Learning, Inference and Supervision for Structured Prediction Tasks

Dan Roth

Department of Computer Science

University of Illinois at Urbana-Champaign

June 2015

NOML Summer School, Mumbai, India

COGNITIVE COMPUTATION GROUP
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

MULTIMODAL INFORMATION ACCESS & SYNTHESIS mias

Learning, Inference and Supervision for Structured Prediction Tasks

Dan Roth

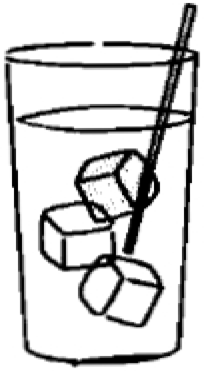
Department of Computer Science

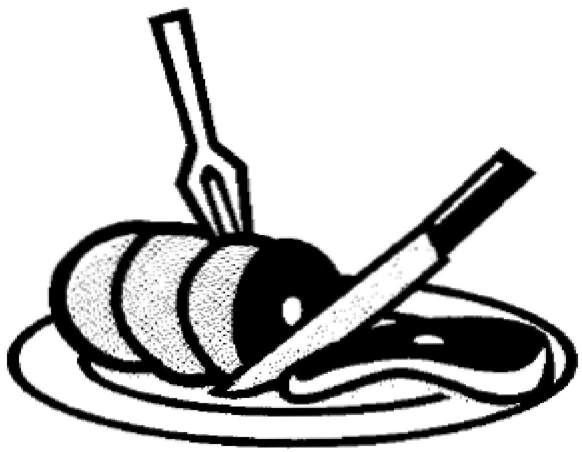
University of Illinois at Urbana-Champaign

With thanks to:

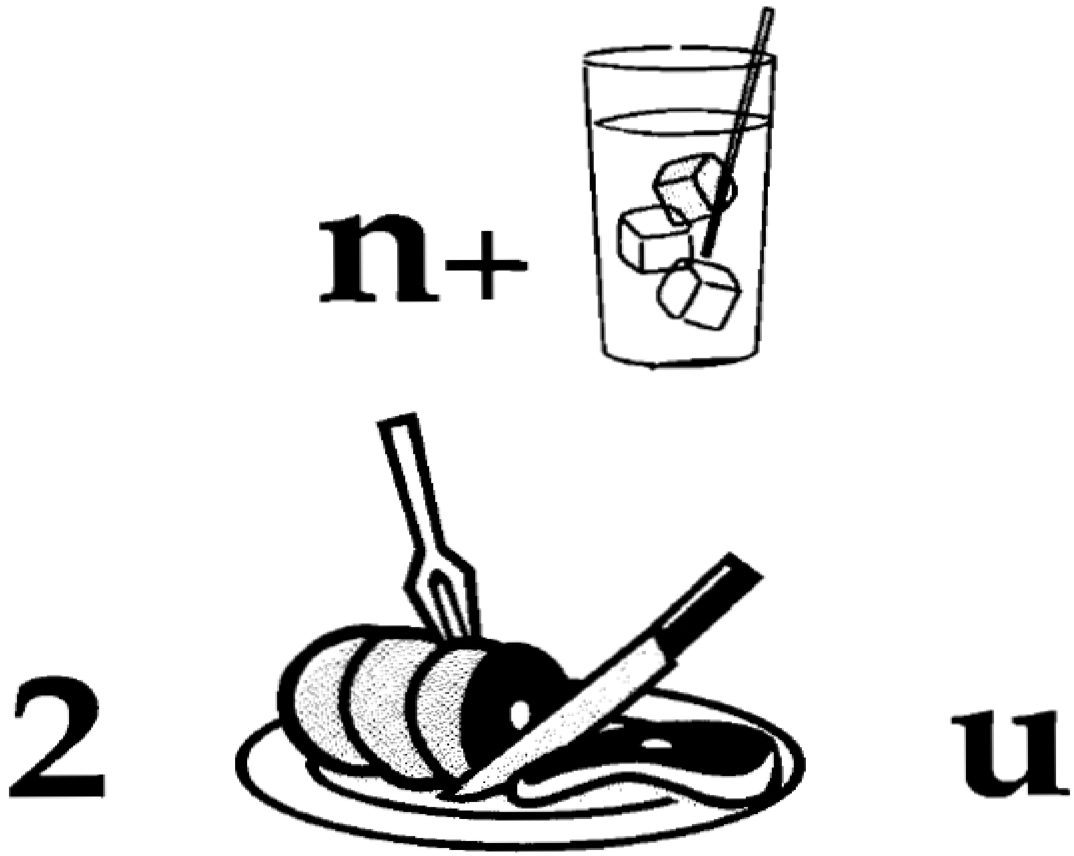
Collaborators: **Kai-Wei Chang, Ming-Wei Chang, Xiao Chen, Dan Goldwasser, Gourab Kundu, Lev Ratinov, Vivek Srikumar; Many others**

Funding: NSF; DHS; NIH; DARPA; IARPA, ARL, ONR
DASH Optimization (Xpress-MP); Gurobi.

n+ 

2  u

Nice to Meet You



Learning and Inference in NLP

- Natural Language Decisions are Structured
  - Global decisions in which several local decisions play a role but there are mutual dependencies on their outcome.

Learning and Inference in NLP

- Natural Language Decisions are Structured
  - Global decisions in which several local decisions play a role but there are mutual dependencies on their outcome.
- It is essential to make coherent decisions in a way that takes the interdependencies into account. Joint, Global Inference.

Learning and Inference in NLP

- Natural Language Decisions are Structured
  - Global decisions in which several local decisions play a role but there are mutual dependencies on their outcome.
- It is essential to make coherent decisions in a way that takes the interdependencies into account. Joint, Global Inference.
- TODAY:

Learning and Inference in NLP

- Natural Language Decisions are Structured
  - Global decisions in which several local decisions play a role but there are mutual dependencies on their outcome.
- It is essential to make coherent decisions in a way that takes the interdependencies into account. Joint, Global Inference.
- TODAY:
  - How to support real, high level, natural language decisions

Learning and Inference in NLP

- Natural Language Decisions are Structured
  - Global decisions in which several local decisions play a role but there are mutual dependencies on their outcome.
- It is essential to make coherent decisions in a way that takes the interdependencies into account. Joint, Global Inference.
- TODAY:
  - How to support real, high level, natural language decisions
  - How to learn models that are used, eventually, to make global decisions

Learning and Inference in NLP

- Natural Language Decisions are Structured
  - Global decisions in which several local decisions play a role but there are mutual dependencies on their outcome.
- It is essential to make coherent decisions in a way that takes the interdependencies into account. Joint, Global Inference.
- TODAY:
  - How to support real, high level, natural language decisions
  - How to learn models that are used, eventually, to make global decisions

  - A framework that allows one to exploit interdependencies among decision variables both in inference (decision making) and in learning.
  - **Inference:** A formulation for incorporating expressive declarative knowledge in decision making.
  - **Learning:** Ability to learn simple models; amplify its power by exploiting interdependencies.

Comprehension

(ENGLAND, June, 1989) - Christopher Robin is alive and well. He lives in England. He is the same person that you read about in the book, Winnie the Pooh. As a boy, Chris lived in a pretty home called Cotchfield Farm. When Chris was three years old, his father wrote a poem about him. The poem was printed in a magazine for others to read. Mr. Robin then wrote a book. He made up a fairy tale land where Chris lived. His friends were animals. There was a bear called Winnie the Pooh. There was also an owl and a young pig, called a piglet. All the animals were stuffed toys that Chris owned. Mr. Robin made them come to life with his words. The places in the story were all near Cotchfield Farm. Winnie the Pooh was written in 1925. Children still love to read about Christopher Robin and his animal friends. Most people don't know he is a real person who is grown now. He has written two books of his own. They tell what it is like to be famous.

Comprehension

(ENGLAND, June, 1989) - Christopher Robin is alive and well. He lives in England. He is the same person that you read about in the book, Winnie the Pooh. As a boy, Chris lived in a pretty home called Cotchfield Farm. When Chris was three years old, his father wrote a poem about him. The poem was printed in a magazine for others to read. Mr. Robin then wrote a book. He made up a fairy tale land where Chris lived. His friends were animals. There was a bear called Winnie the Pooh. There was also an owl and a young pig, called a piglet. All the animals were stuffed toys that Chris owned. Mr. Robin made them come to life with his words. The places in the story were all near Cotchfield Farm. Winnie the Pooh was written in 1925. Children still love to read about Christopher Robin and his animal friends. Most people don't know he is a real person who is grown now. He has written two books of his own. They tell what it is like to be famous.

1. Christopher Robin was born in England.
2. Winnie the Pooh is a title of a book.
3. Christopher Robin's dad was a magician.
4. Christopher Robin must be at least 65 now.

Comprehension

(ENGLAND, June, 1989) - Christopher Robin is alive and well. He lives in England. He is the same person that you read about in the book, Winnie the Pooh. As a boy, Chris lived in a pretty home called Cotchfield Farm. When Chris was three years old, his father wrote a poem about him. The poem was printed in a magazine for others to read. Mr. Robin then wrote a book. He made up a fairy tale land where Chris lived. His friends were animals. There was a bear called Winnie the Pooh. There was also an owl and a young pig, called a piglet. All the animals were stuffed toys that Chris owned. Mr. Robin made them come to life with his words. The places in the story were all near Cotchfield Farm. Winnie the Pooh was written in 1925. Children still love to read about Christopher Robin and his animal friends. Most people don't know he is a real person who is grown now. He has written two books of his own. They tell what it is like to be famous.

1. Christopher Robin was born in England.
2. Winnie the Pooh is a title of a book.
3. Christopher Robin's dad was a magician.
4. Christopher Robin must be at least 65 now.

Comprehension

(ENGLAND, June, 1989) - Christopher Robin is alive and well. He lives in England. He is the same person that you read about in the book, Winnie the Pooh. As a boy, Chris lived in a pretty home called Cotchfield Farm. When Chris was three years old, his father wrote a poem about him. The poem was printed in a magazine for others to read. Mr. Robin then wrote a book. He made up a fairy tale land where Chris lived. His friends were animals. There was a bear called Winnie the Pooh. There was also an owl and a young pig, called a piglet. All the animals were stuffed toys that Chris owned. Mr. Robin made them come to life with his words. The places in the story were all near Cotchfield Farm. Winnie the Pooh was written in 1925. Children still love to read about Christopher Robin and his animal friends. Most people don't know he is a real person who is grown now. He has written two books of his own. They tell what it is like to be famous.

1. Christopher Robin was born in England.
2. Winnie the Pooh is a title of a book.
3. Christopher Robin's dad was a magician.
4. Christopher Robin must be at least 65 now.

Comprehension

(ENGLAND, June, 1989) - Christopher Robin is alive and well. He lives in England. He is the same person that you read about in the book, Winnie the Pooh. As a boy, Chris lived in a pretty home called Cotchfield Farm. When Chris was three years old, his father wrote a poem about him. The poem was printed in a magazine for others to read. Mr. Robin then wrote a book. He made up a fairy tale land where Chris lived. His friends were animals. There was a bear called Winnie the Pooh. There was also an owl and a young pig, called a piglet. All the animals were stuffed toys that Chris owned. Mr. Robin made them come to life with his words. The places in the story were all near Cotchfield Farm. Winnie the Pooh was written in 1925. Children still love to read about Christopher Robin and his animal friends. Most people don't know he is a real person who is grown now. He has written two books of his own. They tell what it is like to be famous.

1. Christopher Robin was born in England.
2. Winnie the Pooh is a title of a book.
3. Christopher Robin's dad was a magician.
4. Christopher Robin must be at least 65 now.

Comprehension

(ENGLAND, June, 1989) - Christopher Robin is alive and well. He lives in England. He is the same person that you read about in the book, Winnie the Pooh. As a boy, Chris lived in a pretty home called Cotchfield Farm. When Chris was three years old, his father wrote a poem about him. The poem was printed in a magazine for others to read. Mr. Robin then wrote a book. He made up a fairy tale land where Chris lived. His friends were animals. There was a bear called Winnie the Pooh. There was also an owl and a young pig, called a piglet. All the animals were stuffed toys that Chris owned. Mr. Robin made them come to life with his words. The places in the story were all near Cotchfield Farm. Winnie the Pooh was written in 1925. Children still love to read about Christopher Robin and his animal friends. Most people don't know he is a real person who is grown now. He has written two books of his own. They tell what it is like to be famous.

1. Christopher Robin was born in England.
2. Winnie the Pooh is a title of a book.
3. Christopher Robin's dad was a magician.
4. Christopher Robin must be at least 65 now.

Comprehension

(ENGLAND, June, 1989) - Christopher Robin is alive and well. He lives in England. He is the same person that you read about in the book, Winnie the Pooh. As a boy, Chris lived in a pretty home called Cotchfield Farm. When Chris was three years old, his father wrote a poem about him. The poem was printed in a magazine for others to read. Mr. Robin then wrote a book. He made up a fairy tale land where Chris lived. His friends were animals. There was a bear called Winnie the Pooh. There was also an owl and a young pig, called a piglet. All the animals were stuffed toys that Chris owned. Mr. Robin made them come to life with his words. The places in the story were all near Cotchfield Farm. Winnie the Pooh was written in 1925. Children still love to read about Christopher Robin and his animal friends. Most people don't know he is a real person who is grown now. He has written two books of his own. They tell what it is like to be famous.

1. Christopher Robin was born in England.
2. Winnie the Pooh is a title of a book.
3. Christopher Robin's dad was a magician.
4. Christopher Robin must be at least 65 now.

This is an Inference Problem

Learning and Inference

- Natural language understanding decisions are global decisions in which several local decisions play a role, but there are mutual dependencies on their outcome.

Learning and Inference

- Natural language understanding decisions are global decisions in which several local decisions play a role, but there are mutual dependencies on their outcome.

  - We need to think about:
    - **(Learned) models for different sub-problems**
    - **Reasoning with knowledge relating sub-problems**
    - **Knowledge that may appear only at evaluation time**

Learning and Inference

- Natural language understanding decisions are global decisions in which several local decisions play a role, but there are mutual dependencies on their outcome.
    - We need to think about:
        - **(Learned) models for different sub-problems**
        - **Reasoning with knowledge relating sub-problems**
        - **Knowledge that may appear only at evaluation time**
- Goal: Incorporate models' information, along with knowledge (constraints) in making coherent decisions
    - Decisions that respect the local models as well as domain & context specific knowledge/constraints.

Learning and Inference

- Natural language understanding decisions are global decisions in which several local decisions play a role, but there are mutual dependencies on their outcome.

Natural Language Interpretation is an Inference Problem that is best thought of as a knowledge constrained optimization problem, done on top of multiple statistically learned models.

  - We need to think about:
    - **(Learned) models for different sub-problems**
    - **Reasoning with knowledge relating sub-problems**
    - **Knowledge that may appear only at evaluation time**
- Goal: Incorporate models' information, along with knowledge (constraints) in making coherent decisions
  - Decisions that respect the local models as well as domain & context specific knowledge/constraints.

Learning and Inference

- Natural language understanding decisions are global decisions in which several local decisions play a role, but there are mutual dependencies on their outcome.

Natural Language Interpretation is an Inference Problem that is best thought of as a knowledge constrained optimization problem, done on top of multiple statistically learned models.

  - We need to think about:
    - **(Learned) models for different sub-problems**
    - **Reasoning with knowledge relating sub-problems**
    - **Knowledge that may appear only at evaluation time**
- Goal: Incorporate models' information, along with knowledge (constraints) in making coherent decisions
  - Decisions that respect the local models as well as domain & context specific knowledge/constraints.

Many forms of Inference; a lot boil down to determining best assignment

Outline

- Constrained Conditional Models
  - A formulation for global inference with knowledge modeled as expressive structural constraints
  - Some examples
- Learning with Constrained Latent Representation
- Constraints Driven Learning
  - Training Paradigms for Constrained Conditional Models
  - Constraints Driven Learning (CoDL)
  - Unified (Constrained) Expectation Maximization
- Amortized Integer Linear Programming Inference
  - Exploiting Previous Inference Results
    - **In Inference and in Structured Learning**

Three Ideas Underlying Constrained Conditional Models

- **Idea 1:**

  Separate modeling and problem formulation from algorithms

  - Similar to the philosophy of probabilistic modeling

- **Idea 2:**

  Keep models simple, make expressive decisions (via constraints)

  - Unlike probabilistic modeling, where models become more expressive

- **Idea 3:**

  Expressive structured decisions can be supported by simply learned models

  - Global Inference can be used to amplify simple models (and even allow training with minimal supervision).

Three Ideas Underlying Constrained Conditional Models

Modeling

- **Idea 1:**

  Separate modeling and problem formulation from algorithms

  - Similar to the philosophy of probabilistic modeling

- **Idea 2:**

  Keep models simple, make expressive decisions (via constraints)

  - Unlike probabilistic modeling, where models become more expressive

- **Idea 3:**

  Expressive structured decisions can be supported by simply learned models

  - Global Inference can be used to amplify simple models (and even allow training with minimal supervision).

Three Ideas Underlying Constrained Conditional Models

- **Idea 1:**

  Modeling

  Separate modeling and problem formulation from algorithms
  - Similar to the philosophy of probabilistic modeling

- **Idea 2:**

  Inference

  Keep models simple, make expressive decisions (via constraints)
  - Unlike probabilistic modeling, where models become more expressive

- **Idea 3:**

  Expressive structured decisions can be supported by simply learned models
  - Global Inference can be used to amplify simple models (and even allow training with minimal supervision).

Three Ideas Underlying Constrained Conditional Models

- **Idea 1:** Modeling

  Separate modeling and problem formulation from algorithms
  - Similar to the philosophy of probabilistic modeling

- **Idea 2:** Inference

  Keep models simple, make expressive decisions (via constraints)
  - Unlike probabilistic modeling, where models become more expressive

- **Idea 3:** Learning

  Expressive structured decisions can be supported by simply learned models
  - Global Inference can be used to amplify simple models (and even allow training with minimal supervision).

Inference with General Constraint Structure [Roth&Yih'04,07]

Recognizing Entities and Relations

$\boxed{\text{Dole}}$'s wife, $\boxed{\text{Elizabeth}}$, is a native of $\boxed{\text{N.C.}}$

$$E1 \xrightarrow{R_{12}} E2 \xrightarrow{R_{23}} E3$$

COGNITIVE COMPUTATION GROUP
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Inference with General Constraint Structure [Roth&Yih'04,07]

Recognizing Entities and Relations

other	0.05
per	0.85
loc	0.10

other	0.10
per	0.60
loc	0.30

other	0.05
per	0.50
loc	0.45

Dole 's wife, Elizabeth , is a native of N.C.

