Global Learning of
Entailment Graphs

Jonathan Berant
Joint work with Ido Dagan and Jacob Goldberger
January, 2011
Task: Entailment (inference) Rules between Predicates

• **Binary predicates**: specify relations between a pair of arguments:
  – $X$ cause an increase in $Y$
  – $X$ treat $Y$

• **Entailment rules**:
  – $Y$ is a symptom of $X$ $\Rightarrow$ $X$ cause $Y$
  – $X$ cause an increase in $Y$ $\Rightarrow$ $X$ affect $Y$
  – $X$’s treatment of $Y$ $\Rightarrow$ $X$ treat $Y$
Motivation

- Textual entailment systems are useful for applications

**QA:**
**Question:** What affects blood pressure?
“Salt causes an increase in blood pressure”

**IE:** X purchase Y

<table>
<thead>
<tr>
<th>Company</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>Coremetrics</td>
</tr>
<tr>
<td>Google</td>
<td>reMail</td>
</tr>
<tr>
<td>Yahoo</td>
<td>Overture</td>
</tr>
</tbody>
</table>

**IR:**
**Query:** symptoms of IBS
“IBS is characterized by vomiting”
Entailment Graphs

- **Nodes**: predicates
- **Edges**: entailment rules

- Entailment is transitive
- Strong connectivity components represent “equivalence”.
- The DIRT rule-base (Lin and Pantel, 2001) uses only pairwise information.
Contributions

- **Global graph learning**: utilizing interactions between entailment rules for learning
- Integer Linear Programming (ILP) formulation
- **Scaling**: applying ILP on larger graphs using *Decomposition* and *Incremental ILP*
Outline

1. Background
2. Learning algorithm (high-level)
3. Focused entailment graphs
4. Typed entailment graphs
5. Conclusions
Background
Local Learning

**Input**: pair of predicates \((p_1, p_2)\)

**Question**: \(p_1 \rightarrow p_2\) ? (or \(p_1 \leftrightarrow p_2\))

- **Sources of information (monolingual corpus):**
  1. Lexicographic: WordNet (Szpektor and Dagan, 2009), FrameNet (Ben-aharon et al, 2010)
  2. Pattern-based (Chklovsky and Pantel, 2004)
  3. Distributional similarity (Lin and Pantel, 2001; Sekine, 2005; Bhagat et al, 2007; Yates and Etzioni, 2009; Poon and Domingos 2010; Schoenmackers et al., 2010)
Properties of Information Sources

1. Lexicographic:
   – WordNet relations: hyponym, derivation, entailment
   – Limited coverage: “X cause a reduction in Y”

2. Pattern-based:
   – “he scared and even startled me”
   – Requires very large corpus (web-scale)
   – Distinguishes different semantic relations:
     • “to X and even Y” vs. “Either X or Y”

3. Distributional similarity
Distribution Similarity

<table>
<thead>
<tr>
<th>X affect Y</th>
<th>X</th>
<th>Y</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>insulin</td>
<td>metabolism</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Zantac</td>
<td>BP</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X treat Y</th>
<th>X</th>
<th>Y</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zantac</td>
<td>BP</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>diet</td>
<td>diabetes</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

- Vary w.r.t to representation and similarity computation
- Good coverage
- Discerning exact semantic relation is difficult

Background → Learning algorithm → Focused entailment graphs → Typed entailment graphs → Conclusions

Global Learning of Entailment Graphs
Global Learning

**Input:** Set of predicates $P$

**Question:** Find $E = \{(p_1, p_2) \mid p_1 \rightarrow p_2\}$

- Snow et al. (2006) presented an algorithm for taxonomy induction.
- At each step add the concept that maximizes the likelihood under a transitivity constraint.
Global Learning

• Resolver (Yates and Etzioni, 2009)
  – Clustering of concepts and relations

• OntoUSP (Poon and Domingos, 2010)
  – Unsupervised semantic parsing
  – Ontology Induction (is-a hierarchy)

• Co-reference Resolution (Finkel and Manning, 2008)

• Temporal Information Extraction (Ling and Weld, 2010)
High-level Description of Algorithm
Learning Entailment Graph Edges

• Two step algorithm:

