



Global Learning of Focused Entailment Graphs

Jonathan Berant

Joint work with Ido Dagan and Jacob Goldberger

10/6/10

Task: Entailment (inference) Rules for Predicates

- Predicates: specify relations between entities or properties of entities.

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 (inference) systems are useful for applications

QA:

Question: What affects blood pressure?

“Salt causes an increase in blood pressure”

IE: X purchase Y

IBM	Coremetrics
Google	reMail
Yahoo	Overture

IR:

Query: symptoms of IBS

“IBS is characterized by vomiting”

Outline

1. Background – an attempt to map the space
2. Entailment graph structure
3. Data Exploration Application
4. Global Learning algorithm
5. Evaluation
6. Future directions

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:
 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; Szpektor and Dagan, 2008; 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
- No mapping of arguments (buy → sell)
- 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

Distributional Similarity

X affect Y		
X	Y	#
insulin	metabolism	7
Zantac	BP	4
...



X treat Y		
X	Y	#
Zantac	BP	9
diet	diabetes	2
...

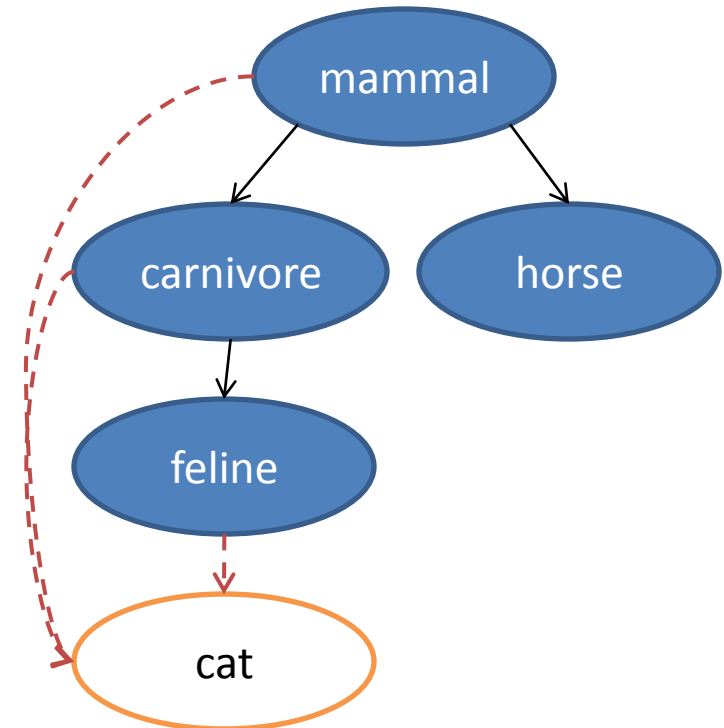
- Vary w.r.t to representation and similarity computation
- Good coverage
- Discerning exact semantic relation is difficult

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 they add the concept that maximizes the likelihood of the taxonomy given the 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)

Mapping the space

- Paraphrasing/entailment
- Local/global
- Sources of information:
 - Distributional similarity
 - Arguments or argument pairs features
 - Lexicographic
 - Patterns
- Supervision
- Representation

*But check out “**Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods**”
Madnani and Dorr, 2010.*

System	Source	Task	supervision	Rep.	method	Arg.
Resolver	distsim	↔	No	String	Global	2
DIRT	distsim	↔	No	Dep.	Local	2
Blnc	distsim	→	No	Dep.	Local	1
TEASE	distsim	↔	No	Dep.	Local	2
OntoUSP	distsim	→	No	QLF	Global	1
Sherlock	distsim	→	No	String	Local	2
AmWN	Lexicographic	→	WordNet	Dep.	Local+	-
Fred	Lexicographic	→	FrameNet	Dep.	Local	-
VerbOcean	Patterns	→	Manual	String	Local+	-
Berant et al.	Lex+distsim	→	Distant	Dep.	Global	1+2