$E1 \xrightarrow{R_{12}} E2 \xrightarrow{R_{23}} E3$

irrelevant	0.05
spouse_of	0.45
born_in	0.50

irrelevant	0.10
spouse_of	0.05
born_in	0.85

Inference with General Constraint Structure [Roth&Yih'04,07]

Recognizing Entities and Relations

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

other	0.05
per	**0.50**
loc	0.45

Dole 's wife, Elizabeth , is a native of N.C.

E1 $\xrightarrow{R_{12}}$ E2 $\xrightarrow{R_{23}}$ E3

irrelevant	0.05
spouse_of	0.45
born_in	**0.50**

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Inference with General Constraint Structure [Roth&Yih'04,07]

Recognizing Entities and Relations

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

other	0.05
per	**0.50**
loc	0.45

Dole 's wife, Elizabeth , is a native of N.C.

$E_1 \xrightarrow{R_{12}} E_2 \xrightarrow{R_{23}} E_3$

irrelevant	0.05
spouse_of	0.45
born_in	**0.50**

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Inference with General Constraint Structure [Roth&Yih'04,07]

Recognizing Entities and Relations

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

other	0.05
per	0.50
loc	**0.45**

Dole 's wife, Elizabeth , is a native of N.C.

$E_1 \xrightarrow{R_{12}} E_2 \xrightarrow{R_{23}} E_3$

irrelevant	0.05
spouse_of	0.45
born_in	**0.50**

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Inference with General Constraint Structure [Roth&Yih'04,07]

Recognizing Entities and Relations

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

other	0.05
per	0.50
loc	**0.45**

Dole 's wife, Elizabeth , is a native of N.C.

$E_1 \xrightarrow{R_{12}} E_2 \xrightarrow{R_{23}} E_3$

irrelevant	0.05
spouse_of	0.45
born_in	**0.50**

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Inference with General Constraint Structure [Roth&Yih'04,07]

Recognizing Entities and Relations

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

other	0.05
per	0.50
loc	**0.45**

Dole 's wife, Elizabeth , is a native of N.C.

E1 $\xrightarrow{R_{12}}$ E2 $\xrightarrow{R_{23}}$ E3

irrelevant	0.05
spouse_of	**0.45**
born_in	0.50

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Inference with General Constraint Structure [Roth&Yih

Improvement over no inference: 2-5%

Recognizing Entities and Relations

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

othe	0.05
[illegible]	0.50
loc	**0.45**

Dole 's wife, Elizabeth , is a native of N.C.

$E1 \xrightarrow{R_{12}} E2 \xrightarrow{R_{23}} E3$

irrelevant	0.05
spouse_of	**0.45**
born_in	0.50

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Inference with General Constraint Structure [Roth&Yih

Recognizing Entities and Relations

Improvement over no inference: 2-5%

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

othe	0.05
[illegible]	0.50
loc	**0.45**

Note: **Non Sequential Model**

Dole 's wife, Elizabeth , is a native of N.C.

E_1 $\xrightarrow{R_{12}}$ E_2 $\xrightarrow{R_{23}}$ E_3

irrelevant	0.05
spouse_of	**0.45**
born_in	0.50

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Inference with General Constraint Structure [Roth&Yih

Recognizing Entities and Relations

Improvement over no inference: 2-5%

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

othe	0.05
	0.50
loc	**0.45**

Note: **Non Sequential Model**

Dole 's wife, Elizabeth , is a na

E1 → E2

R_{12} R_{23}

Key Questions:

How to guide the global inference?

How to learn? Why not Jointly?

irrelevant	0.05
spouse_of	**0.45**
born_in	0.50

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Inference with General Constraint Structure [Roth&Yih

Recognizing Entities and Relations

Improvement over no inference: 2-5%

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

othe	0.05
	0.50
loc	**0.45**

Note: **Non Sequential Model**

Dole 's wife, Elizabeth , is a na

E1 E2

R_{12} R_{23}

Key Questions:

How to guide the global inference?

How to learn? Why not Jointly?

irrelevant	0.05
spouse_of	**0.45**
born_in	0.50

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Models could be learned separately; constraints may come up only at decision time.

Inference with General Constraint Structure [Roth&Yi

Recognizing Entities and Relations

Improvement over no inference: 2-5%

other	0.05

other	0.10

othe	0.05

$$Y = \text{argmax} \sum_y \text{score}(y=v)\ [[y=v]] =$$

$$= \text{argmax}\ \text{score}(E_1 = \text{PER})\cdot\ [[E_1 = \text{PER}]] + \text{score}(E_1 = \text{LOC})\cdot\ [[E_1 = \text{LOC}]] + \ldots$$
$$\text{score}(R_1 = \text{S-of})\cdot\ [[R_1 = \text{S-of}]] + \ldots..$$

Subject to Constraints

irrelevant	0.05
spouse_of	**0.45**
born_in	0.50

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Models could be learned separately; constraints may come up only at decision time.

Inference with General Constraint Structure [Roth&Yih

Improvement over no inference: 2-5%

Recognizing Entities and Relations

other	0.05	other	0.10	othe	0.05

Y = argm

= argm

scor

Subject to Constraints

An Objective function that incorporates learned models with knowledge (constraints)

A constrained Conditional Model

irrelevant	0.05
spouse_of	**0.45**
born_in	0.50

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Models could be learned separately; constraints may come up only at decision time.

Constrained Conditional Models

$$\operatorname*{argmax}_{y} \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Constrained Conditional Models

$$\operatorname*{argmax}_{y} \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Constrained Conditional Models

$$\operatorname*{argmax}_{y} \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Weight Vector for "local" models

Constrained Conditional Models

$$\underset{y}{\operatorname{argmax}} \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Weight Vector for "local" models

Features, classifiers; log-linear models (HMM, CRF) or a combination

Constrained Conditional Models

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

(Soft) constraints component

Weight Vector for "local" models

Features, classifiers; log-linear models (HMM, CRF) or a combination

Constrained Conditional Models

Penalty for violating the constraint.

$$\operatorname*{argmax}_{y} \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

(Soft) constraints component

Weight Vector for "local" models

Features, classifiers; log-linear models (HMM, CRF) or a combination

How far y is from a "legal" assignment

Constrained Conditional Models

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Penalty for violating the constraint.

(Soft) constraints component

Weight Vector for "local" models

Features, classifiers; log-linear models (HMM, CRF) or a combination

How far y is from a "legal" assignment

How to solve?

This is an Integer Linear Program

Solving using ILP packages gives an exact solution.

Cutting Planes, Dual Decomposition & other search techniques are possible

How to train?

Training is learning the objective function

Decouple? Decompose?

How to exploit the structure to minimize supervision?

Structured Prediction: Inference

- Inference: given input **x** (a document, a sentence),

  predict **the best structure** $y = \{y_1,y_2,\ldots,y_n\} \in Y$ (entities & relations)

  - Assign values to the $y_1,y_2,\ldots,y_n$, accounting for **dependencies among** y_is

Structured Prediction: Inference

- Inference: given input **x** (a document, a sentence),

  predict **the best structure** $y = \{y_1, y_2, \ldots, y_n\} \in Y$ (entities & relations)
  - Assign values to the $y_1, y_2, \ldots, y_n$, accounting for **dependencies among** y_is

Structured Prediction: Inference

- Inference: given input **x** (a document, a sentence),

  predict **the best structure** $y = \{y_1,y_2,\ldots,y_n\} \in Y$ (entities & relations)

  - Assign values to the $y_1,y_2,\ldots,y_n$, accounting for **dependencies among** y_is
- Inference is expressed as a maximization of a ***scoring function***

$$y' = \mathrm{argmax}_{y \in \mathcal{Y}}\, w^T \phi(x,y)$$

Structured Prediction: Inference

- Inference: given input **x** (a document, a sentence),

  predict **the best structure** $\mathbf{y} = \{y_1, y_2, \ldots, y_n\} \in Y$ (entities & relations)

  - Assign values to the $y_1, y_2, \ldots, y_n$, accounting for **dependencies among** y_is
- Inference is expressed as a maximization of a ***scoring function***

$$y' = \text{argmax}_{y \in \mathcal{Y}} w^T \phi(x,y)$$

Joint features on inputs and outputs

Structured Prediction: Inference

- Inference: given input **x** (a document, a sentence),

  predict **the best structure** $y = \{y_1, y_2, \ldots, y_n\} \in Y$ (entities & relations)

  - Assign values to the $y_1, y_2, \ldots, y_n$, accounting for **dependencies among** y_is
- Inference is expressed as a maximization of a ***scoring function***

$$y' = \text{argmax}_{y \in \mathcal{Y}} \, w^T \phi(x,y)$$

Joint features on inputs and outputs

Feature Weights (estimated during learning)

Structured Prediction: Inference

- Inference: given input **x** (a document, a sentence),

  predict **the best structure** $\mathbf{y} = \{y_1, y_2, \ldots, y_n\} \in \mathrm{Y}$ (entities & relations)

  - Assign values to the $y_1, y_2, \ldots, y_n$, accounting for **dependencies among** y_is
- Inference is expressed as a maximization of a ***scoring function***

$$y' = \mathrm{argmax}_{y \in \mathcal{Y}} \, w^T \phi(x,y)$$

Set of allowed structures

Feature Weights
(estimated during learning)

Joint features on inputs and outputs

Structured Prediction: Inference

- Inference: given input **x** (a document, a sentence),

  predict **the best structure** $y = \{y_1, y_2, \ldots, y_n\} \in Y$ (entities & relations)

  - Assign values to the $y_1, y_2, \ldots, y_n$, accounting for **dependencies among** y_is
- Inference is expressed as a maximization of a ***scoring function***

$$y' = \mathrm{argmax}_{y \in \mathcal{Y}}\, w^T \phi\,(x,y)$$

Set of allowed structures

Feature Weights (estimated during learning)

Joint features on inputs and outputs

- Inference requires, in principle, touching all $y \in Y$ at decision time, when we are given $x \in X$ and attempt to determine the best $y \in Y$ for it, given w

Structured Prediction: Inference

- Inference: given input **x** (a document, a sentence),

  predict **the best structure** $y = \{y_1, y_2, \ldots, y_n\} \in Y$ (entities & relations)

  - Assign values to the $y_1, y_2, \ldots, y_n$, accounting for **dependencies among** y_is
- Inference is expressed as a maximization of a ***scoring function***

$$y' = \operatorname{argmax}_{y \in \mathcal{Y}} w^T \phi(x,y)$$

Set of allowed structures

Feature Weights (estimated during learning)

Joint features on inputs and outputs

- Inference requires, in principle, touching all $y \in Y$ at decision time, when we are given $x \in X$ and attempt to determine the best $y \in Y$ for it, given w
  - For some structures, inference is computationally easy.
  - Eg: Using the Viterbi algorithm
  - In general, NP-hard (can be formulated as an ILP)

Structured Prediction: Learning

- Learning: given a set of structured examples {(x,y)}
  find a scoring function w that minimizes empirical loss.

Structured Prediction: Learning

- Learning: given a set of structured examples $\{(x,y)\}$

  find a scoring function w that minimizes empirical loss.
- Learning is thus driven by the attempt to find a weight vector w such that for each given annotated example (x_i, y_i):

Structured Prediction: Learning

- Learning: given a set of structured examples {(x,y)}
  find a scoring function w that minimizes empirical loss.
- Learning is thus driven by the attempt to find a weight vector w such that for each given annotated example (x_i, y_i):

Score of annotated structure $\geq$ **Score of any other structure** $+$ **Penalty for predicting other structure**

Structured Prediction: Learning

- Learning: given a set of structured examples {(x,y)}
  find a scoring function w that minimizes empirical loss.
- Learning is thus driven by the attempt to find a weight vector w such that for each given annotated example (x_i, y_i):

$$\mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) \geq \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}_i)$$

Structured Prediction: Learning

- Learning: given a set of structured examples {(x,y)}
  find a scoring function w that minimizes empirical loss.
- Learning is thus driven by the attempt to find a weight vector w such that for each given annotated example (x_i, y_i):

$$\forall y \quad \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) \geq \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}_i)$$

Structured Prediction: Learning

- Learning: given a set of structured examples $\{(x,y)\}$
  find a scoring function w that minimizes empirical loss.
- Learning is thus driven by the attempt to find a weight vector w such that for each given annotated example (x_i, y_i):

$$\forall\, y \quad \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) \geq \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}_i)$$

- We call these conditions the learning constraints.

Structured Prediction: Learning

- Learning: given a set of structured examples $\{(x,y)\}$
  find a scoring function w that minimizes empirical loss.
- Learning is thus driven by the attempt to find a weight vector w such that for each given annotated example (x_i, y_i):

$$\forall y \quad \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) \geq \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}_i)$$

- We call these conditions the learning constraints.

- In most learning algorithms used today, the update of the weight vector w is done in an on-line fashion,
  - Think about it as Perceptron; this procedure applies to Structured Perceptron, CRFs, Linear Structured SVM

Structured Prediction: Learning

- Learning: given a set of structured examples {(x,y)}
  find a scoring function w that minimizes empirical loss.
- Learning is thus driven by the attempt to find a weight vector w such that for each given annotated example (x_i, y_i):

$$\forall y \quad \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) \geq \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}_i)$$

- We call these conditions the learning constraints.

- In most learning algorithms used today, the update of the weight vector w is done in an on-line fashion,
  - Think about it as Perceptron; this procedure applies to Structured Perceptron, CRFs, Linear Structured SVM
- W.l.o.g. (almost) we can thus write the generic structured learning algorithm as follows:

Structured Prediction: Learning Algorithm

- For each example (x_i, y_i)
- Do: (with the current weight vector w)
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = \text{argmax}_{y \in \mathcal{Y}}\, w^T \phi(x_i, y)}$
  - **Check** the learning constraints
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndFor

Structured Prediction: Learning Algorithm

- For each example (x_i, y_i)
- Do: (with the current weight vector w)
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = \operatorname{argmax}_{y \in \mathcal{Y}} w^T \phi(x_i, y)}$
  - **Check** the learning constraints
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndFor

Structured Prediction: Learning Algorithm

- For each example (x_i, y_i)
- Do: (with the current weight vector w)
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = argmax_{y \in \mathcal{Y}}\ w^T \phi(x_i, y)}$
  - **Check** the learning constraints
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndFor

Structured Prediction: Learning Algorithm

- For each example (x_i, y_i)
- Do: (with the current weight vector w)
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = \text{argmax}_{y \in \mathcal{Y}}\, w^T \phi(x_i, y)}$
  - **Check** the learning constraints
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndFor

Structured Prediction: Learning Algorithm

In the structured case, the prediction (inference) step is often intractable and needs to be done many times

- For each example (x_i, y_i)
- Do: (with the current weight vector w)
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = argmax_{y \in \mathcal{Y}}\, w^T \phi(x_i, y)}$
  - **Check** the learning constraints
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndFor

Structured Prediction: Learning Algorithm

Solution I: decompose the scoring function to EASY and HARD parts

- For each example (x_i, y_i)
- Do:
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = \mathrm{argmax}_{y \in \mathcal{Y}}\ w_{EASY}^T \phi_{EASY}(x_i, y) + w_{HARD}^T \phi_{HARD}(x_i, y)}$
  - **Check** the learning constraint
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndDo

Structured Prediction: Learning Algorithm

Solution I: decompose the scoring function to EASY and HARD parts

- For each example (x_i, y_i)
- Do:
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = argmax_{y \in \mathcal{Y}}\ w_{EASY}^T\ \phi_{EASY}(x_i, y) + w_{HARD}^T\ \phi_{HARD}(x_i, y)}$
  - **Check** the learning constraint
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndDo

EASY: could be feature functions that correspond to an HMM, a linear CRF, or even $\phi_{EASY}(x,y) = \phi(x)$, omiting dependence on y, corresponding to classifiers. May not be enough if the HARD part is still part of each inference step.

Structured Prediction: Learning Algorithm

Solution II: Disregard some of the dependencies: **assume a simple model.**

- For each example (x_i, y_i)
- Do:
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = argmax_{y \in \mathcal{Y}}\ w_{EASY}^T \phi_{EASY}(x_i, y) + w_{HARD}^T \phi_{HARD}(x_i, y)}$
  - **Check** the learning constraint
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndDo

Structured Prediction: Learning Algorithm

Solution II: Disregard some of the dependencies: **assume a simple model.**

- For each example (x_i, y_i)
- Do:
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = \text{argmax}_{y \in \mathcal{Y}}\ w_{EASY}^T \phi_{EASY}(x_i, y) + \cancel{w_{HARD}^T \phi_{HARD}(x_i, y)}}$
  - **Check** the learning constraint
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndDo

Structured Prediction: Learning Algorithm

- For each example (x_i, y_i)
- Do:
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = argmax_{y \in \mathcal{Y}}\ w_{EASY}^T \phi_{EASY}(x_i, y) + w_{HARD}^T \phi_{HARD}(x_i, y)}$
  - **Check** the learning constraint
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndDo

Structured Prediction: Learning Algorithm

- For each example (x_i, y_i)
- Do:
  - Predict: perform Inference with the current weight vector
    - $y_i' = \mathrm{argmax}_{y \in \mathcal{Y}}\ w_{EASY}^T \phi_{EASY}(x_i, y) + \cancel{w_{HARD}^T \phi_{HARD}(x_i, y)}$
  - **Check** the learning constraint
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndDo
- $y_i' = \mathrm{argmax}_{y \in \mathcal{Y}}\ w_{EASY}^T \phi_{EASY}(x_i, y) + w_{HARD}^T \phi_{HARD}(x_i, y)$

Structured Prediction: Learning Algorithm

Solution III: Disregard some of the dependencies during learning; take into account at decision time

- For each example (x_i, y_i)
- Do:
  - Predict: perform Inference with the current weight vector
    - $\mathbf{y_i' = argmax_{y \in \mathcal{Y}}\ w_{EASY}^T \phi_{EASY}(x_i, y)} + \cancel{\mathbf{w_{HARD}^T \phi_{HARD}(x_i, y)}}$
  - **Check** the learning constraint
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndDo
- $\mathbf{y_i' = argmax_{y \in \mathcal{Y}}\ w_{EASY}^T \phi_{EASY}(x_i, y) + w_{HARD}^T \phi_{HARD}(x_i, y)}$

Structured Prediction: Learning Algorithm

Solution III: Disregard some of the dependencies during learning; take into account at decision time

- For each example (x_i, y_i)
- Do:
  - Predict: perform Inference with the current weight vector
    - $y_i' = \text{argmax}_{y \in \mathcal{Y}} \; w_{EASY}^T \phi_{EASY}(x_i, y) + \cancel{w_{HARD}^T \phi_{HARD}(x_i, y)}$
  - **Check** the learning constraint
    - **Is the score of the current prediction better than of (x_i, y_i)?**
  - If **Yes** – a mistaken prediction
    - **Update w**
  - Otherwise: no need to update w on this example
- EndDo
- $y_i' = \text{argmax}_{y \in \mathcal{Y}} \; w_{EASY}^T \phi_{EASY}(x_i, y) + w_{HARD}^T \phi_{HARD}(x_i, y)$

This is the most commonly used solution in NLP today

Constrained Conditional Models

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Penalty for violating the constraint.