Input: set of predicates $P$

1. Train a **local** entailment classifier: given the predicates $(p_1, p_2)$, estimate a score for $p_1 \rightarrow p_2$

2. Learn the edges of the graph using the local entailment classifier and a **transitivity constraint**
Local Entailment Classifier – Step 1

Corpus → \( (P, a_1, a_2)_1 \) → \( (P, a_1, a_2)_2 \) → \( \ldots \) → WordNet

Dist Sim:
1. DIRT
2. Binc
3. Cos
4. SR

Positives:
- Treat → Affect
  - (DIRT=0.3, Binc=0.1, Cos=0.9,\ldots)

Negatives:
- Raise → Lower
  - (DIRT=0.0, Binc=0.05, Cos=0.01,\ldots)

Classifier

Distant supervision
Learning Edges - Challenges

• Ambiguity:
  – Focused entailment graphs
  – Typed entailment graphs

• Scalability – problem is NP-hard:
  – ILP formulation
  – Scaling: *decomposition and incremental ILP*
Focused Entailment Graphs
Focused Entailment Graphs

- Argument is instantiated by a target concept *(nausea)*
- Instantiating an argument reduces ambiguity and scalability problems
Motivation - Hierarchical Summarization

- **Scenario**: user queries about a concept *(nausea)*
- **Summarize** facts using a predicate entailment hierarchy interleaved with a concept taxonomies.
Global Learning of Edges

Input: Set of nodes $V$, weighting function $f: V \times V \rightarrow R$

Output: Set of directed edges $E$ respecting transitivity that maximizes sum of weights

• Problem is NP-hard:
  – Reduction from “Transitive Subgraph” (Yannakakis, 1978)

• Integer Linear Programming (ILP) formulation:
  – Optimal solution (not approximation)
  – Often an LP relaxation will provide an integer solution
**Integer Linear Program**

![Graph Diagram]

\[ \hat{G} = \arg \max \sum_{u \neq v} f(u, v) \cdot X_{uv} \]

\[ \forall u, v, w \in V. X_{uv} + X_{vw} - X_{uw} \leq 1 \]

\[ \forall (u, v) \in NEG. X_{uv} = 0 \]

\[ \forall (u, v) \in POS. X_{uv} = 1 \]

\[ X_{uv} \in \{0,1\} \]

- Indicator variable \( X_{uv} \) for every pair of nodes
- Objective function maximizes sum of edge scores
- Transitivity and background-knowledge provide constraints

---

21 Global Learning of Entailment Graphs
Objective Function

- Given a probabilistic classifier that estimates \( P_{uv} = P(X_{uv}=1 | F_{uv}) \) and some simplifying independence assumptions:

\[
\hat{G} = \arg \max_G P(G | F) \\
= \arg \max_G \sum_{u \neq v} \log \frac{P_{uv} \cdot P(X_{uv}=1)}{(1 - P_{uv}) \cdot P(X_{uv}=0)} X_{uv} \\
= \arg \max_G \sum_{u \neq v} \log \frac{P_{uv}}{(1 - P_{uv})} \cdot X_{uv} + \lambda \cdot |E|
\]

- Objective Function

\( f(u,v) \)

prior
Experimental Evaluation

- 50 million word tokens healthcare corpus
- Ten medical students prepared gold standard graphs for 23 medical concepts:
  - Smoking, seizure, headache, lungs, diarrhea, chemotherapy, HPV, Salmonella, Asthma, etc.
- Evaluation:
  - $F_1$ on set of learned edges vs. gold standard
  - $F_1$ on set of learned facts vs. gold standard
Gold Standard Graph
Evaluated algorithms

• Local algorithms
  – Distributional similarity (DIRT, BInc, etc.)
  – WordNet
  – ILP with No transitivity constraints