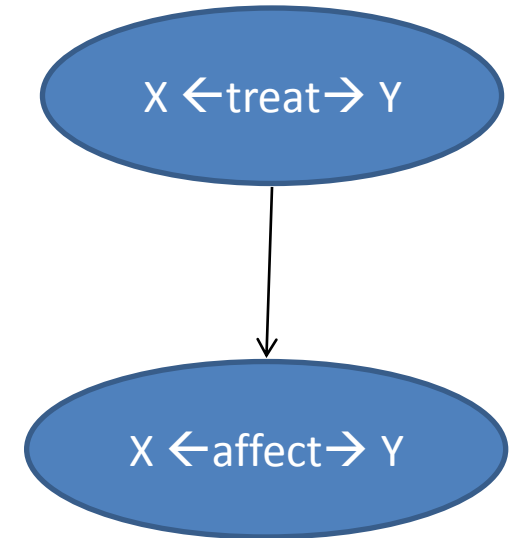
Contributions

- Global graph learning: utilizing interactions between predicates for rule learning
- Optimization through an Integer Linear Programming formulation
 - Alternative to Snow et al. (2006)
- Data exploration application: visualizing facts through a predicate hierarchy

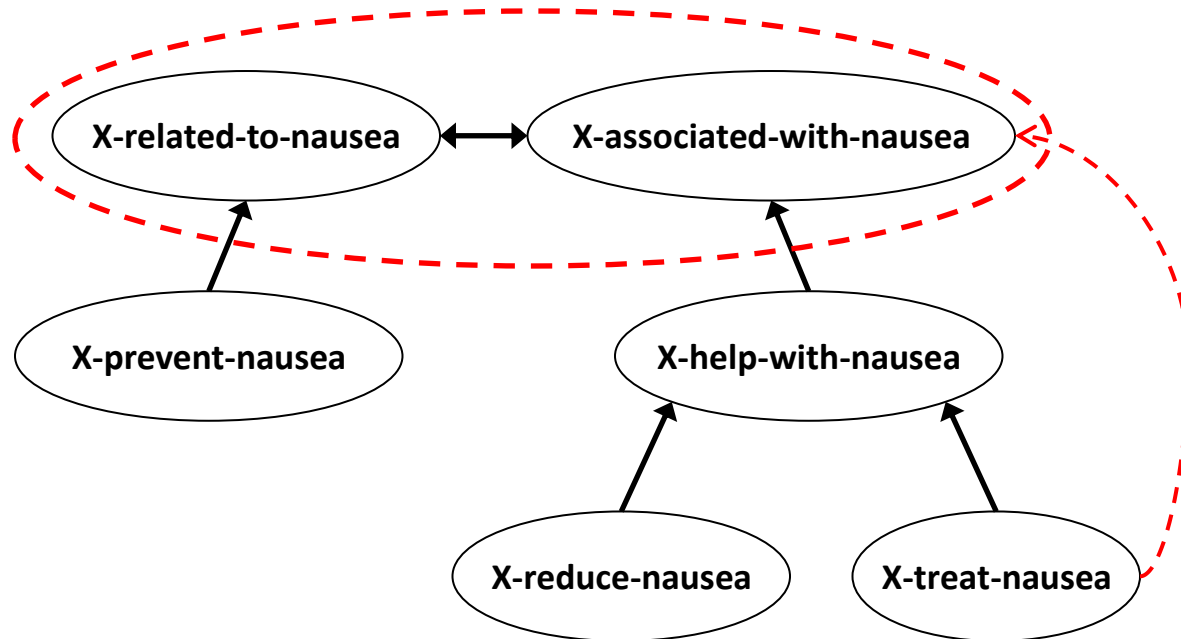
Focused Entailment Graphs

Focused Entailment Graph

- Nodes are dependency paths
 - Lexical predicate
 - Arguments with syntactic relation
 - One variable argument at least
- Edges represent entailment
- Assumption: Predicates are monosomous
 - Small graphs
 - One topic