(Soft) constraints component

Weight Vector for "local" models

Features, classifiers; log-linear models (HMM, CRF) or a combination

How far y is from a "legal" assignment

How to solve?

This is an Integer Linear Program

Solving using ILP packages gives an exact solution.

Cutting Planes, Dual Decomposition & other search techniques are possible

How to train?

Training is learning the objective function

Decouple? Decompose?

How to exploit the structure to minimize supervision?

Constrained Conditional Models

Any MAP problem w.r.t. any probabilistic model, can be formulated as an ILP [Roth+ 04, Taskar 04]

$$\operatorname*{argmax}_{y} \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Penalty for violating the constraint.

(Soft) constraints component

Weight Vector for "local" models

Features, classifiers; log-linear models (HMM, CRF) or a combination

How far y is from a "legal" assignment

How to solve?

This is an Integer Linear Program

Solving using ILP packages gives an exact solution.

Cutting Planes, Dual Decomposition & other search techniques are possible

How to train?

Training is learning the objective function

Decouple? Decompose?

How to exploit the structure to minimize supervision?

Examples: CCM Formulations

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Examples: CCM Formulations

$$\operatorname*{argmax}_{y} \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

CCMs can be viewed as a general interface to easily combine declarative domain knowledge with data driven statistical models

Examples: CCM Formulations

$$\underset{y}{\text{argmax}}\, \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

CCMs can be viewed as a general interface to easily combine declarative domain knowledge with data driven statistical models

Formulate NLP Problems as ILP problems (inference may be done otherwise)

1. Sequence tagging (HMM/CRF + Global constraints)
2. Sentence Compression (Language Model + Global Constraints)
3. SRL (Independent classifiers + Global Constraints)

Examples: CCM Formulations

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

CCMs can be viewed as a general interface to easily combine declarative domain knowledge with data driven statistical models

Formulate NLP Problems as ILP problems (inference may be done otherwise)

→ 1. Sequence tagging (HMM/CRF + Global constraints)

2. Sentence Compression (Language Model + Global Constraints)

3. SRL (Independent classifiers + Global Constraints)

Sequential Prediction

HMM/CRF based:

Argmax $\sum \lambda_{ij} x_{ij}$

Linguistics Constraints

Cannot have both A states and B states in an output sequence.

Examples: CCM Formulations

$$\underset{y}{\operatorname{argmax}}\ \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

CCMs can be viewed as a general interface to easily combine declarative domain knowledge with data driven statistical models

Formulate NLP Problems as ILP problems (inference may be done otherwise)

1. Sequence tagging (HMM/CRF + Global constraints)
2. Sentence Compression (Language Model + Global Constraints)
3. SRL (Independent classifiers + Global Constraints)

Sentence Compression/Summarization:

Language Model based:

Argmax $\sum \lambda_{ijk} x_{ijk}$

Linguistics Constraints

If a modifier chosen, include its head
If verb is chosen, include its arguments

Examples: CCM Formulations

$$\underset{y}{\operatorname{argmax}} \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

CCMs can be viewed as a general interface to easily combine declarative domain knowledge with data driven statistical models

Formulate NLP Problems as ILP problems (inference may be done otherwise)

1. Sequence tagging (HMM/CRF + Global constraints)
2. Sentence Compression (Language Model + Global Constraints)
3. SRL (Independent classifiers + Global Constraints)

Sentence Compression/Summarization:

Language Model based:

Argmax $\sum \lambda_{ijk} x_{ijk}$

Linguistics Constraints

If a modifier chosen, include its head
If verb is chosen, include its arguments

Examples: CCM Formulations

$$\operatorname*{argmax}_y \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

CCMs can be viewed as a general interface to easily combine declarative domain knowledge with data driven statistical models

Formulate NLP Problems as ILP problems (inference may be done otherwise)

1. Sequence tagging (HMM/CRF + Global constraints)
2. Sentence Compression (Language Model + Global Constraints)
3. SRL (Independent classifiers + Global Constraints)

Constrained Conditional Models Allow:

- Learning a simple model (or multiple; or pipelines)
- Make decisions with a more complex model
- Accomplished by directly incorporating constraints to bias/re-rank global decisions composed of simpler models' decisions
- More sophisticated algorithmic approaches exist to bias the output
  [CoDL: Cheng et. al 07,12; PR: Ganchev et. al. 10; DecL, UEM: Samdani et. al 12]

Semantic Role Labeling (SRL)

I *left* my pearls to my daughter in my will .

[I]$_{A0}$ *left* [my pearls]$_{A1}$ [to my daughter]$_{A2}$ [in my will]$_{AM\text{-}LOC}$.

- ***A0*** Leaver
- ***A1*** Things left
- ***A2*** Benefactor
- ***AM-LOC*** Location

I *left* my pearls to my daughter in my will .

Semantic Role Labeling (SRL)

Archetypical Information Extraction Problem: E.g., Concept Identification and Typing, Event Identification, etc.

I *left* my pearls to my daughter in my will .

[I]$_{A0}$ *left* [my pearls]$_{A1}$ [to my daughter]$_{A2}$ [in my will]$_{AM\text{-}LOC}$.

- *A0* Leaver
- *A1* Things left
- *A2* Benefactor
- *AM-LOC* Location

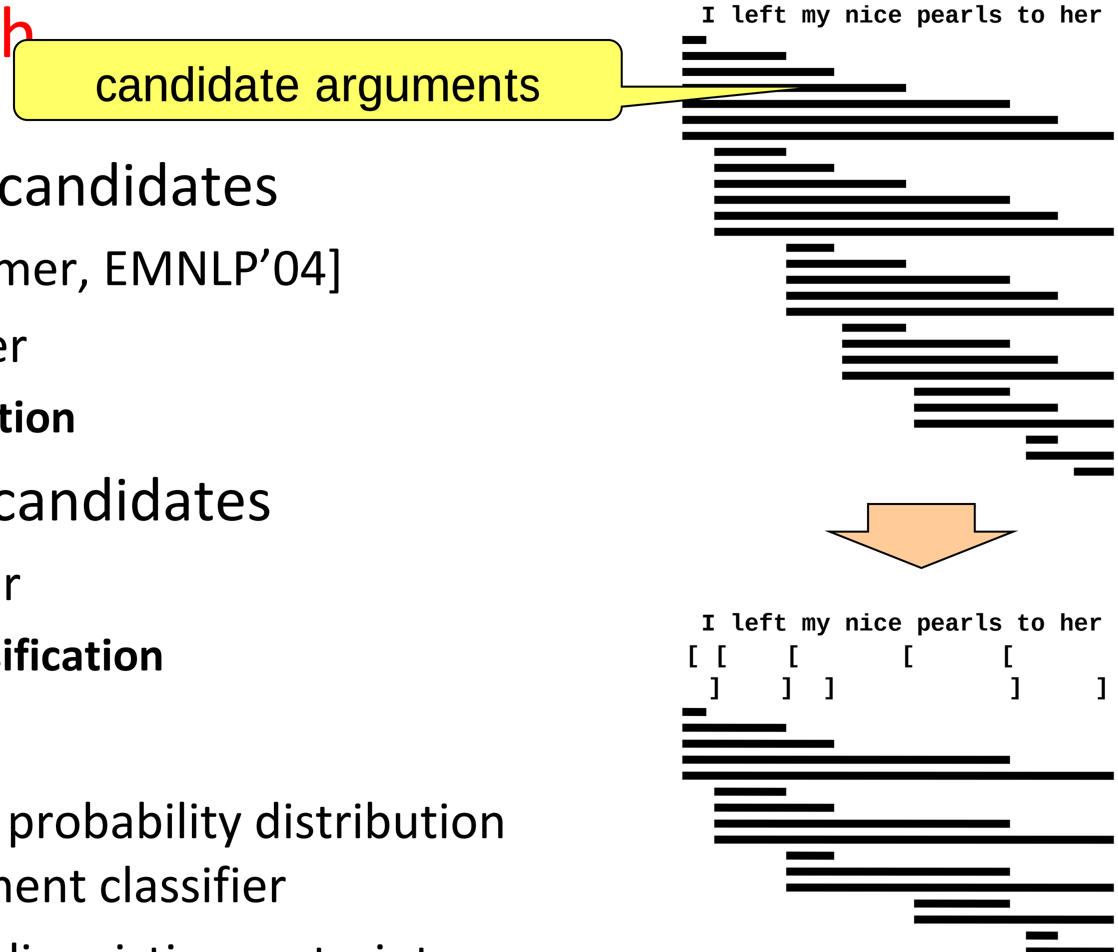
I *left* my pearls to my daughter in my will .

Algorithmic Approach

- Identify argument candidates
  - Pruning [Xue&Palmer, EMNLP'04]
  - Argument Identifier
    - **Binary classification**
- Classify argument candidates
  - Argument Classifier
    - **Multi-class classification**
- Inference
  - Use the estimated probability distribution given by the argument classifier
  - Use structural and linguistic constraints
  - Infer the optimal global output

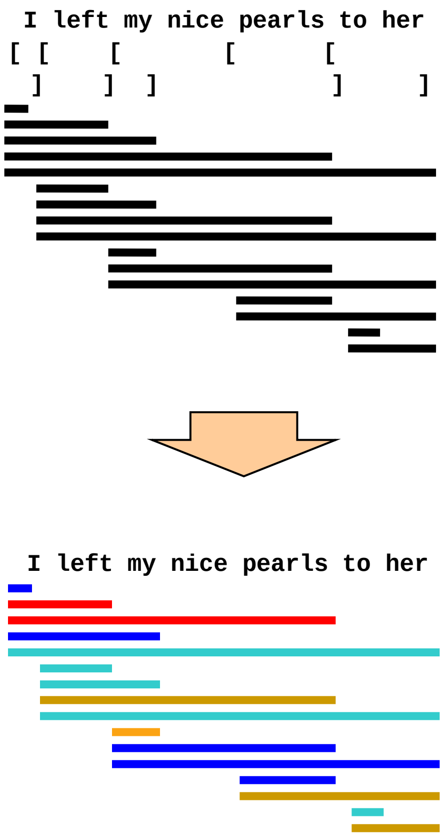
Algorithmic Approach

- Identify argument candidates
  - Pruning [Xue&Palmer, EMNLP'04]
  - Argument Identifier
    - **Binary classification**
- Classify argument candidates
  - Argument Classifier
    - **Multi-class classification**
- Inference
  - Use the estimated probability distribution given by the argument classifier
  - Use structural and linguistic constraints
  - Infer the optimal global output



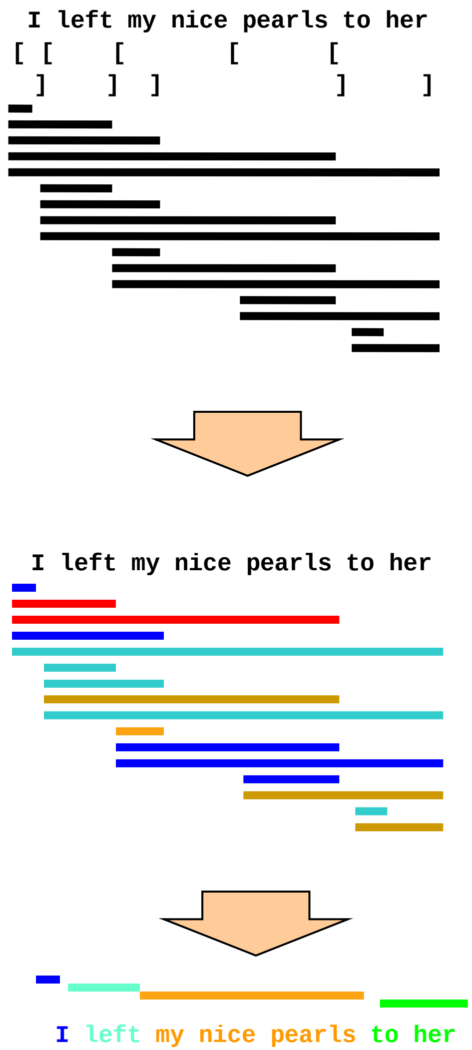
Algorithmic Approach

- Identify argument candidates
  - Pruning [Xue&Palmer, EMNLP'04]
  - Argument Identifier
    - **Binary classification**
- Classify argument candidates
  - Argument Classifier
    - **Multi-class classification**
- Inference
  - Use the estimated probability distribution given by the argument classifier
  - Use structural and linguistic constraints
  - Infer the optimal global output



Algorithmic Approach

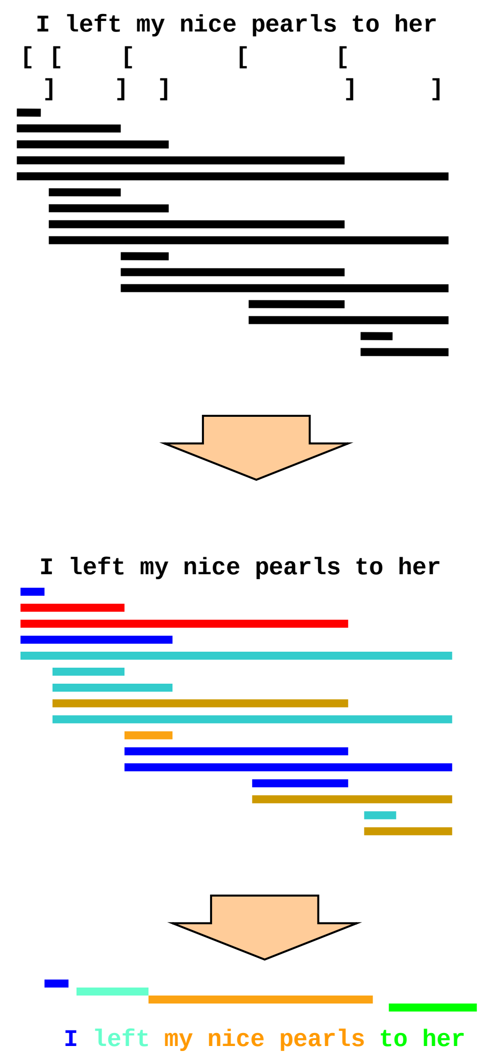
- **Identify** argument candidates
  - Pruning [Xue&Palmer, EMNLP'04]
  - Argument Identifier
    - **Binary classification**
- **Classify** argument candidates
  - Argument Classifier
    - **Multi-class classification**
- **Inference**
  - Use the estimated probability distribution given by the argument classifier
  - Use structural and linguistic constraints
  - Infer the optimal global output



Algorithmic Approach

- **Identify** argument candidates
  - Pruning [Xue&Palmer, EMNLP'04]
  - Argument Identifier
    - **Binary classification**
- **Classify** argument candidates
  - Argument Classifier
    - **Multi-class classification**
- **Inference**
  - Use the estimated probability distribution given by the argument classifier
  - Use st ic constraints
  - Infer t utput

One inference problem for each verb predicate.



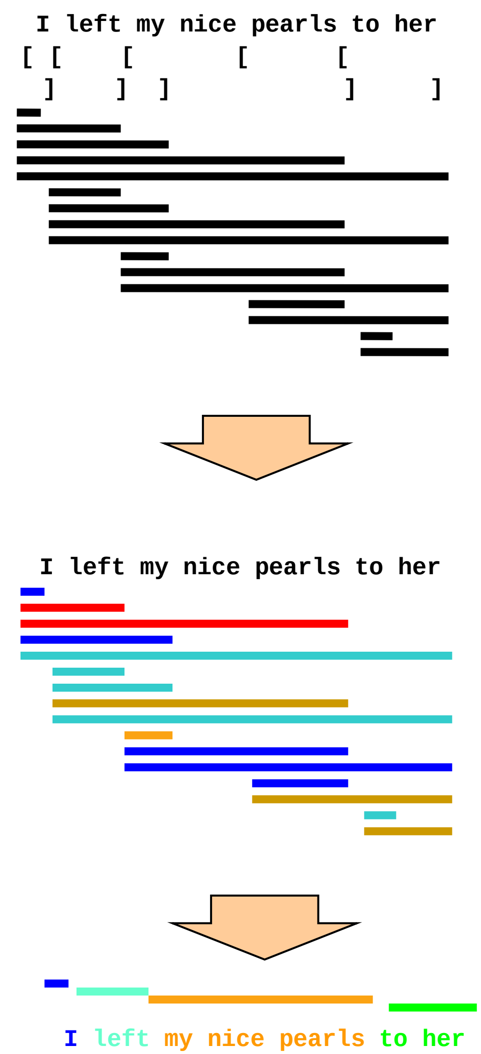
Algorithmic Approach

- Identify argument candidates
  - Pruning [Xue&Palmer, EMNLP'04]
  - Argument Identifier
    - **Binary classification**
- Classify argument candidates
  - Argument Classifier
    - **Multi-class classification**
- Inference

argmax $\sum_{a,t} y^{a,t} c^{a,t} = \sum_{a,t} 1_{a=t} c_{a=t}$

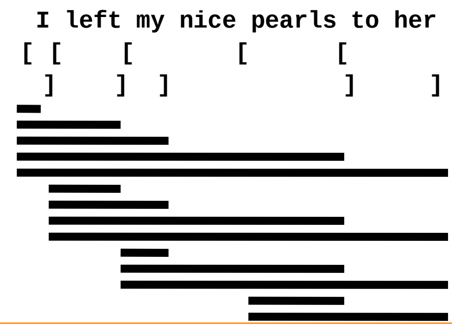
Subject to:

- One label per argument: $\sum_{t} y^{a,t} = 1$
- No overlapping or embedding
- Relations between verbs and arguments,….