• Global algorithms
  – ILP/Snow et al. (greedy optimization)
## Results

<table>
<thead>
<tr>
<th></th>
<th>Edges</th>
<th>Propositions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>recall</td>
<td>Precision</td>
<td>$F_1$</td>
</tr>
<tr>
<td>ILP-global</td>
<td>46.0</td>
<td>50.1</td>
<td><strong>43.8</strong></td>
</tr>
<tr>
<td>Greedy</td>
<td>45.7</td>
<td>37.1</td>
<td>36.6</td>
</tr>
<tr>
<td>ILP-local</td>
<td>44.5</td>
<td>45.3</td>
<td>38.1</td>
</tr>
<tr>
<td>Local$_1$</td>
<td>53.5</td>
<td>34.9</td>
<td>37.5</td>
</tr>
<tr>
<td>Local$_2$</td>
<td>52.5</td>
<td>31.6</td>
<td>37.7</td>
</tr>
<tr>
<td>Local*$_1$</td>
<td>53.5</td>
<td>38.0</td>
<td>39.8</td>
</tr>
<tr>
<td>Local*$_2$</td>
<td>52.5</td>
<td>32.1</td>
<td>38.1</td>
</tr>
<tr>
<td>WordNet</td>
<td>10.8</td>
<td>44.1</td>
<td>13.2</td>
</tr>
</tbody>
</table>

- The algorithm significantly outperforms all other baselines.
Precision-recall Curve

Background → Learning algorithm → **Focused entailment graphs** → Typed entailment graphs → Conclusions
Results

<table>
<thead>
<tr>
<th></th>
<th>Global=true/Local=false</th>
<th>Global=false/Local=true</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold standard = true</td>
<td>48</td>
<td>42</td>
</tr>
<tr>
<td>Gold standard = false</td>
<td>78</td>
<td>494</td>
</tr>
</tbody>
</table>

• Global algorithm avoids false positives.
Background → Learning algorithm → **Focused entailment graphs** → Typed entailment graphs → Conclusions

**Local**

- Diarrhea caused by $X$
  - $X$ is a cause of diarrhea
  - $X$ cause diarrhea
  - $X$ result in diarrhea
    - $X$ prevent diarrhea
      - $X$ treat diarrhea
      - $X$ control diarrhea

**Global**

- Diarrhea caused by $X$
  - $X$ is a cause of diarrhea
  - $X$ cause diarrhea
  - $X$ results in diarrhea
    - $X$ prevent diarrhea
      - $X$ treat diarrhea
      - $X$ control diarrhea
Typed Entailment Graphs
Typed Predicates

• Focused graphs learn low-applicability rules

• Predicate variables are typed:
  \[ X_{company} \text{ acquire } Y_{company} \rightarrow Y_{company} \text{ is sold to } X_{company} \]

• Rules of wide-applicability but less ambiguous

• Schoenmackers et al. (EMNLP 2010) used a local algorithm to learn 30,000 entailment rules

• We want to use a global algorithm
Typed Entailment Graphs

• A graph is defined for every pair of types
• “single-type” graphs contain “direct-mapping” edges and “transposed-mapping” edges
• Problems:
  • How to represent “single-type” graphs
  • Hard to solve graphs with >50 nodes
ILP for “single-type graphs”

\[ \hat{G} = \arg \max \sum_{u \neq v} f_x(u, v) \cdot X_{uv} + \sum_{u, v} f_y(u, v) \cdot Y_{uv} \]

\[ \forall u, v, w \in V. X_{uv} + X_{vw} - X_{uw} \leq 1 \]

\[ \forall u, v, w \in V. X_{uv} + Y_{vw} - Y_{uw} \leq 1 \]

\[ \forall u, v, w \in V. Y_{uv} + X_{vw} - Y_{uw} \leq 1 \]

\[ \forall u, v, w \in V. Y_{uv} + Y_{vw} - X_{uw} \leq 1 \]

- The functions \( f_x \) and \( f_y \) provide local scores for direct and reversed mappings
- Cut the size of ILP in half comparing to naïve solution
Decomposition

- **Sparsity:** Most predicates do not entail one another
- **Proposition:** If we can partition the nodes to $U, W$ such that $f(u, w) < 0$ for every $u, w$ then any $(u, w)$ is not an edge in the optimal solution
Decomposition Algorithm

**Input**: Nodes $V$ and function $f:V \times V \rightarrow R$

1. Insert undirected edges for any $(u,v)$ such that $f(u,v) > 0$

2. Find connected components $V_1, \ldots, V_k$

3. For $i = 1 \ldots k$

   $$E_i = \text{ILP-solve}(V_i,f)$$

**Output**: $E_1, \ldots, E_k$ guaranteed to be optimal

- Step 1 and 2 are efficient
Incremental ILP

• Given a good classifier most transitivity constraints are not violated

• Add constraints only if they are violated
Incremental ILP Algorithm

**Input**: Nodes $V$ and function $f: V \times V \rightarrow R$

1. $ACT, VIO = \emptyset$

2. repeat
   a) $E = \text{ILP-solve}(V, f, ACT)$
   b) $VIO = \text{violated}(V, E)$
   c) $ACT = ACT \cup VIO$