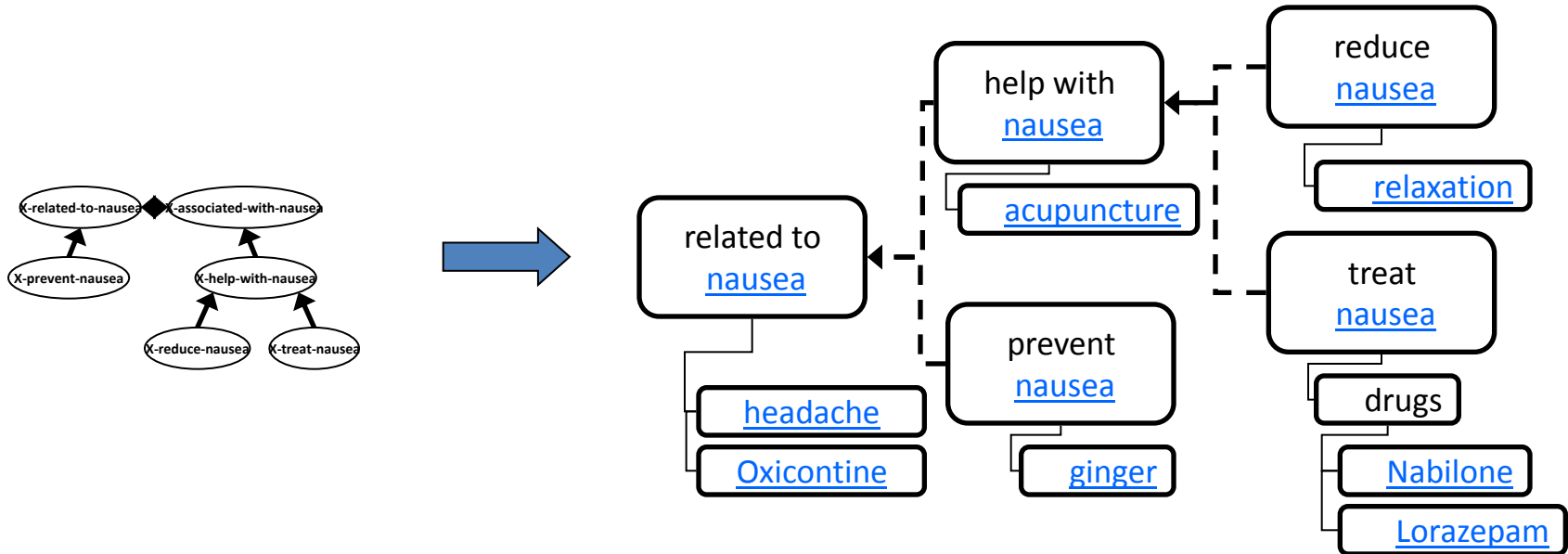


Entailment Graph Properties



- Edges are transitive (monosemous predicates).
- Strong connectivity components represent synonyms.
- Merging strong connectivity components to a single node results in a Directed Acyclic Graph.

Hierarchical Summarization



- Scenario: user queries about a concept (*nausea*)
- Summarize **facts** using a **predicate entailment hierarchy** interleaved with a **taxonomy**.
- Evaluation is based on this application