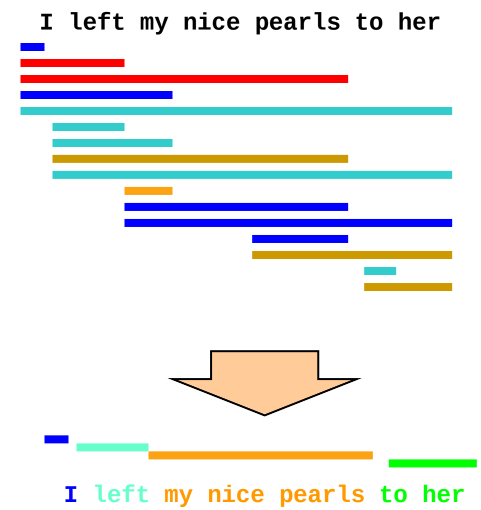


Algorithmic Approach

- **Identify** argument candidates
  - Pruning [Xue&Palmer, EMNLP'04]
  - Argument Identifier
    - **Binary classification**
- **Classify** argument candidates
  - Argument Classifier
    - **Multi-class classification**
- **Inference**



Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score



$$\text{argmax} \sum_{a,t} y^{a,t} c^{a,t} = \sum_{a,t} 1_{a=t} c_{a=t}$$

Subject to:

- One label per argument: $\sum_t y^{a,t} = 1$
- No overlapping or embedding
- Relations between verbs and arguments,….

Algorithmic Approach

- Identify argument candid
  - Pruning [Xue&Palmer, EM
  - Argument Identifier
    - **Binary classification**
- Classify argument candida
  - Argument Classifier
    - **Multi-class classification**
- Inference

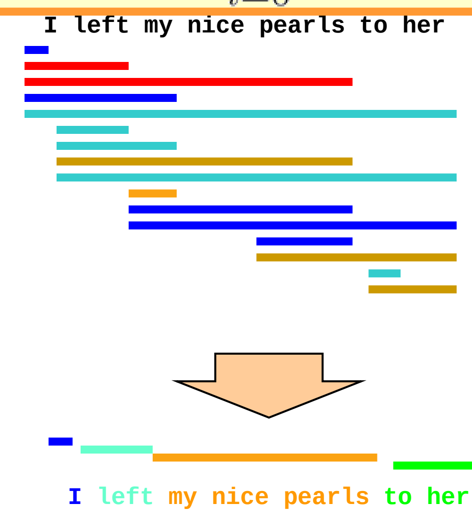
No duplicate argument classes $\forall i, \sum_{y\in\mathcal{Y}} 1_{\{y_i=y\}} = 1$

Unique labels

$$\forall y \in \mathcal{Y}, \sum_{i=0}^{n-1} 1_{\{y_i=y\}} \le 1$$

$$\forall y \in \mathcal{Y}_R, \sum_{i=0}^{n-1} 1_{\{y_i=y=\text{``R-Ax''}\}} \le \sum_{i=0}^{n-1} 1_{\{y_i=\text{``Ax''}\}}$$

$$\forall j, y \in \mathcal{Y}_C, 1_{\{y_j=y=\text{``C-Ax''}\}} \le \sum_{i=0}^{j} 1_{\{y_i=\text{``Ax''}\}}$$



argmax $\sum_{a,t} y^{a,t} c^{a,t} = \sum_{a,t} 1_{a=t} c_{a=t}$

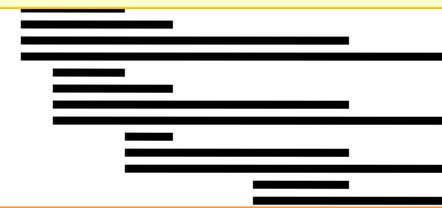
Subject to:

- One label per argument: $\sum_t y^{a,t} = 1$
- No overlapping or embedding
- Relations between verbs and arguments,....

Algorithmic Approach

- Identify argument candidates
  - Pruning [Xue&Palmer, EMNLP'04]
  - Argument Identifier
    - **Binary classification**
- Classify argument candidates
  - Argument Classifier
    - **Multi-class classification**
- Inference

Learning Based Java: allows a developer to encode constraints in First Order Logic; these are compiled into linear inequalities automatically.

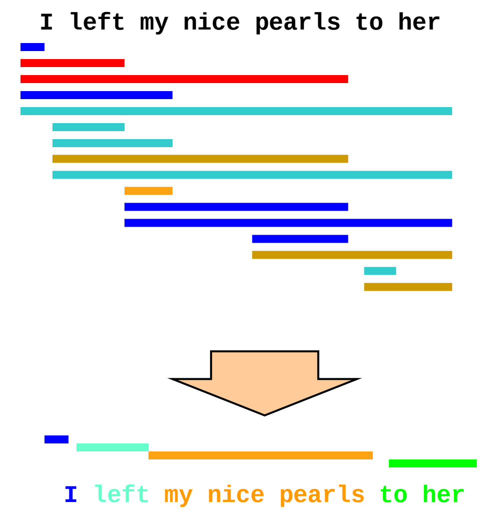


Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

$$\text{argmax} \sum_{a,t} y^{a,t} c^{a,t} = \sum_{a,t} 1_{a=t}\, c_{a=t}$$

Subject to:

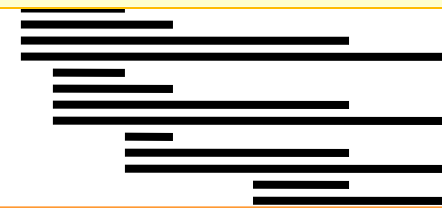
- One label per argument: $\sum_t y^{a,t} = 1$
- No overlapping or embedding
- Relations between verbs and arguments,....



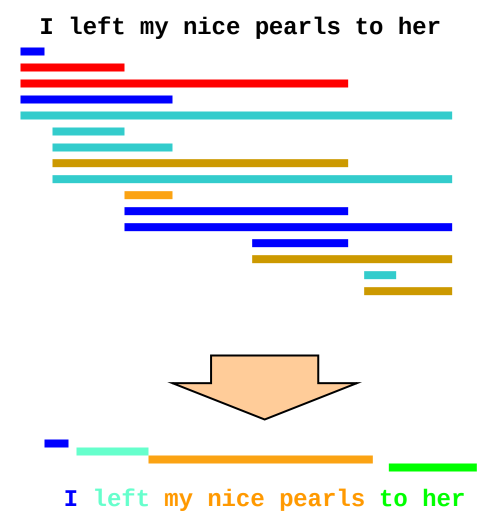
Algorithmic Approach

- Identify argument candidates
  - Pruning [Xue&Palmer, EMNLP'04]
  - Argument Identifier
    - **Binary classification**
- Classify argument candidates
  - Argument Classifier
    - **Multi-class classification**
- Inference

Learning Based Java: allows a developer to encode constraints in First Order Logic; these are compiled into linear inequalities automatically.



Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score



$$\text{argmax} \sum_{a,t} y^{a,t} c^{a,t} = \sum_{a,t} 1_{a=t} c_{a=t}$$

Subject to:

- One label per argument: $\sum_t y^{a,t} = 1$
- No overlapping or embedding
- Relations between verbs and arguments,….

Use the **pipeline architecture's simplicity** while **maintaining uncertainty**: keep probability distributions over decisions & use global inference at decision time.

SRL: Posing the Problem

$$\boxed{\text{maximize}} \sum_{i=0}^{n-1} \sum_{y \in \mathcal{Y}} \lambda_{\mathbf{x}_i, y} 1_{\{y_i = y\}}$$

$$\text{where} \quad \lambda_{\mathbf{x},y} = \lambda \cdot F(\mathbf{x}, y) = \lambda_y \cdot F(\mathbf{x})$$

$$\boxed{\text{subject to}}$$

A	bomb [A1]	killer [A0]
car		
bomb		
that	bomb (Reference) [R-A1]	
exploded	V: explode	
outside	location [AM-LOC]	
the		
U.S.		
military	temporal [AM-TMP]	
base		
in	location [AM-LOC]	
Beniji		
killed		V: kill
11		corpse [A1]
Iraqi		
citizens		
.		

Demo:
http://cogcomp.cs.illinois.edu/

SRL: Posing the Problem

$$\boxed{\text{maximize}} \quad \sum_{i=0}^{n-1} \sum_{y \in \mathcal{Y}} \lambda_{\mathbf{x}_i, y} 1_{\{y_i = y\}}$$

$$\text{where} \quad \lambda_{\mathbf{x},y} = \lambda \cdot F(\mathbf{x}, y) = \lambda_y \cdot F(\mathbf{x})$$

$$\boxed{\text{subject to}}$$

$$\forall i, \sum_{y \in \mathcal{Y}} 1_{\{y_i = y\}} = 1$$

$$\forall y \in \mathcal{Y}, \sum_{i=0}^{n-1} 1_{\{y_i = y\}} \leq 1$$

$$\forall y \in \mathcal{Y}_R, \sum_{i=0}^{n-1} 1_{\{y_i = y = \text{“R-Ax”}\}} \leq \sum_{i=0}^{n-1} 1_{\{y_i = \text{“Ax”}\}}$$

$$\forall j, y \in \mathcal{Y}_C, \ 1_{\{y_j = y = \text{“C-Ax”}\}} \leq \sum_{i=0}^{j} 1_{\{y_i = \text{“Ax”}\}}$$

	⊟	⊟
A	bomb [A1]	killer [A0]
car		
bomb		
that	bomb (Reference) [R-A1]	
exploded	V: explode	
outside	location [AM-LOC]	
the		
U.S.		
military	temporal [AM-TMP]	
base		
in	location [AM-LOC]	
Beniji		
killed		V: kill
11		corpse [A1]
Iraqi		
citizens		
.		

Demo:
http://cogcomp.cs.illinois.edu/

SRL: Posing the Problem

$$\boxed{\text{maximize}} \sum_{i=0}^{n-1} \sum_{y \in \mathcal{Y}} \lambda_{\mathbf{x}_i, y} 1_{\{y_i = y\}}$$

$$\text{where} \quad \lambda_{\mathbf{x},y} = \lambda \cdot F(\mathbf{x}, y) = \lambda_y \cdot F(\mathbf{x})$$

$$\boxed{\text{subject to}}$$

$$\forall i, \sum_{y \in \mathcal{Y}} 1_{\{y_i = y\}} = 1$$

$$\forall y \in \mathcal{Y}, \sum_{i=0}^{n-1} 1_{\{y_i = y\}} \leq 1$$

$$\forall y \in \mathcal{Y}_R, \sum_{i=0}^{n-1} 1_{\{y_i = y = \text{“R-Ax”}\}} \leq \sum_{i=0}^{n-1} 1_{\{y_i = \text{“Ax”}\}}$$

$$\forall j, y \in \mathcal{Y}_C, \ 1_{\{y_j = y = \text{“C-Ax”}\}} \leq \sum_{i=0}^{j} 1_{\{y_i = \text{“Ax”}\}}$$

A	bomb [A1]	killer [A0]
car		
bomb		
that	bomb (Reference) [R-A1]	
exploded	V: explode	
outside	location [AM-LOC]	
the		
U.S.		
military	temporal [AM-TMP]	
base		
in	location [AM-LOC]	
Benjii		
killed		V: kill
11		corpse [A1]
Iraqi		
citizens		
.		

Demo: http://cogcomp.cs.illinois.edu/

If there is an Reference-Ax phrase, there is an Ax

If there is an Continuation-x phrase, there is an Ax before it

SRL: Posing the Problem

In this case, independent learners

$$\boxed{\text{maximize}} \quad \sum_{i=0}^{n-1} \sum_{y \in \mathcal{Y}} \lambda_{\mathbf{x}_i, y} 1_{\{y_i = y\}}$$

$$\text{where} \quad \lambda_{\mathbf{x},y} = \lambda \cdot F(\mathbf{x}, y) = \lambda_y \cdot F(\mathbf{x})$$

$$\boxed{\text{subject to}}$$

$$\forall i, \ \sum_{y \in \mathcal{Y}} 1_{\{y_i = y\}} = 1$$

$$\forall y \in \mathcal{Y}, \ \sum_{i=0}^{n-1} 1_{\{y_i = y\}} \leq 1$$

$$\forall y \in \mathcal{Y}_R, \ \sum_{i=0}^{n-1} 1_{\{y_i = y = \text{“R-Ax”}\}} \leq \sum_{i=0}^{n-1} 1_{\{y_i = \text{“Ax”}\}}$$

$$\forall j, y \in \mathcal{Y}_C, \ 1_{\{y_j = y = \text{“C-Ax”}\}} \leq \sum_{i=0}^{j} 1_{\{y_i = \text{“Ax”}\}}$$

	⊟	⊟
A	bomb [A1]	killer [A0]
car		
bomb		
that	bomb (Reference) [R-A1]	
exploded	V: explode	
outside	location [AM-LOC]	
the		
U.S.		
military	temporal [AM-TMP]	
base		
in	location [AM-LOC]	
Benjji		
killed		V: kill
11		corpse [A1]
Iraqi		
citizens		
.		

Demo: http://cogcomp.cs.illinois.edu/

If there is an Reference-Ax phrase, there is an Ax

If there is an Continuation-x phrase, there is an Ax before it

Verb SRL is not Sufficient

- *John, a fast-rising politician, slept on the train to Chicago.*
- **Verb Predicate: sleep**

Verb SRL is not Sufficient

- *John, a fast-rising politician, slept on the train to Chicago.*
- **Verb Predicate: sleep**
  - Sleeper: John, a fast-rising politician
  - Location: on the train to Chicago

Verb SRL is not Sufficient

- *John, a fast-rising politician, slept on the train to Chicago.*
- **Verb Predicate: sleep**
  - Sleeper: John, a fast-rising politician
  - Location: on the train to Chicago

- **Who was John?**

Verb SRL is not Sufficient

- *John, a fast-rising politician, slept on the train to Chicago.*
- **Verb Predicate: sleep**
  - Sleeper: John, a fast-rising politician
  - Location: on the train to Chicago
- **Who was John?**
  - Relation: Apposition (comma)
  - John, a fast-rising politician

Verb SRL is not Sufficient

- *John, a fast-rising politician, slept on the train to Chicago.*
- **Verb Predicate: sleep**
  - Sleeper: John, a fast-rising politician
  - Location: on the train to Chicago
- **Who was John?**
  - Relation: Apposition (comma)
  - John, a fast-rising politician
- **What was John's destination?**

Verb SRL is not Sufficient

- *John, a fast-rising politician, slept on the train to Chicago.*
- **Verb Predicate: sleep**
  - Sleeper: John, a fast-rising politician
  - Location: on the train to Chicago
- **Who was John?**
  - Relation: Apposition (comma)
  - John, a fast-rising politician
- **What was John's destination?**
  - Relation: Destination (preposition)
  - train to Chicago

Verb SRL is not Sufficient

- *John, a fast-rising politician, slept on the train to Chicago.*
- **Verb Predicate: sleep**
  - Sleeper: John, a fast-rising politician
  - Location: on the train to Chicago
- **Who was John?**
  - Relation: Apposition (comma)
  - John, a fast-rising politician
- **What was John's destination?**
  - Relation: Destination (preposition)
  - train to Chicago

Identify the relation expressed by the predicate, and its arguments

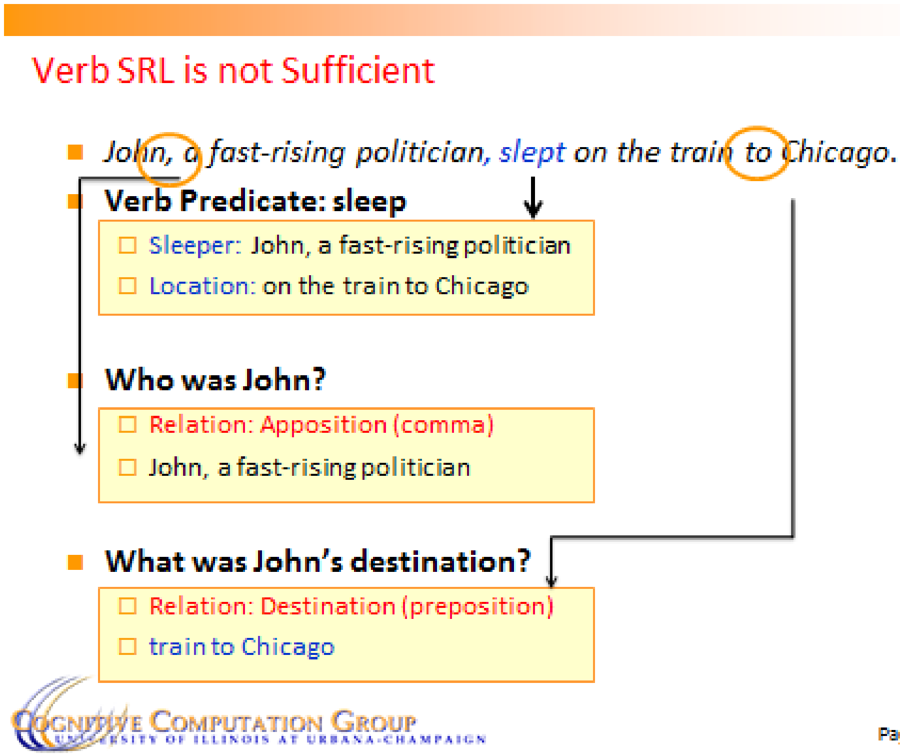
Verb SRL is not Sufficient

- *John, a fast-rising politician, slept on the train to Chicago.*
- **Verb Predicate: sleep**
  - Sleeper: John, a fast-rising politician
  - Location: on the train to Chicago
- **Who was John?**
  - Relation: Apposition (comma)
  - John, a fast-rising politician
- **What was John's destination?**
  - Relation: Destination (preposition)
  - train to Chicago