3. Until $|VIO| = 0$

**Output**: $E$ guaranteed to be optimal

- Empirically converges in 6 iterations and reduces number of constraints from $10^6$ to $10^3 - 10^4$
Experiment 1 - Transitivity

- 1 million TextRunner tuples over 10,672 typed predicates and 156 types
- Consist ~2,000 typed entailment graphs
- 10 gold standard graphs of sizes: 7, 14, 22, 30, 38, 53, 59, 62, 86 and 118
- Evaluation:
  - $F_1$ on set of edges vs. gold standard
  - Area Under the Curve (AUC)
Evaluated algorithms

• **Local algorithms**
  – Distributional similarity (DIRT, BInc, etc.)
  – Distributional similarity + background knowledge
  – ILP with No transitivity constraints
  – Sherlock rule resource

• **Global algorithm: ILP**
Background → Learning algorithm → Focused entailment graphs → **Typed entailment graphs** → Conclusions

**Precision-Recall Curve**

![Graph showing precision-recall curve with different markers for BInc, clsf, BInc_k, clsf_k, and ILP_scale.](image)

- **BInc**: Blue diamonds
- **clsf**: Red squares
- **BInc_k**: Green triangles
- **clsf_k**: Purple crosses
- **ILP_scale**: Light blue circles

**Axes**:
- **Precision** on the y-axis
- **Recall** on the x-axis

**Legend**:
- BInc: Blue diamonds
- clsf: Red squares
- BInc_k: Green triangles
- clsf_k: Purple crosses
- ILP_scale: Light blue circles

**Conclusion**:
- The graph illustrates the performance of different entailment methods across varying recall values, with BInc showing the highest precision at higher recall values.
Results

<table>
<thead>
<tr>
<th></th>
<th>Micro-average</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>ILP</td>
<td>43.4</td>
<td>42.2</td>
</tr>
<tr>
<td>Clsf</td>
<td>30.8</td>
<td>37.5</td>
</tr>
<tr>
<td>Sherlock</td>
<td>20.6</td>
<td>43.3</td>
</tr>
<tr>
<td>SR</td>
<td>38.4</td>
<td>23.2</td>
</tr>
<tr>
<td>DIRT</td>
<td>25.7</td>
<td>31.0</td>
</tr>
<tr>
<td>BINC</td>
<td>31.8</td>
<td>34.1</td>
</tr>
</tbody>
</table>

- R/P/F₁ at point of maximal micro-F₁
- Transitivity improves rule learning over typed predicates
Experiment 2 - Scalability

- Run ILP with and without *Decompose Incremental-ILP* over ~2,000 graphs
- Compare for various sparsity parameters:
  - Number of unlearned graphs
  - Number of learned rules
Results

<table>
<thead>
<tr>
<th>Sparsity</th>
<th># unlearned graphs</th>
<th># learned rules</th>
<th>Δ(%)</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.75</td>
<td>9/0</td>
<td>6,242/7,466</td>
<td>+20</td>
<td>75</td>
</tr>
<tr>
<td>-1</td>
<td>9/1</td>
<td>16,790/19,396</td>
<td>+16</td>
<td>29</td>
</tr>
<tr>
<td>-0.6</td>
<td>9/3</td>
<td>26,330/29,732</td>
<td>+13</td>
<td>14</td>
</tr>
</tbody>
</table>

• Scaling techniques add 3,500 new rules to best configuration
• Corresponds to 13% increase in relative recall
Conclusions

• Algorithm for learning entailment rules given both local information and a global transitivity constraint
• ILP formulation for learning entailment rules
• Algorithms that scale ILP to larger graphs
• Application for hierarchical summarization of information for query concepts
• Resource of 30,000 domain-independent typed entailment rules
Future Work

• Untyped graphs
  – Ambiguity
  – Scalability

• Add types of edges.

• Improve entailment classifier

Thank you!