Algorithm

Learning Entailment Graph Edges

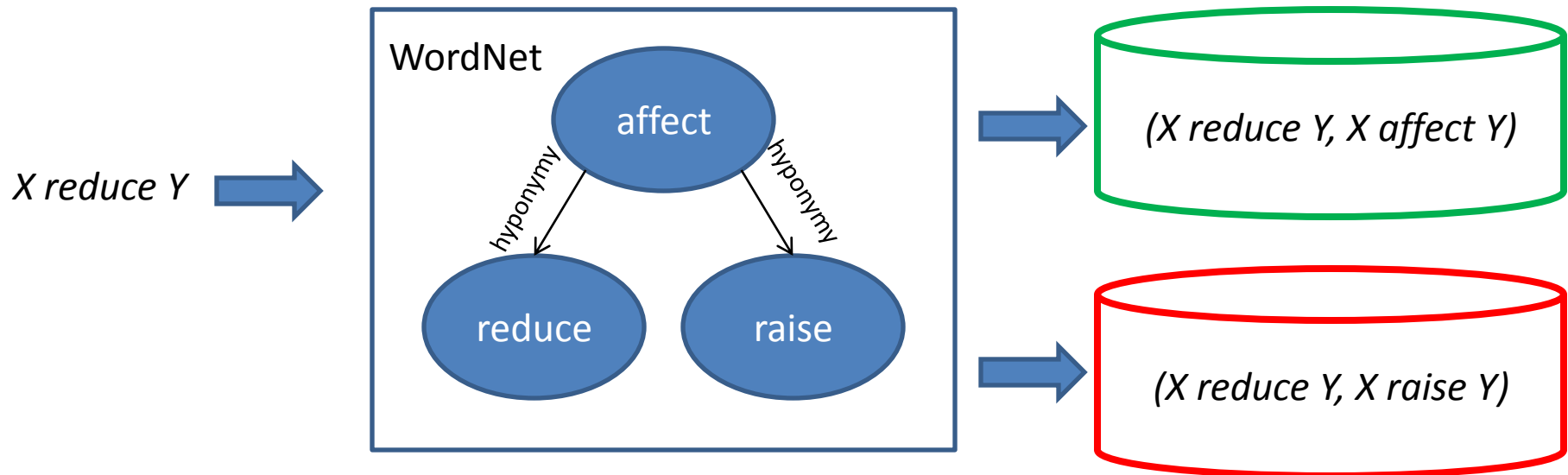
- Two step algorithm:
 1. Train a **local** entailment classifier **once**: given the predicates (p_1, p_2) , estimate a score for $p_1 \rightarrow p_2$
 2. Given the nodes of a focused entailment graph, learn the edges of the graph using the entailment classifier

Entailment Classifier

- Input: large corpus and lexical resource (WN)
- Steps:
 1. Extract “tuples” and predicates from corpus
Aspirin ← treat → headache
 2. Generate automatically positive and negative examples using lexical resource

Train Set Generation

- Use predicates from the corpus and lexical database:



- Generation method similar to “distant supervision” (Snow et al., 2005).

Entailment Classifier

- Input: large corpus and lexical resource (WN)
- Steps:
 1. Extract “tuples” from corpus
Aspirin ← treat → headache
 2. Generate automatically positive and negative examples using lexical resource
 3. Represent training set with similarity meta-features

Representing Template Pairs

- A pair (p_1, p_2) is represented by various distributional similarity algorithms

Measure	score
<i>DIRT</i>	0.549
<i>Blnc</i>	0.919
<i>TEASE</i>	0.711
...	0.0

- Reminiscent of Connor and Roth, 2007.

Entailment Classifier

- Input: large corpus and lexical resource (WN)
- Steps:
 1. Extract “tuples” from corpus
Aspirin ← treat → headache
 2. Generate automatically positive and negative examples using lexical resource
 3. Represent training set with similarity meta-features
- Output: local entailment classifier

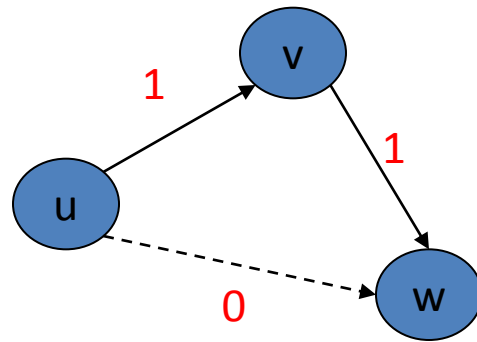
Global Learning of Edges

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

Output: Set of directed edges E respecting transitivity

- Problem is NP-hard:
 - Reduction from “Transitive Graph” (Yannakakis, 1978)
- Integer Linear Programming formulation:
 - Optimal solution (unlike Snow et al., 2006) in polynomial time in our experimental setting

Integer Linear Program



$$\hat{G} = \arg \max \sum_{u \neq v} f(u, v) \cdot I_{uv}$$

$$\forall u, v, w \in V. I_{uv} + I_{vw} - I_{uw} \leq 1$$

$$\forall (u, v) \in NEG. I_{uv} = 0$$

$$\forall (u, v) \in POS. I_{uv} = 1$$

$$I_{uv} \in \{0, 1\}$$

- Binary variables I_{uv} for every pair of nodes
- Objective function maximizes sum of edge scores
- Transitivity and initial information provide constraints