Identify the relation expressed by the predicate, and its arguments

Computational Challenges

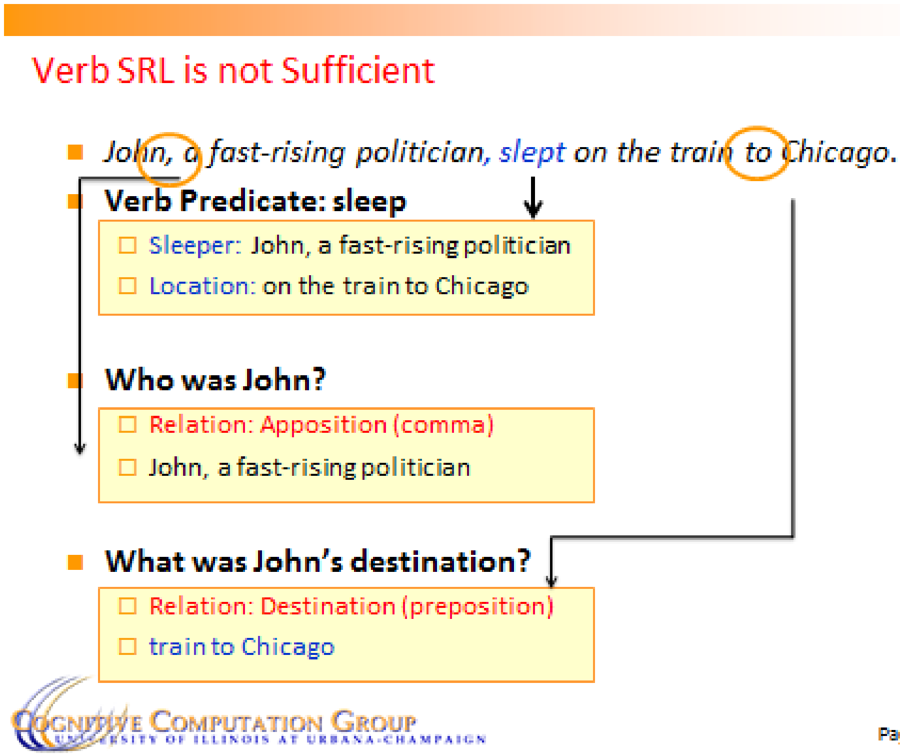
- Predict the preposition relations
  - [EMNLP, '11]
- Identify the relation's arguments
  - [Trans. Of ACL, '13]



Computational Challenges

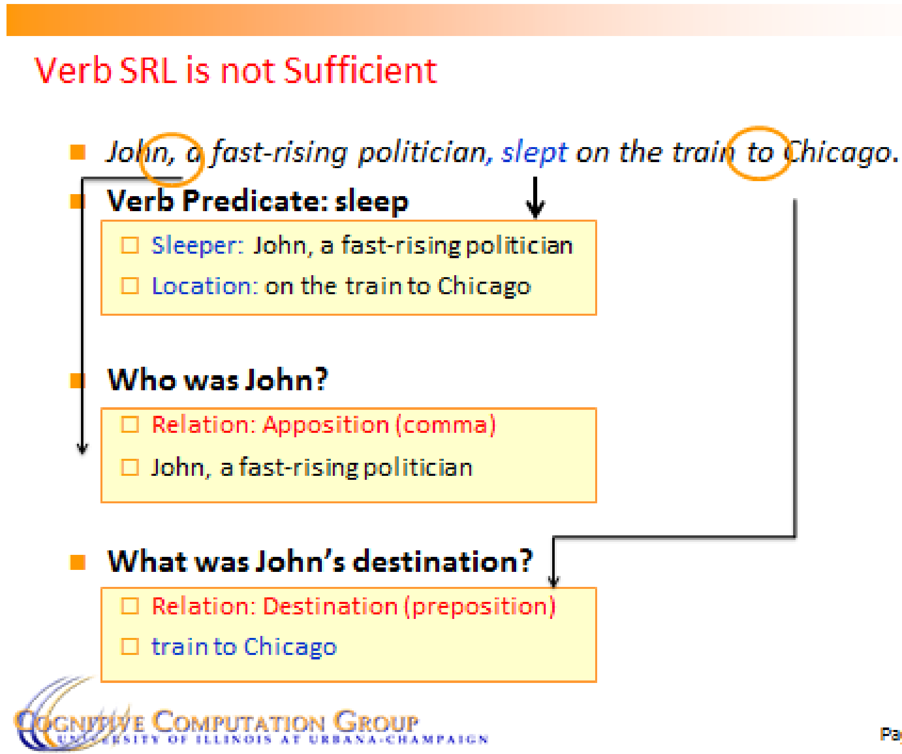
- Predict the preposition relations
  - [EMNLP, '11]
- Identify the relation's arguments
  - [Trans. Of ACL, '13]

- Very little supervised data
  - per phenomena
- Minimal annotation
  - only at the predicate level



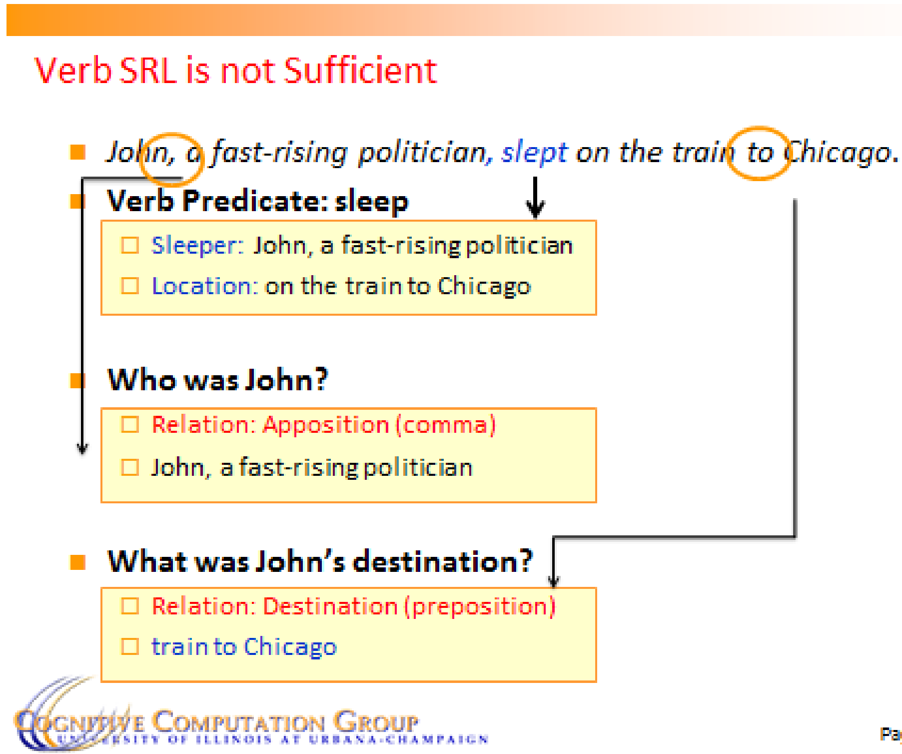
Computational Challenges

- Predict the preposition relations
  - [EMNLP, ’11]
- Identify the relation’s arguments
  - [Trans. Of ACL, ‘13]



- Very little supervised data
  - per phenomena
- Minimal annotation
  - only at the predicate level
- The Learning & Inference paradigm exploits two principles:
  - Coherency among multiple phenomena
  - Constraining latent structures (relating observed and latent variables)

Computational Challenges

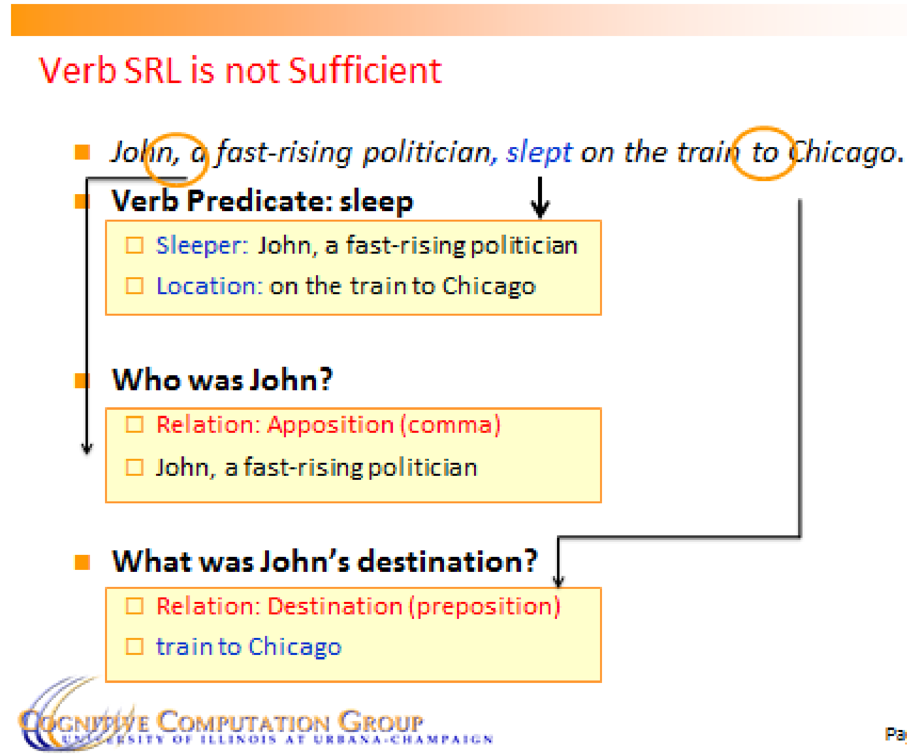


- Predict the preposition relations
  - [EMNLP, '11]
- Identify the relation's arguments
  - [Trans. Of ACL, '13]
- Very little supervised data
  - per phenomena
- Minimal annotation
  - only at the predicate level
- The Learning & Inference paradigm exploits two principles:
  - Coherency among multiple phenomena
  - Constraining latent structures (relating observed and latent variables)

Input & relation

Argument & their types

Computational Challenges



- Predict the preposition relations
  - [EMNLP, '11]
- Identify the relation's arguments
  - [Trans. Of ACL, '13]
- Very little supervised data
  - per phenomena
- Minimal annotation
  - only at the predicate level
- The Learning & Inference paradigm exploits two principles:
  - Coherency among multiple phenomena
  - Constraining latent structures (relating observed and latent variables)
  - Skip

Input & relation

Argument & their types

Extended Semantic Role labeling I

[EMNLP'12, TACL'13]

Verb Predicates, Noun predicates, prepositions, each dictates some relations, which have to cohere.

The bus was heading for Nairobi in Kenya.

Location

Destination

Extended Semantic Role labeling I

[EMNLP'12, TACL'13]

Verb Predicates, Noun predicates, prepositions, each dictates some relations, which have to cohere.

The bus was heading for Nairobi in Kenya.

Location

Destination

Predicate: head.02

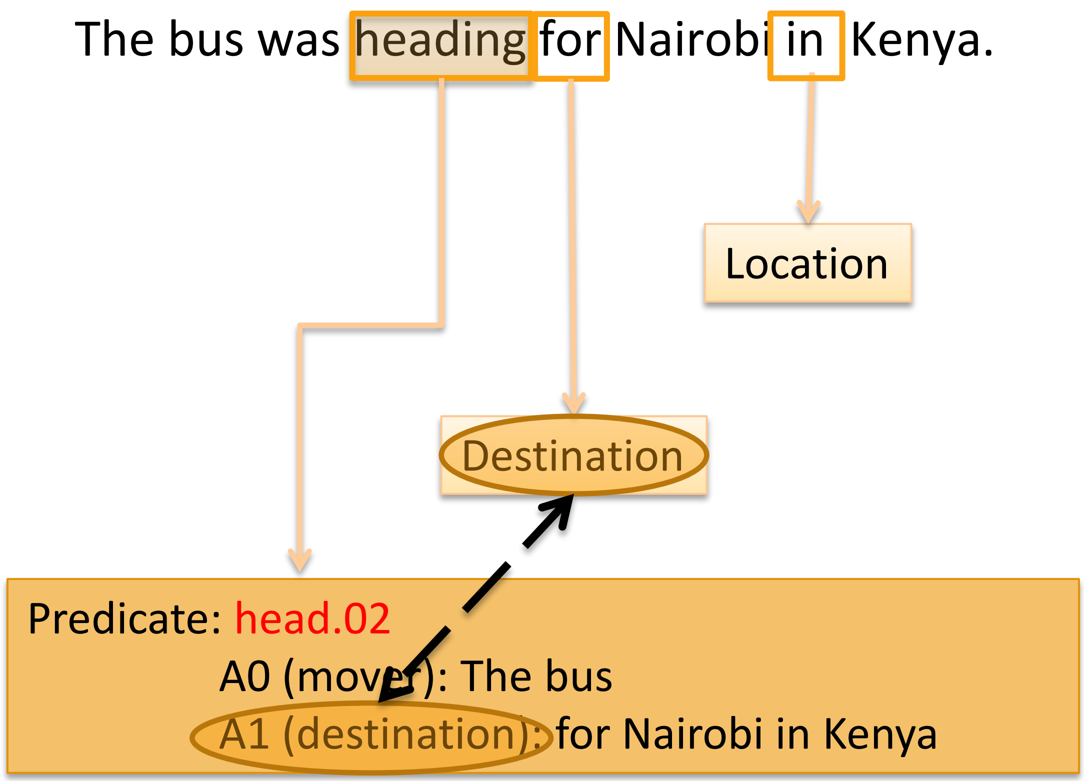
A0 (mover): The bus

A1 (destination): for Nairobi in Kenya

Cognitive Computation Group
University of Illinois at Urbana-Champaign

Extended Semantic Role labeling I
[EMNLP'12, TACL'13]

Verb Predicates, Noun predicates, prepositions, each dictates some relations, which have to cohere.

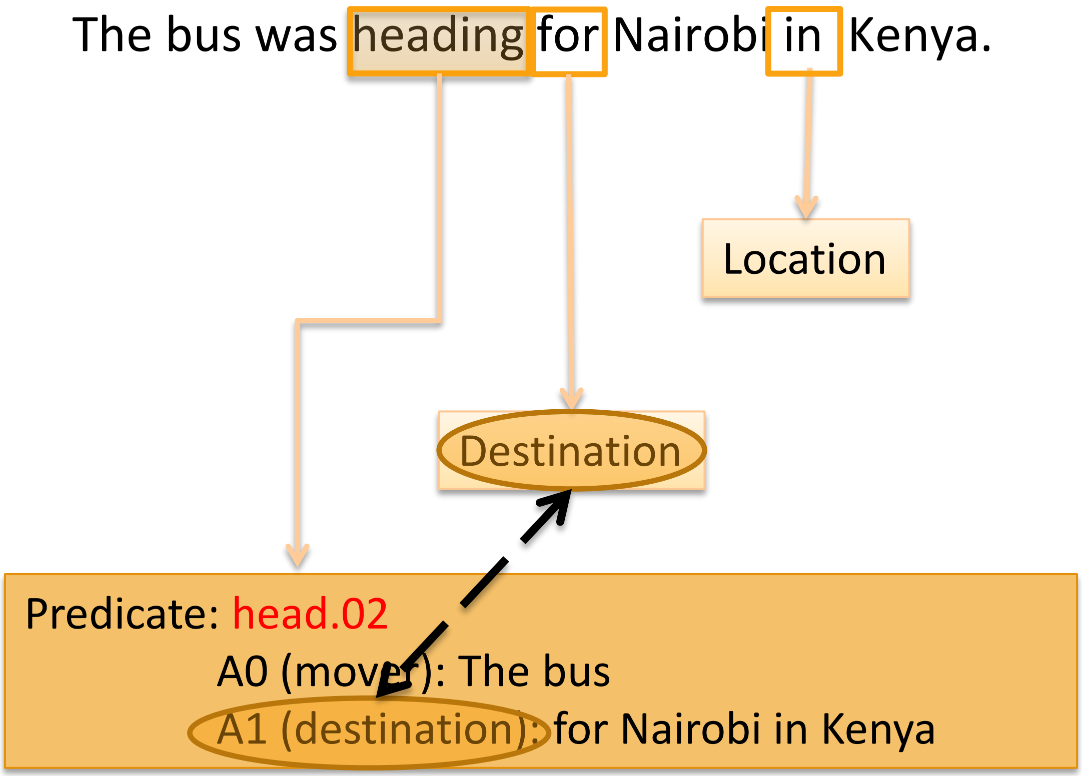


Extended Semantic Role labeling I

[EMNLP'12, TACL'13]

Verb Predicates, Noun predicates, prepositions, each dictates some relations, which have to cohere.

Predicate arguments from different triggers should be consistent



Cognitive Computation Group
University of Illinois at Urbana-Champaign

Extended Semantic Role labeling I

[EMNLP'12, TACL'13]

Verb Predicates, Noun predicates, prepositions, each dictates some relations, which have to cohere.

Predicate arguments from different triggers should be consistent

The bus was heading for Nairobi in Kenya.

Joint constraints

linking the two tasks.

Destination ⇔ A1

Location

Destination

Predicate: head.02

A0 (mover): The bus

A1 (destination): for Nairobi in Kenya

COGNITIVE COMPUTATION GROUP

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Joint inference (CCMs)

Verb arguments

$$\max_{\mathbf{y}} \sum_{t} \sum_{a} y^{a,t} c^{a,t}$$

Joint inference (CCMs)

Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

Verb arguments

$$\max_{\mathbf{y}} \sum_{t} \sum_{a} y^{a,t} c^{a,t}$$

Joint inference (CCMs)

Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

Verb arguments

$$\max_{\mathbf{y}} \sum_{t} \sum_{a} y^{a,t} c^{a,t}$$

Argument candidates

Each argument label

Joint inference (CCMs)

Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

$\boxed{\text{Verb arguments}}$

$$\max_{\mathbf{y}} \sum_{t} \sum_{a} y^{a,t} c^{a,t}$$

Constraints:

Verb SRL constraints

Joint inference (CCMs)

Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

Verb arguments

$$\max_{\mathbf{y}} \sum_t \sum_a y^{a,t} c^{a,t}$$

Preposition relations

$$\max_{\mathbf{y}} \sum_r \sum_p y^{r,p} c^{r,p}$$

Constraints:

Verb SRL constraints

Joint inference (CCMs)

Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

Verb arguments

$$\max_{\mathbf{y}} \sum_t \sum_a y^{a,t} c^{a,t}$$

Preposition relations

$$\max_{\mathbf{y}} \sum_{\boxed{r}} \sum_{\boxed{p}} y^{r,p} c^{r,p}$$

Preposition relation label

Preposition

Constraints:

Verb SRL constraints

Joint inference (CCMs)

Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

Verb arguments	Preposition relations
$\max_{\mathbf{y}} \sum_t \sum_a y^{a,t} c^{a,t}$	$\max_{\mathbf{y}} \sum_r \sum_p y^{r,p} c^{r,p}$

Constraints:

Verb SRL constraints	Preposition SRL Constraints

Joint inference (CCMs)

Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

Verb arguments

Preposition relations

$$\max_{\mathbf{y}} \sum_t \lambda^t \sum_a y^{a,t} c^{a,t} + \sum_r \lambda^r \sum_p y^{r,p} c^{r,p}$$

Constraints:

Verb SRL constraints

Preposition SRL Constraints

Joint inference (CCMs)

Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

Verb arguments

Preposition relations

$$\max_{\mathbf{y}} \sum_t \lambda^t \sum_a y^{a,t} c^{a,t} + \sum_r \lambda^r \sum_p y^{r,p} c^{r,p}$$

Constraints:

Verb SRL constraints

Preposition SRL Constraints

+ Joint constraints between tasks; easy with ILP formulations

Joint inference (CCMs)

Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

Verb arguments

Preposition relations

$$\max_{\mathbf{y}} \sum_t \lambda^t \sum_a y^{a,t} c^{a,t} + \sum_r \lambda^r \sum_p y^{r,p} c^{r,p}$$

Constraints:

Verb SRL constraints

Preposition SRL Constraints

+ Joint constraints between tasks; easy with ILP formulations

Joint Inference – no (or minimal) joint learning

Joint inference (CCMs)

Variable $y^{a,t}$ indicates whether candidate argument a is assigned a label t.
$c^{a,t}$ is the corresponding model score

Verb arguments

Preposition relations

$$\max_{\mathbf{y}} \sum_t \lambda^t \sum_a y^{a,t} c^{a,t} + \sum_r \lambda^r \sum_p y^{r,p} c^{r,p}$$

+ ....

