</table>
| has_currency   | wordnet.country.108544813 (Burma, Afghanistan)  
wordnet.region.108630039 (Aosta Valley, Southern Flinders Ranges) |
| works_at       | wordnet.scientist.110560637 (Aage Niels Bohr, Adi Shamir)  
wordnet.person.100007846 (Catherine Cornelius, Jamie White) |
| has_capital    | wordnet.state.108654360 (Agusan del Norte, Bali)  
wordnet.region.108630039 (Aosta Valley, Southern Flinders Ranges) |
| born_in        | wordnet.boxer.109870208 (George Chuvalo, Fernando Montiel)  
wordnet.chancellor.109906986 (Godon Brown, Bill Bryson) |
| has_isbn       | wordnet.book.106410904 (Past Imperfect, Berlin Diary)  
wordnet.magazine.106595351 (Railway Age, Investors Chronicle) |
Classes for Attributes

- Classes assigned to attribute nodes

<table>
<thead>
<tr>
<th>YAGO Attribute</th>
<th>Top-2 WordNet Classes Assigned by MAD (example instances for each class are shown in brackets)</th>
</tr>
</thead>
<tbody>
<tr>
<td>has_currency</td>
<td>wordnet_country_108544813 (Burma, Afghanistan)</td>
</tr>
<tr>
<td></td>
<td>wordnet_region_108630039 (Aosta Valley, Southern Flinders Ranges)</td>
</tr>
<tr>
<td></td>
<td>has_isbn</td>
</tr>
<tr>
<td></td>
<td>wordnet_book_106410904 (Past Imperfect, Berlin Diary)</td>
</tr>
<tr>
<td></td>
<td>wordnet_magazine_106595351 (Railway Age, Investors Chronicle)</td>
</tr>
</tbody>
</table>

Evidence that attributes propagate right classes.
Summary
Summary

• Compared three graph-based SSL algorithms for class-instance acquisition
  – MAD is consistently most effective, particularly in dense graphs
Summary

• Compared three graph-based SSL algorithms for class-instance acquisition
  – MAD is consistently most effective, particularly in dense graphs

• Better class-instance acquisition with additional semantic constraints
  – easy to add such constraints in graph-based representation
Summary

• Compared three graph-based SSL algorithms for class-instance acquisition
  – MAD is consistently most effective, particularly in dense graphs

• Better class-instance acquisition with additional semantic constraints
  – easy to add such constraints in graph-based representation

• Datasets available: www.talukdar.net/datasets/class_inst/
  – for reproducibility and extension
Thank You!

http://www.talukdar.net/datasets/class_inst/