Objective Function

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

$$\hat{G} = \arg \max_G P(G | F)$$

$$= \arg \max_G \sum_{u \neq v} \log \frac{P_{uv} \cdot P(I_{uv} = 1)}{(1 - P_{uv}) \cdot P(I_{uv} = 0)} \cdot I_{uv}$$

$$= \arg \max_G \left(\sum_{u \neq v} \log \frac{P_{uv}}{(1 - P_{uv})} \cdot I_{uv} \right) + \lambda \cdot |E|$$

prior

Objective Function

- Given a margin classifier estimating $S_{uv} \in (-\infty, \infty)$:

$$\hat{G} = \arg \max_G \sum_{u \neq v} (S_{uv} - \lambda) \cdot I_{uv}$$

$$= \arg \max_G \left(\sum_{u \neq v} S_{uv} \cdot I_{uv} \right) - \lambda \cdot |E|$$

Regularization



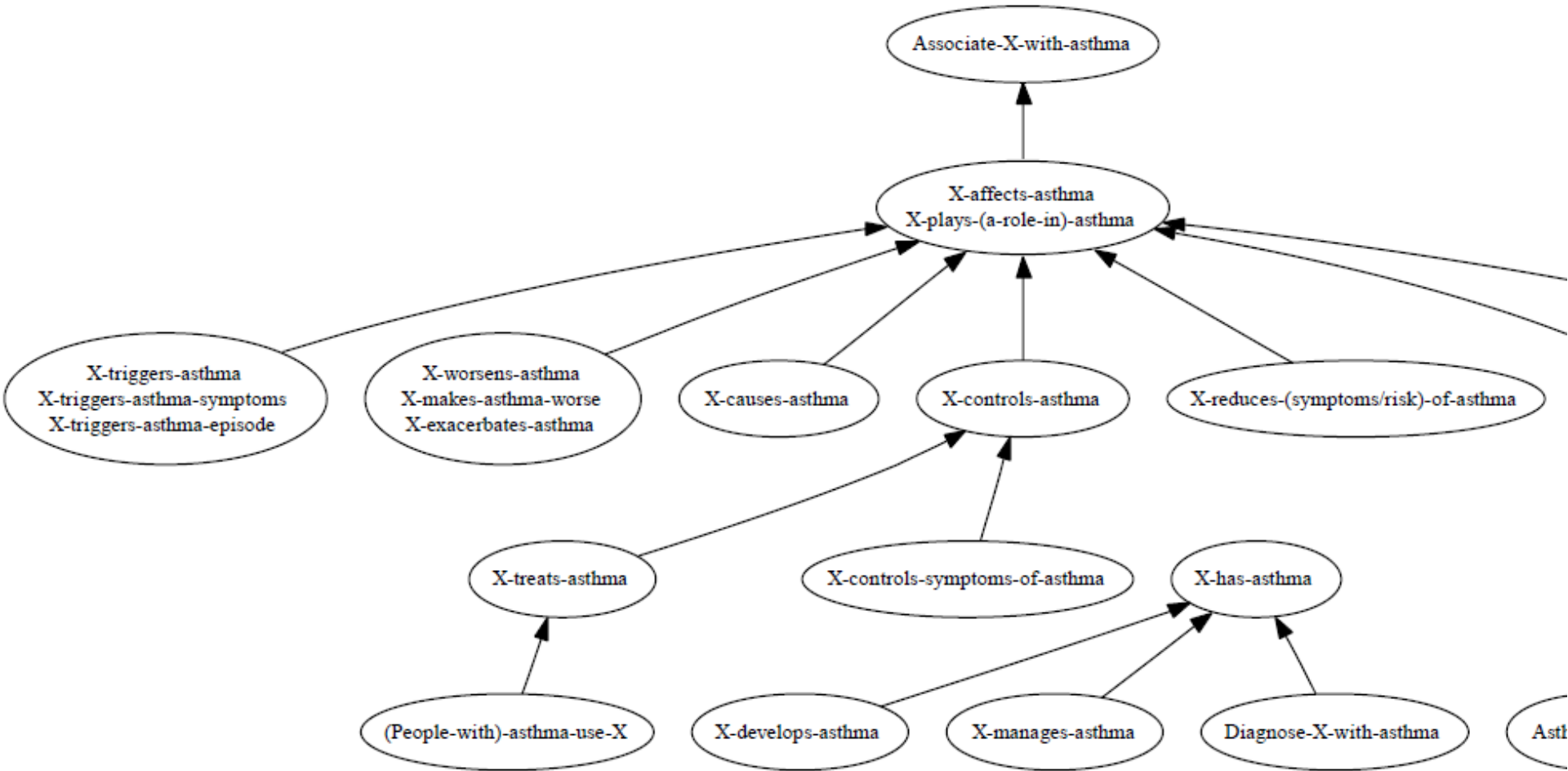
- The two objective functions are similar and can be related to one another