Constraints:

Verb SRL constraints

Preposition SRL Constraints

+ Joint constraints between tasks; easy with ILP formulations

Joint Inference – no (or minimal) joint learning

ESRL II: Predicate-Argument Structure of Prepositions

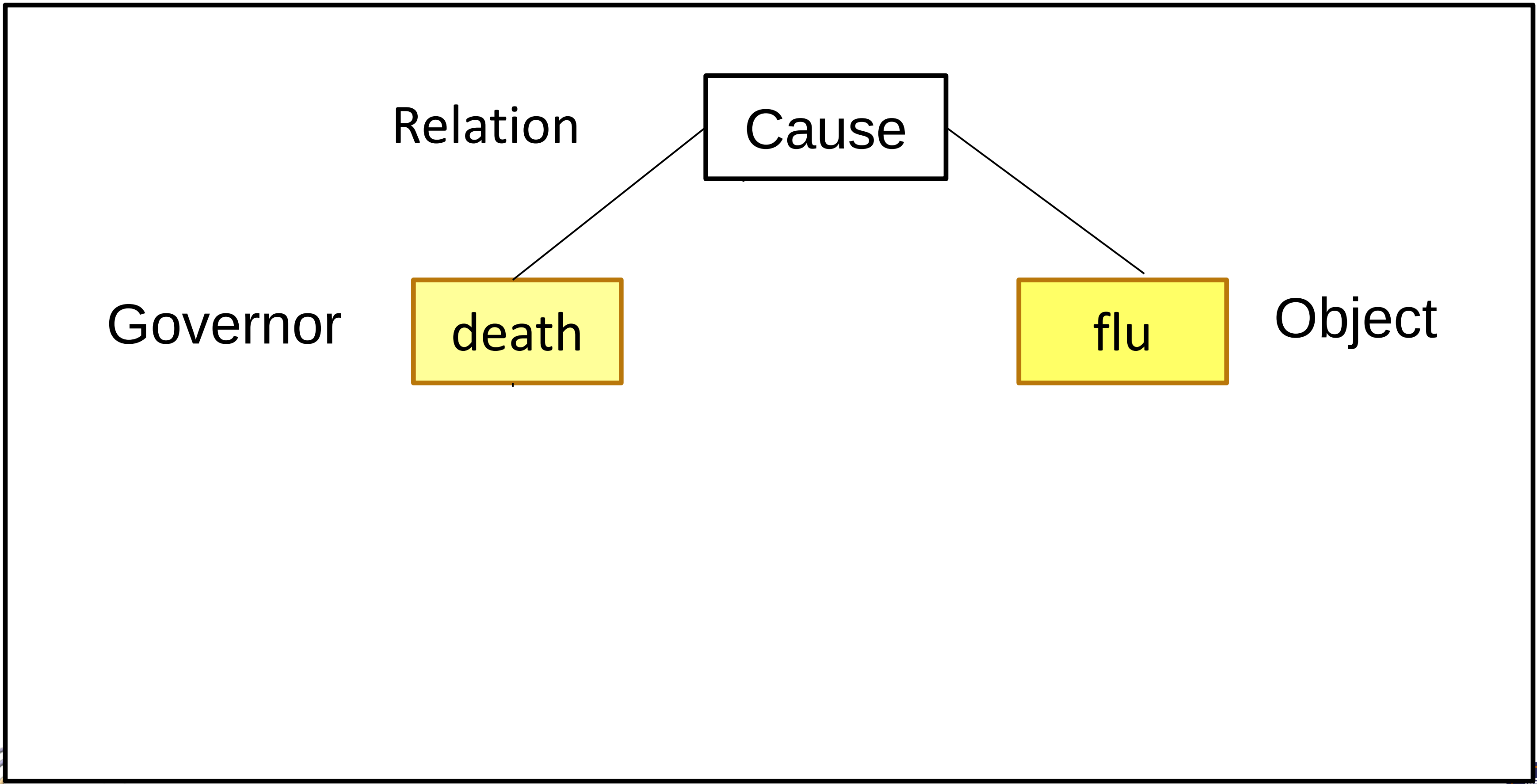
Poor care led to her death **from** flu.

ESRL II: Predicate-Argument Structure of Prepositions

Poor care led to her death **from** flu.

ESRL II: Predicate-Argument Structure of Prepositions

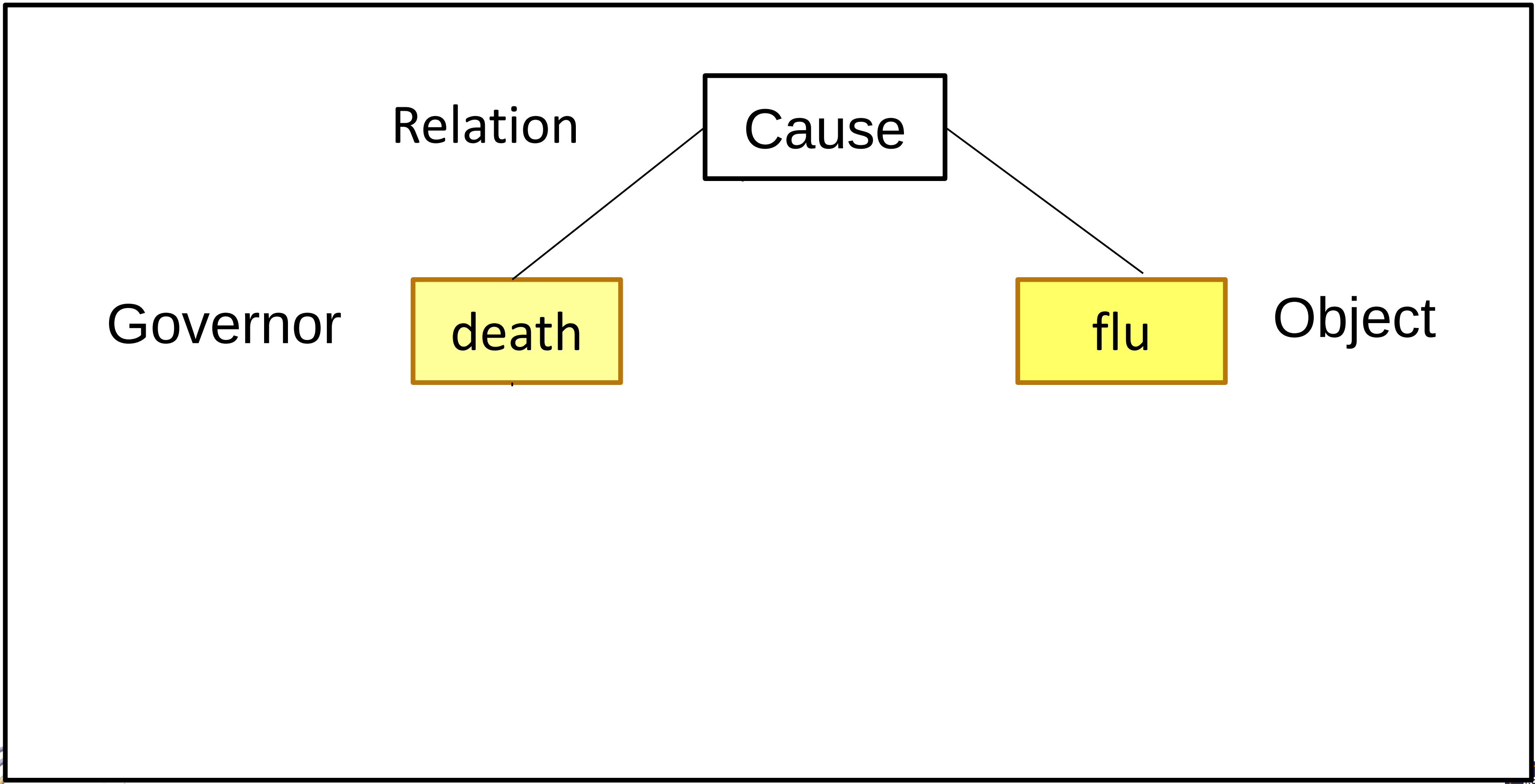
Poor care led to her death **from** flu.



ESRL II: Predicate-Argument Structure of Prepositions

……………her to suffer **from** infection.

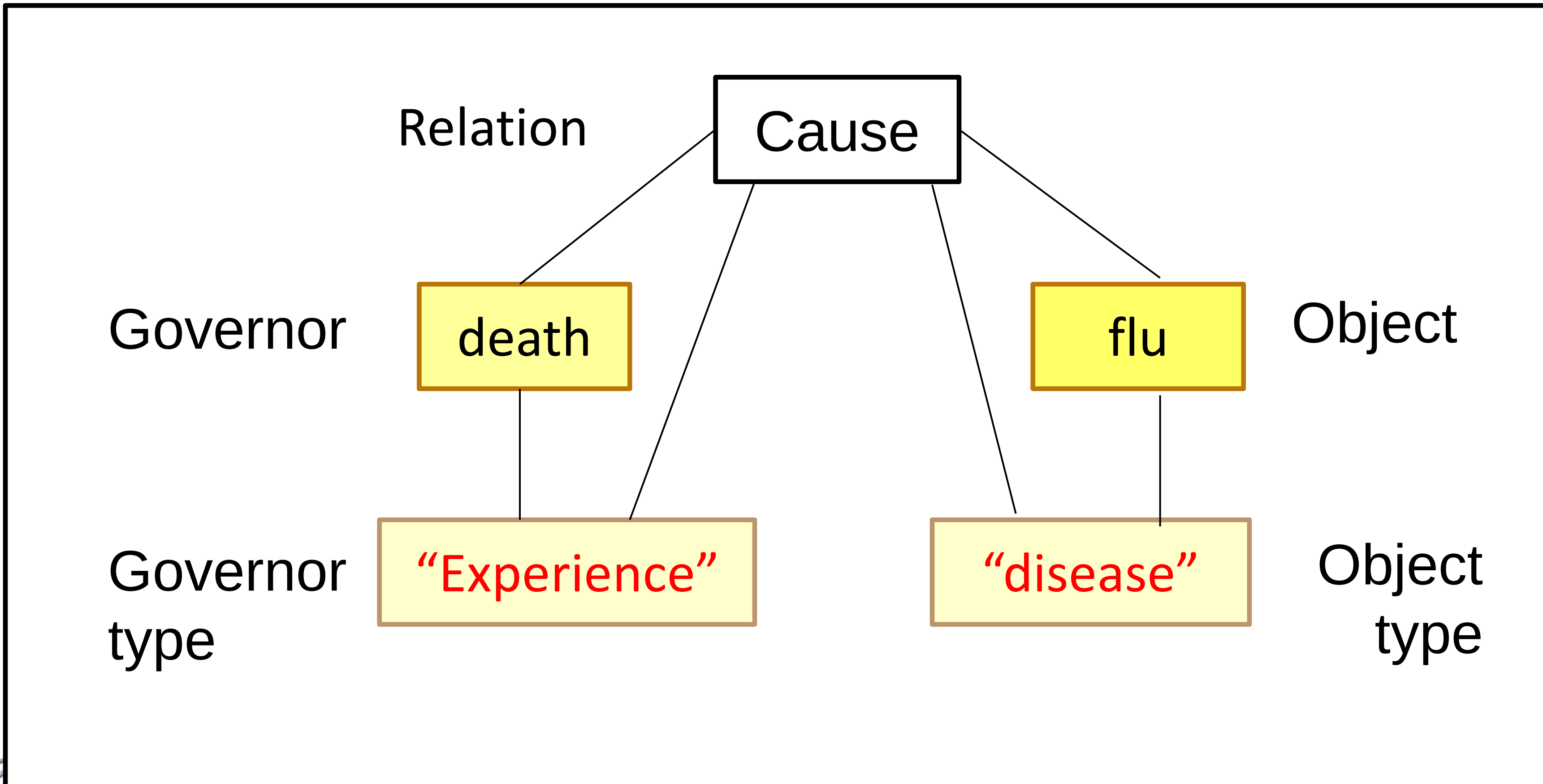
Poor care led to her death **from** flu.



ESRL II: Predicate-Argument Structure of Prepositions

……………her to suffer **from** infection.

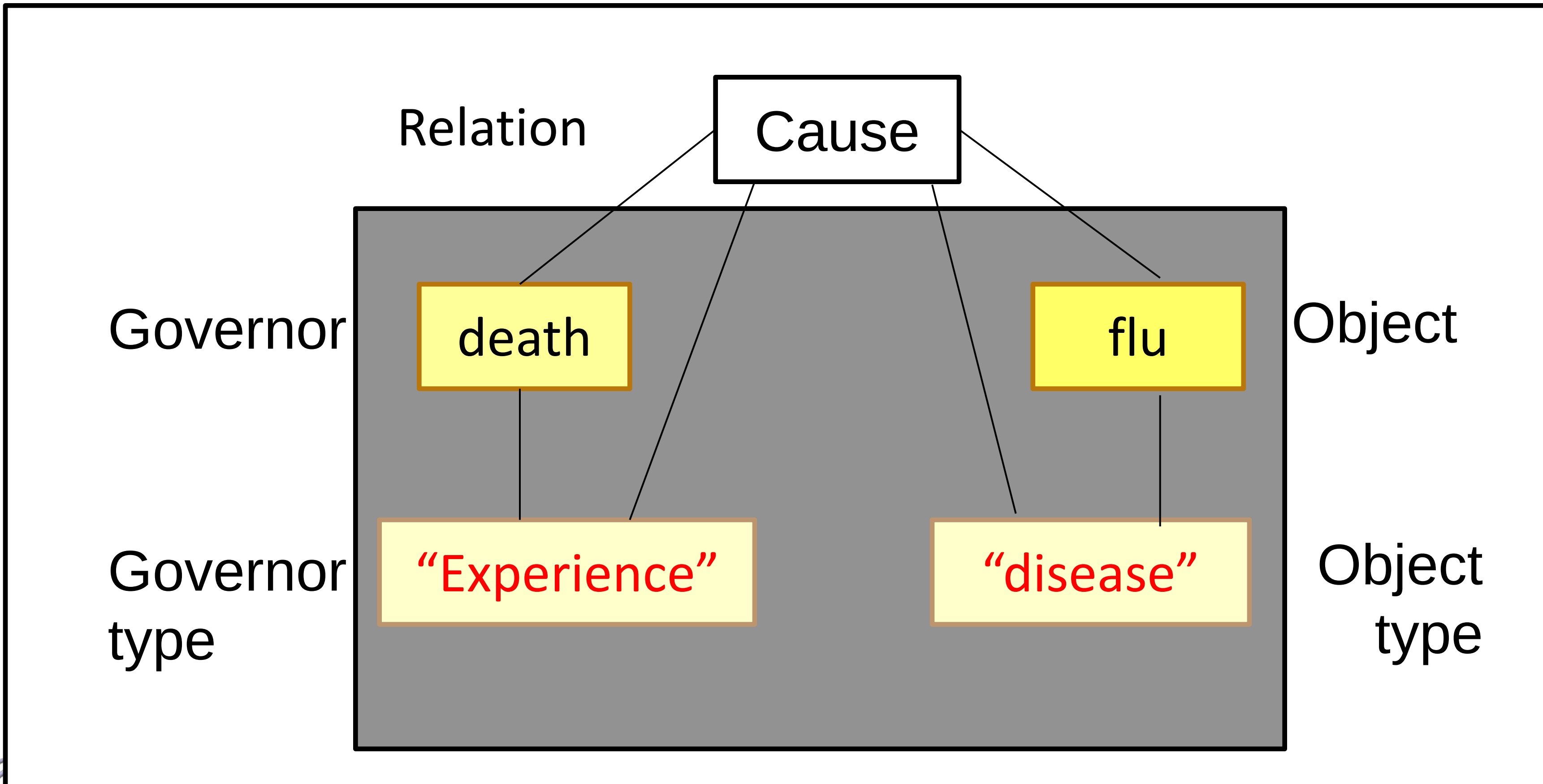
Poor care led to her death **from** flu.



ESRL II: Predicate-Argument Structure of Prepositions

……………her to suffer **from** infection.

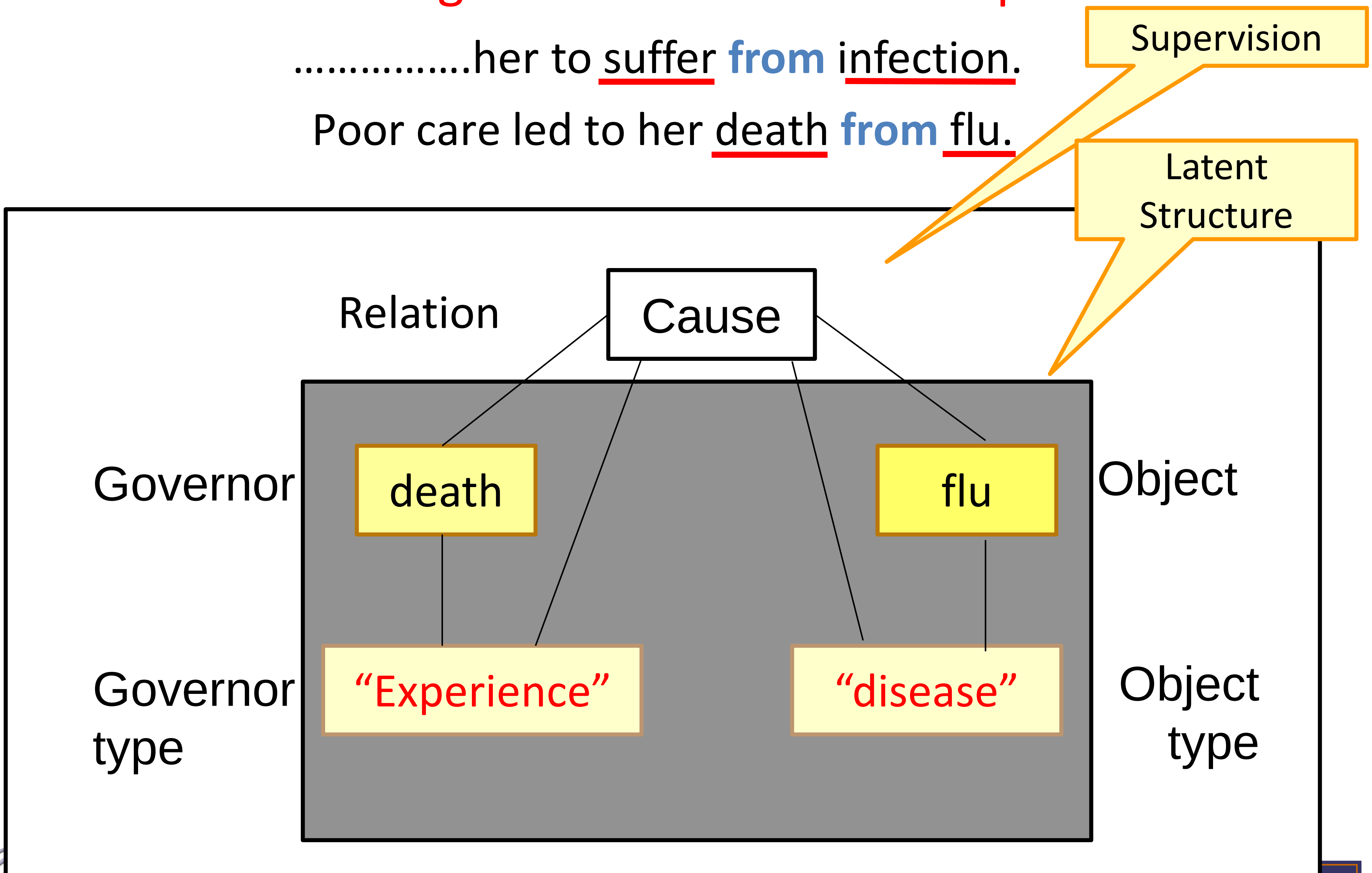
Poor care led to her death **from** flu.



ESRL II: Predicate-Argument Structure of Prepositions

……………her to suffer **from** infection.

Poor care led to her death **from** flu.



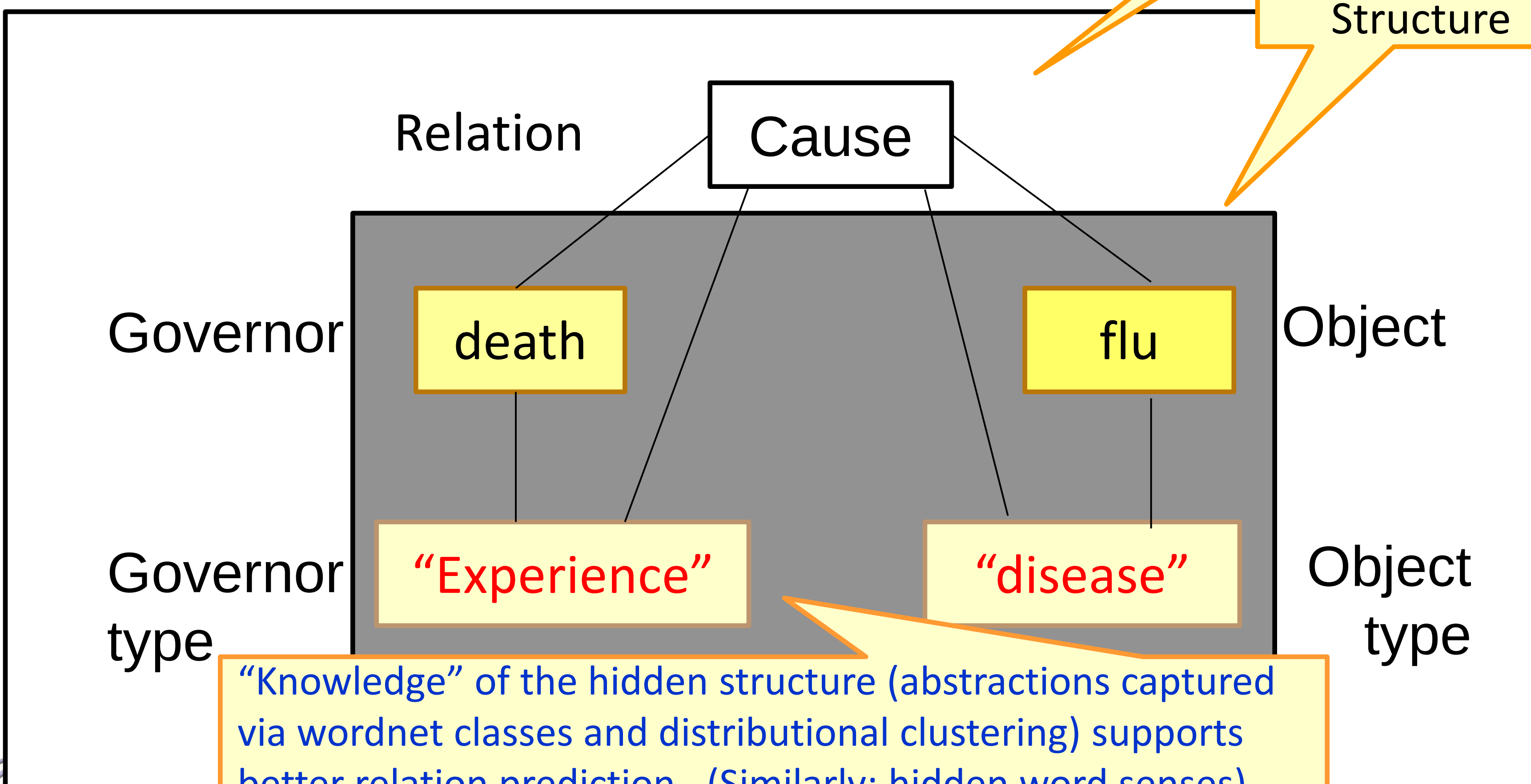
ESRL II: Predicate-Argument Structure of Prepositions

……………her to suffer **from** infection.

Poor care led to her death **from** flu.

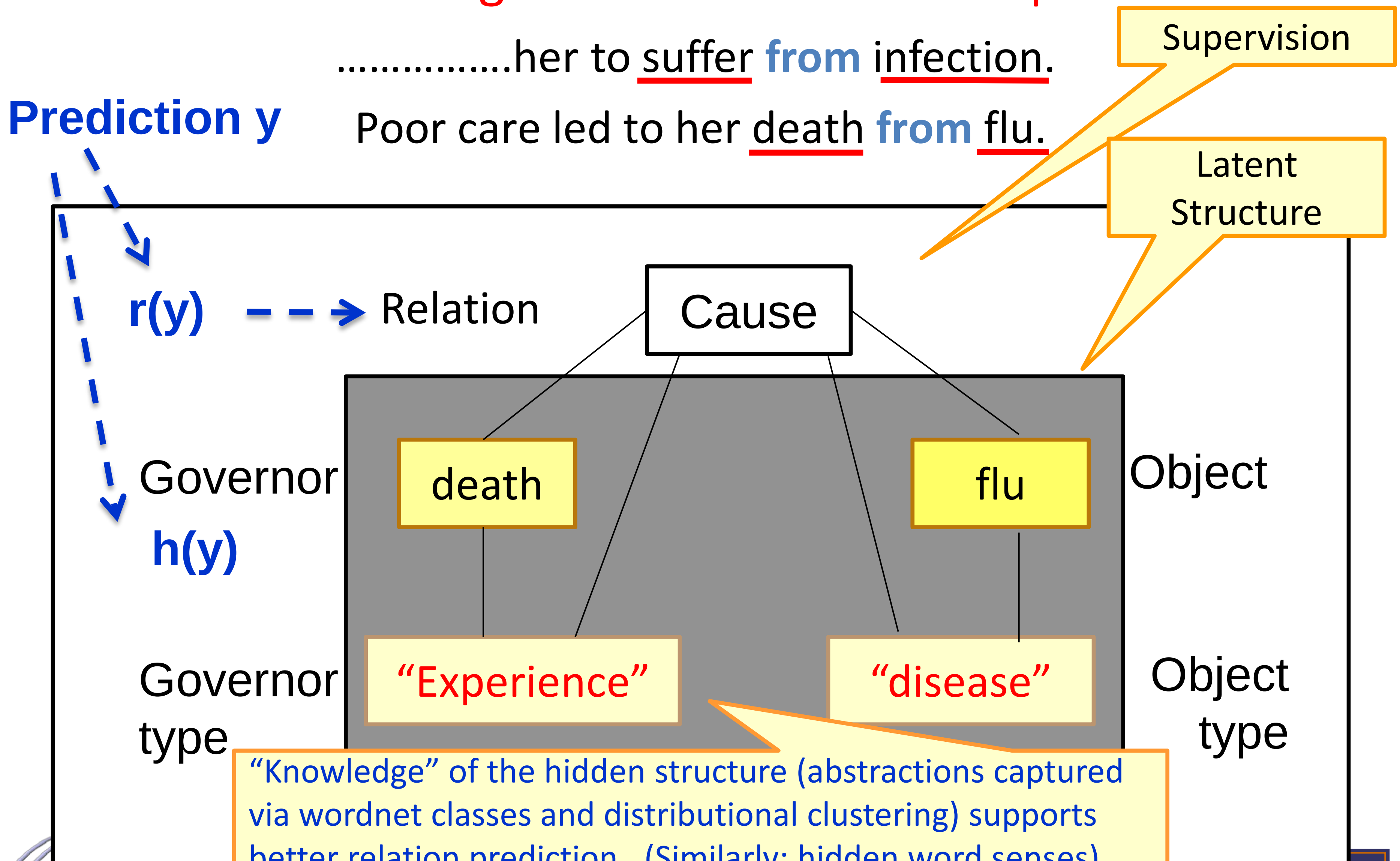
Supervision

Latent Structure



"Knowledge" of the hidden structure (abstractions captured via wordnet classes and distributional clustering) supports better relation prediction. (Similarly: hidden word senses)
Inference relating latent and observed variables is a CCM

ESRL II: Predicate-Argument Structure of Prepositions



"Knowledge" of the hidden structure (abstractions captured via wordnet classes and distributional clustering) supports better relation prediction. (Similarly: hidden word senses)
Inference relating latent and observed variables is a CCM

Learning with Latent Inference

- Given an example annotated with $r(y^*)$, predict with:

$$\text{argmax}_y \, w^T \phi(x,[r(y),h(y)])$$
$$\text{s.t } r(y^*) = r(y)$$

- While satisfying constraints between $r(y)$ and $h(y)$

Learning with Latent Inference

Inference takes into account constrains among parts of the structure (r and h), formulated as a CCM

- Given an example annotated with $r(y^*)$, predict with:

$$\arg\max_y w^T \phi(x,[r(y),h(y)])$$
$$\text{s.t } r(y^*) = r(y)$$

- While satisfying constraints between $r(y)$ and $h(y)$

Learning with Latent Inference

Inference takes into account constrains among parts of the structure (r and h), formulated as a CCM

- Given an example annotated with $r(y^*)$, predict with:

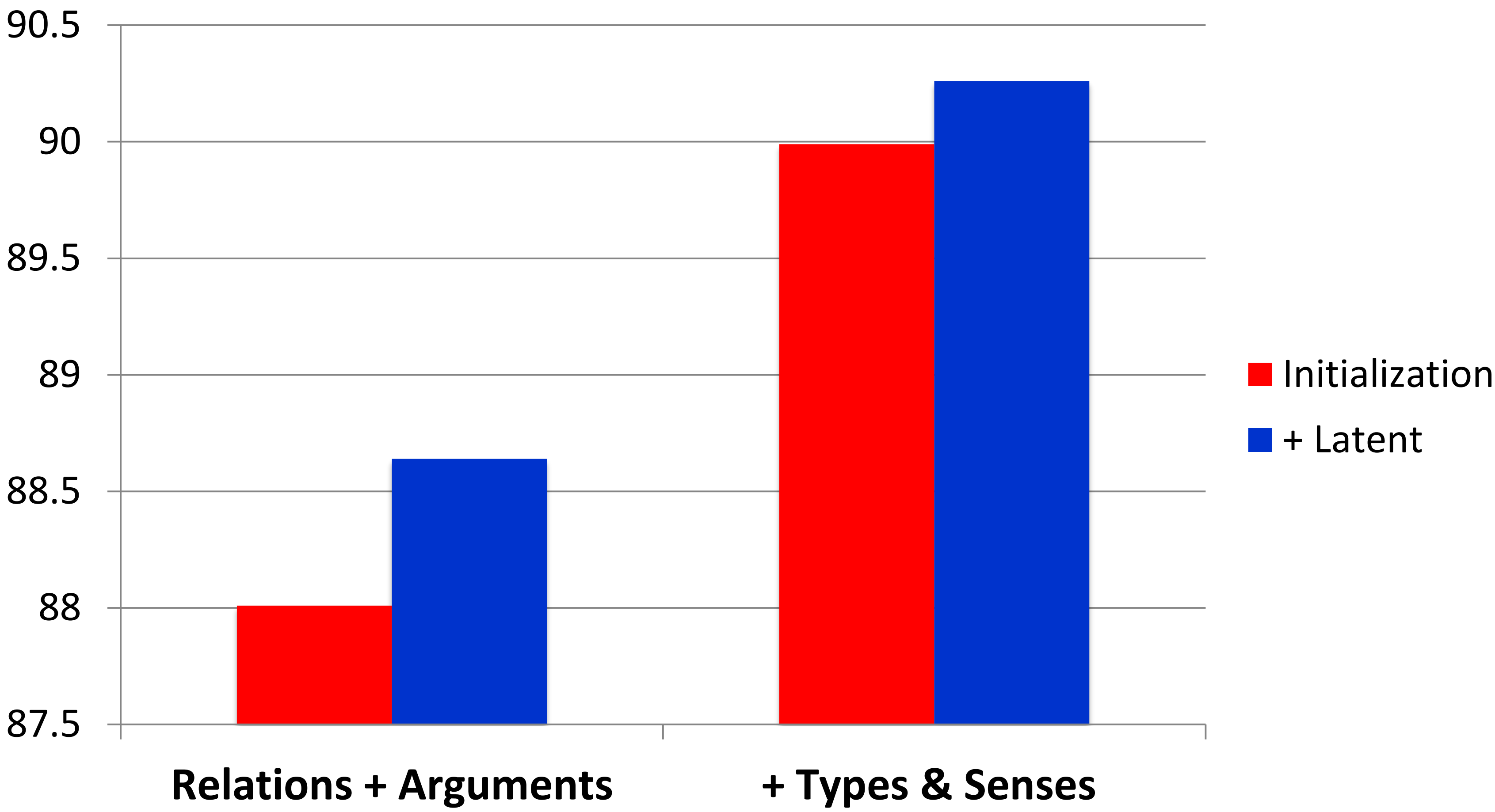
$$\arg\max_y w^T \phi(x,[r(y),h(y)])$$
$$\text{s.t } r(y^*) = r(y)$$

- While satisfying constraints between r(y) and h(y)
- That is: "complete the hidden structure" in the best possible

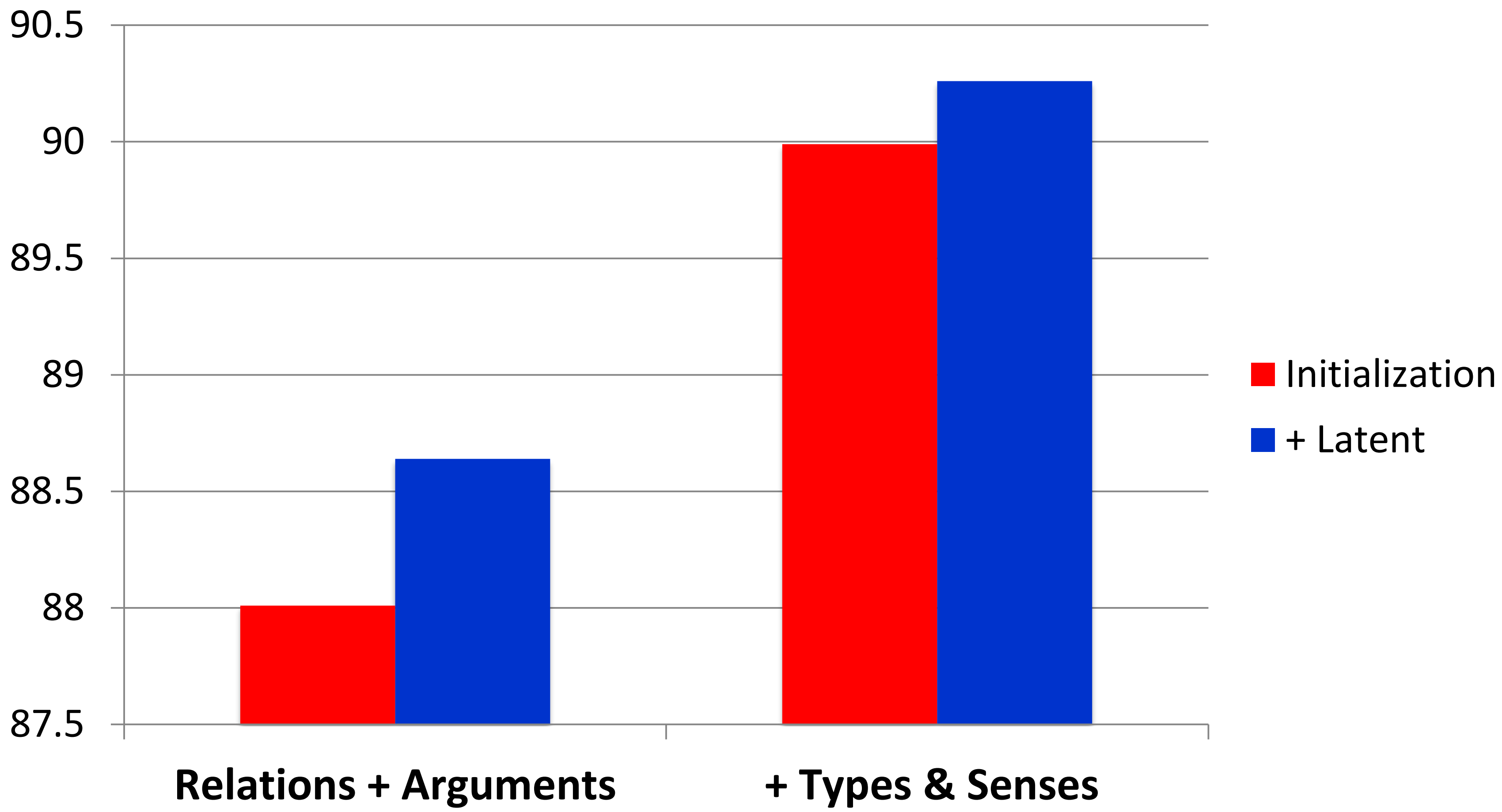
Generalization of Latent Structure SVM [Yu & Joachims '09] &
Indirect Supervision learning [Chang et. al. '10]