Evaluation

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 edges
 - F_1 on set of propositions

Gold Standard Graph



Evaluated algorithms

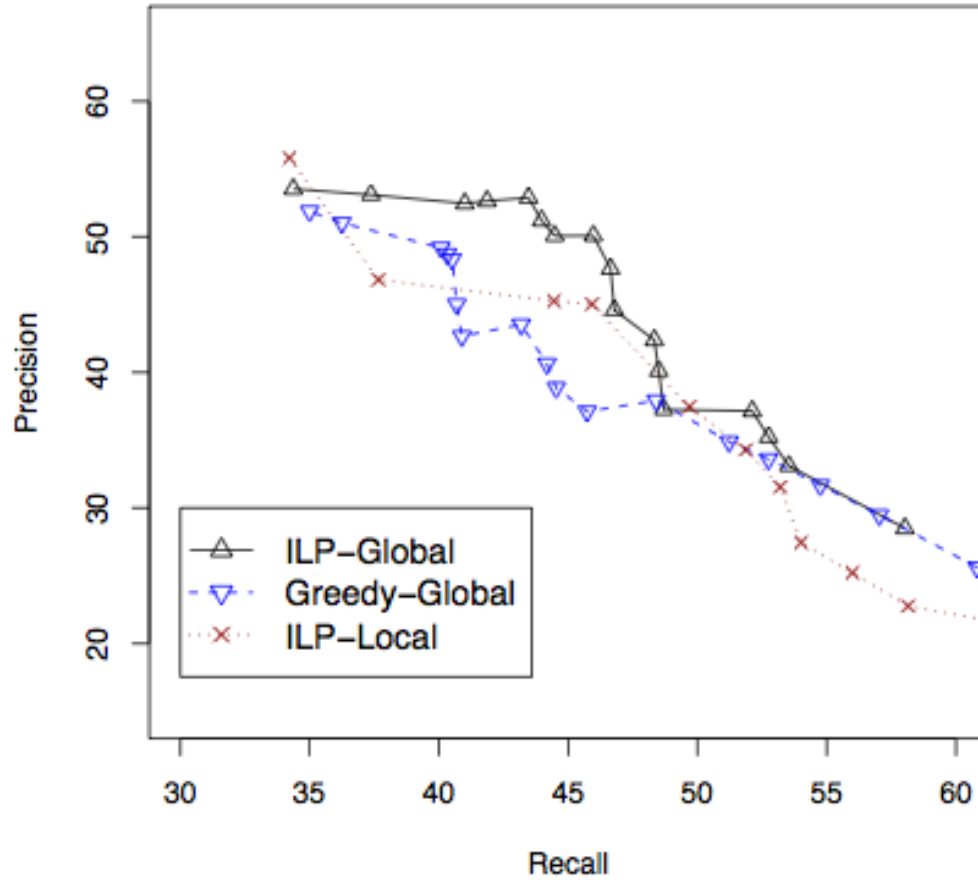
- Local algorithms
 - Distributional similarity (DIRT, Blnc, etc.)
 - WordNet
 - ILP with No transitivity constraints
- Global algorithms
 - ILP/Snow et al. (greedy optimization)

Results

	Edges			Propositions		
	recall	Precision	F ₁	recall	Precision	F ₁
ILP-global	46.0	50.1	43.8*	67.3	69.6	66.2*
Greedy	45.7	37.1	36.6	64.2	57.2	56.3
ILP-local	44.5	45.3	38.1	65.2	61.0	58.6
Local ₁	53.5	34.9	37.5	73.5	50.6	56.1
Local ₂	52.5	31.6	37.7	69.8	50.0	57.1
Local* ₁	53.5	38.0	39.8	73.5	54.6	59.1
Local* ₂	52.5	32.1	38.1	69.8	50.6	57.4
WordNet	10.8	44.1	13.2	39.9	72.4	47.3

- The algorithm significantly outperforms all other baselines.

Recall-Precision Curve

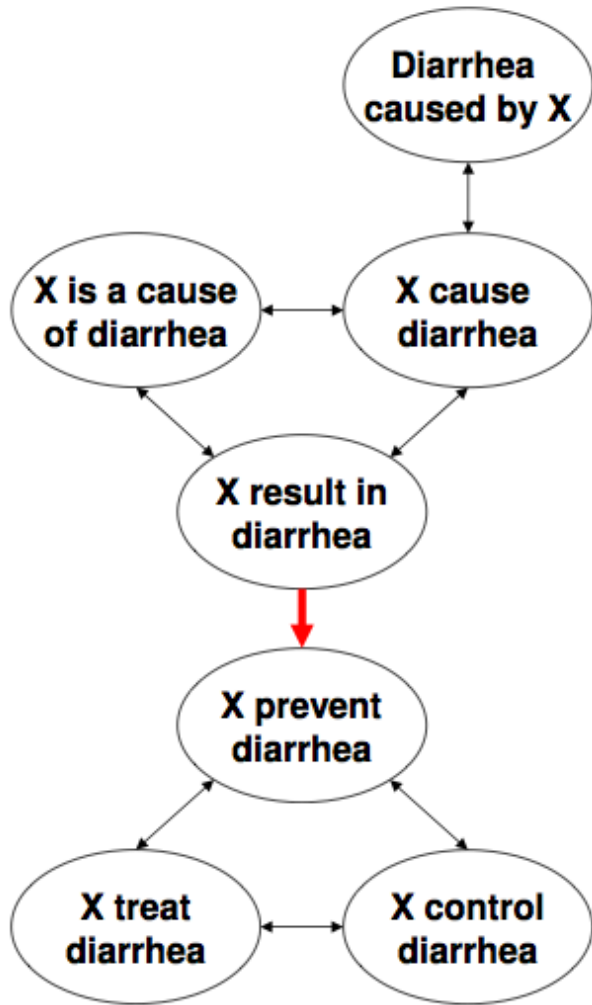


Results

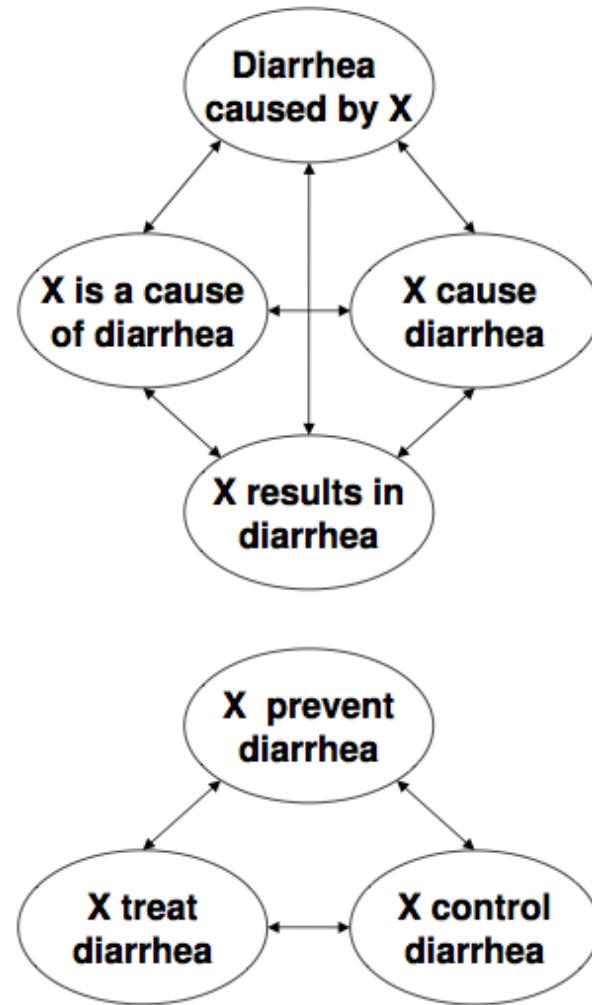
	Global=true/Local=false	Global=false/Local=true
Gold standard = true	48	42
Gold standard = false	78	494

- Global algorithm avoids false positives.

Local



Global



Error analysis

False positives		False negatives	
Classifier error	84.8%	Classifier error	73.5%
Sister term error	18%	“Long” predicate error	36.2%
Direction error	15.1%	Generality error	26.8%
		String overlap error	20.9%

X raise Y → X prevent Y

X affect Y → X raise Y

X cause a reduction in Y → X reduce Y

X cause a reduction in Y → X is associated with Y

Future work

- Scaling graph
- Add types of edges
- Improve entailment classifier
- Joint inference

Entailment Graph Scalability

- Experiments performed on 20-30 domain-specific nodes
- Large domain-independent graphs will provide a larger rule base.
- Problems:
 - Complexity: Number of constraints is $O(V^3)$
 - Ambiguity challenges transitivity constraint

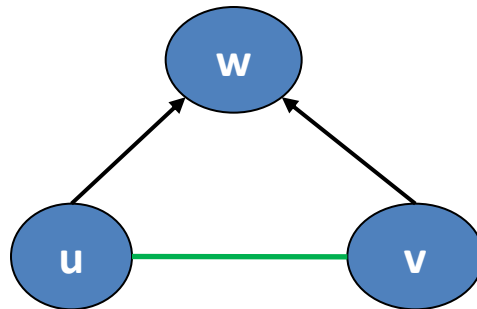
Entailment Graph Scalability

- **Complexity:**
 - Exploit sparseness for a “divide-and-conquer” strategy
 - Change optimization algorithm
 - Incrementally add constraints (cutting-plane method)
 - Less greedy variations of Snow et al.
- **Ambiguity:**
 - Type the arguments (Schoenmakers et al., 2010)
 - Disambiguate predicates
 - Softer transitivity constraints
 - Penalize for violating transitivity (soft ILP or MLNs)
 - Re-score the edges in a global manner

Introducing New Edge Types

- Learn simultaneously entailment edges and sister term edges (*eat* ↔ *drink*)
- New constraints: If two nodes are sisters, they do not entail one another and have a common ancestor
- **Need a sister term classifier (Do and Roth, 2010)**

$$\forall u, v, w. I_{uw} + I_{vw} + (1 - I_{uv}) + (1 - I_{vu}) - C_{uv} \leq 3$$



Improve Entailment Classifier

- The classifier is responsible for most mistakes.
- Improve training signal:
 - Avoid meta-features for distributional similarity
 - Pattern-based methods (**sparseness!**)
 - String and dependency path similarity metrics
 - More...
- Larger corpora for training

Breaking the pipeline

- Currently, distributional information feeds the transitivity information
- But not vice versa...
- Joint inference might improve performance
 - MLNs?

Iterative Algorithm

1. Initialization: train entailment classifier on an automatically generated train set
2. E-step: Use Integer Linear Programming to classify the edges in the set of graphs
3. M-step: Re-estimate the classifier parameters using the test set classification

At each step $P(G | F; \alpha)$ improves

Conclusions

- What was achieved:
 - Algorithm for learning focused entailment graphs using a global transitivity constraint
 - Novel application for data exploration
- Future work:
 - Scaling graph
 - Add edge types
 - Improve entailment classifier
 - Joint inference

Thank you!