Performance

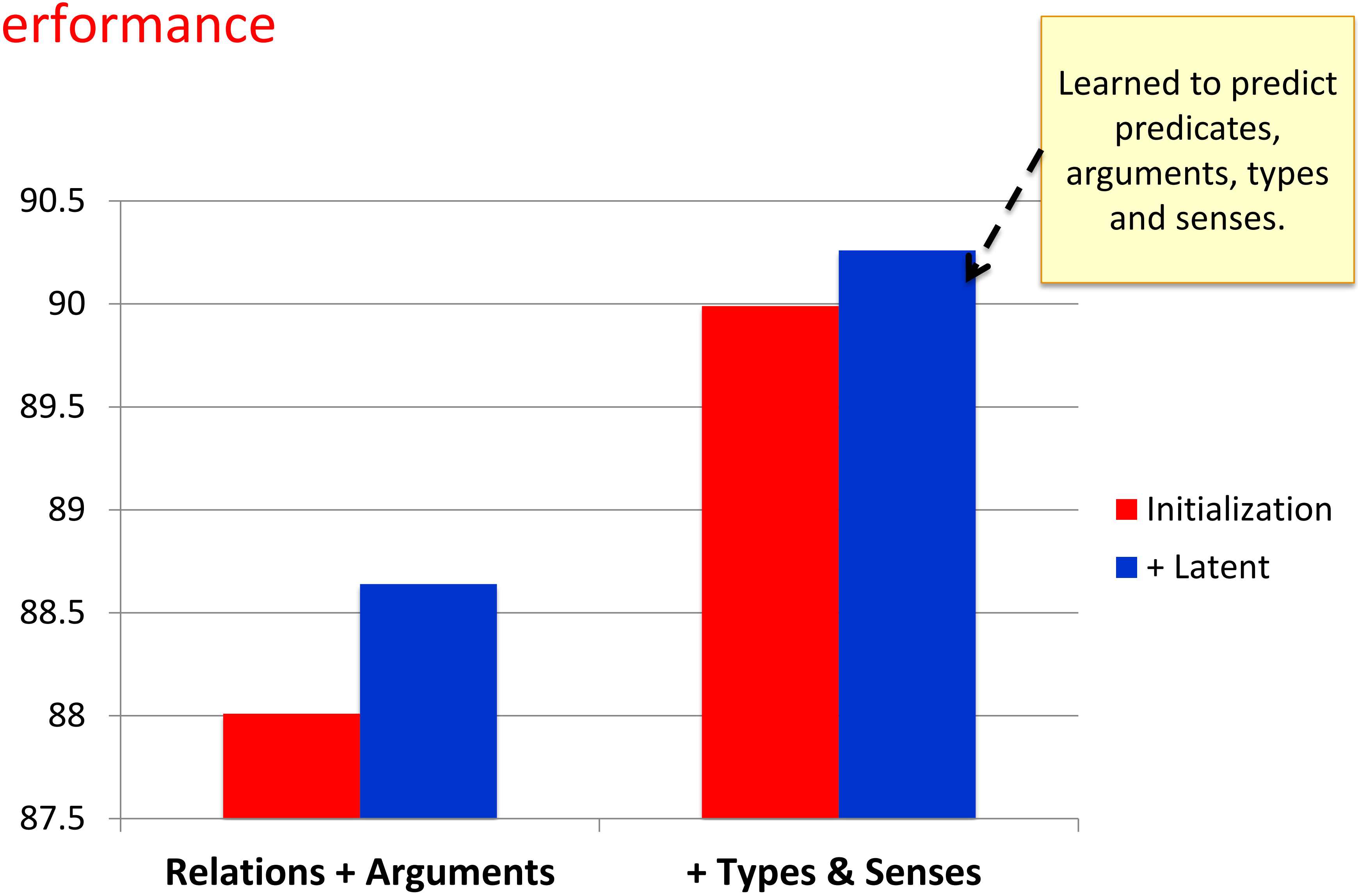
Performance



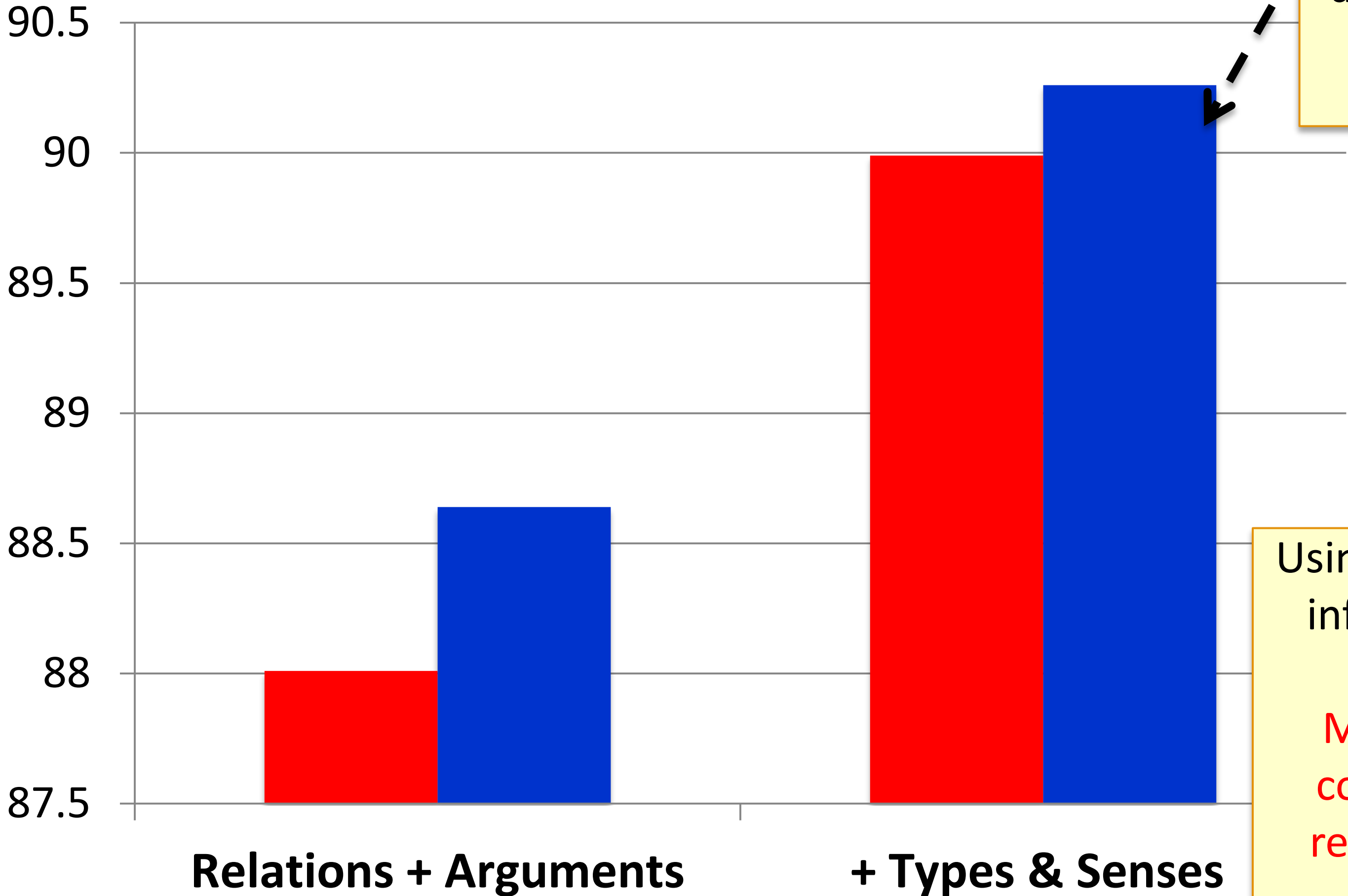
Performance



Performance



Performance

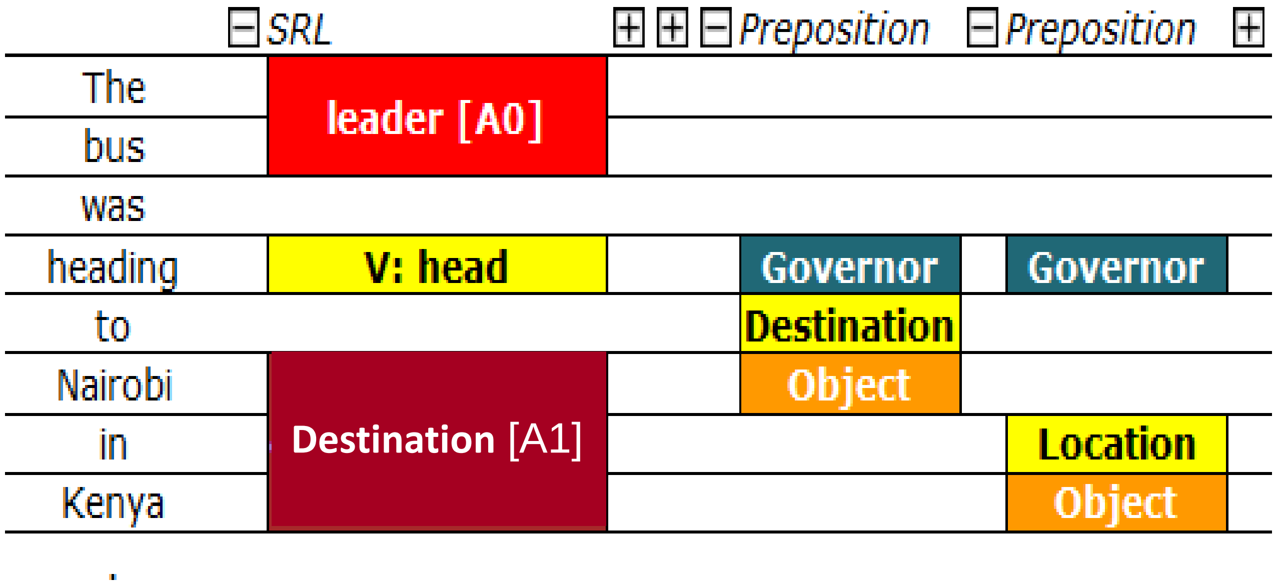


Learned to predict predicates, arguments, types and senses.

Using types helps. Joint inference with word sense helps too

More components constrain inference results and improve performance

Extended SRL [Demo]



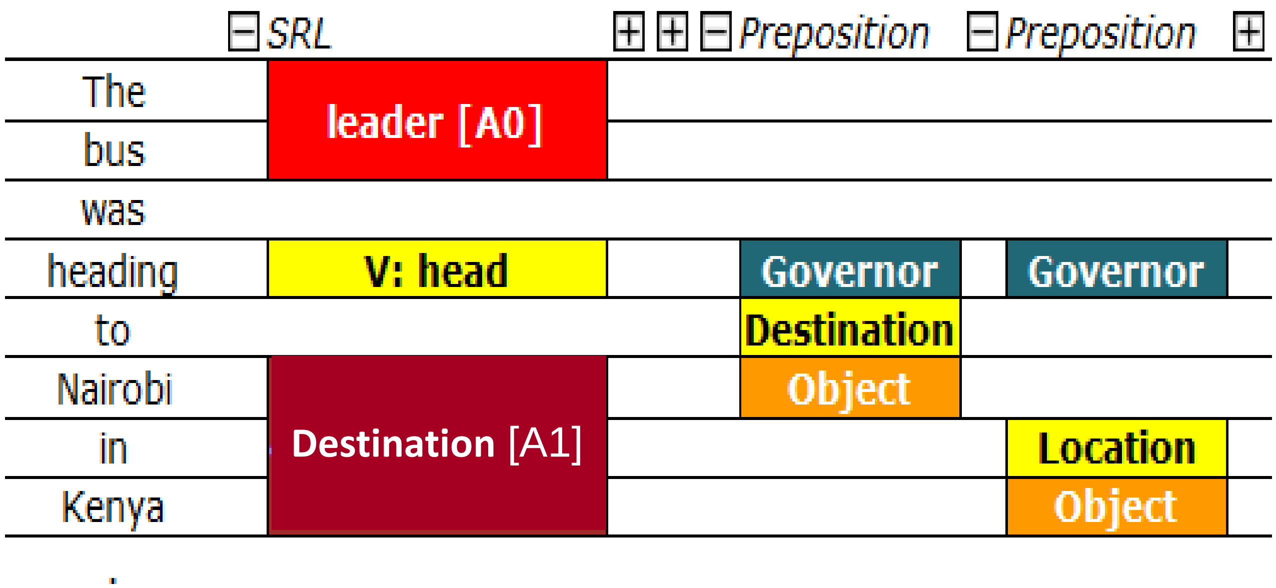
Extended SRL [Demo]

	SRL	Preposition	Preposition
The	leader [A0]		
bus			
was			
heading	V: head	Governor	Governor
to		Destination	
Nairobi	Destination [A1]	Object	
in			Location
Kenya			Object
.			

Joint inference over phenomena specific models to enforce consistency

Models trained with latent structure: senses, types, arguments

Extended SRL [Demo]



Joint inference over phenomena specific models to enforce consistency

Models trained with latent structure: senses, types, arguments

- More to do with other relations, discourse phenomena,…

Constrained Conditional Models—ILP Formulations

- **Have been shown useful in the context of many NLP problems**
- [Roth&Yih, 04,07: Entities and Relations; Punyakanok et. al: SRL …]
  - Summarization; Co-reference; Information & Relation Extraction; Event Identifications and causality ; Transliteration; Textual Entailment; Knowledge Acquisition; Sentiments; Temporal Reasoning, Parsing,…
- Some theoretical work on training paradigms [Punyakanok et. al., 05 more; Constraints Driven Learning, PR, Constrained EM…]
- Some work on Inference, mostly approximations, bringing back ideas on Lagrangian relaxation, etc.

Constrained Conditional Models—ILP Formulations

- **Have been shown useful in the context of many NLP problems**
- [Roth&Yih, 04,07: Entities and Relations; Punyakanok et. al: SRL …]
  - Summarization; Co-reference; Information & Relation Extraction; Event Identifications and causality ; Transliteration; Textual Entailment; Knowledge Acquisition; Sentiments; Temporal Reasoning, Parsing,…
- Some theoretical work on training paradigms [Punyakanok et. al., 05 more; Constraints Driven Learning, PR, Constrained EM…]
- Some work on Inference, mostly approximations, bringing back ideas on Lagrangian relaxation, etc.
- Good summary and description of training paradigms: [Chang, Ratinov & Roth, Machine Learning Journal 2012]
- **Summary of work & a bibliography: http://L2R.cs.uiuc.edu/tutorials.html**

Outline

- Constrained Conditional Models
  - A formulation for global inference with knowledge modeled as expressive structural constraints
  - Some examples
- Learning with Constrained Latent Representation
- Constraints Driven Learning
  - Training Paradigms for Constrained Conditional Models
  - Constraints Driven Learning (CoDL)
  - Unified (Constrained) Expectation Maximization
- Amortized Integer Linear Programming Inference
  - Exploiting Previous Inference Results
    - **In Inference and in Structured Learning**

Constrained Conditional Models (aka ILP Inference)

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Penalty for violating the constraint.

(Soft) constraints component

Weight Vector for "local" models

Features, classifiers; log-linear models (HMM, CRF) or a combination

How far y is from a "legal" assignment

How to solve?

This is an Integer Linear Program

Solving using ILP packages gives an exact solution.

Cutting Planes, Dual Decomposition & other search techniques are possible

How to train?

Training is learning the objective function

Decouple? Decompose?

How to exploit the structure to minimize supervision?

Constrained Conditional Models (aka ILP Inference)

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Penalty for violating the constraint.

(Soft) constraints component

Weight Vector for "local" models

Features, classifiers; log-linear models (HMM, CRF) or a combination

How far y is from a "legal" assignment

How to solve?

This is an Integer Linear Program

Solving using ILP packages gives an exact solution.

Cutting Planes, Dual Decomposition & other search techniques are possible

How to train?

Training is learning the objective function

Decouple? Decompose?

How to exploit the structure to minimize supervision?

Training Constrained Conditional Models

$$\underset{y}{\operatorname{argmax}} \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Training Constrained Conditional Models

$$\underset{y}{\text{argmax}}\, \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Decompose Model from constraints

- Training:
  - Independently of the constraints (L+I)

Training Constrained Conditional Models

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Decompose Model from constraints

- Training:
  - Independently of the constraints (L+I)
  - Jointly, in the presence of the constraints (IBT)

Training Constrained Conditional Models

Decompose Model

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Decompose Model from constraints

- Training:
  - Independently of the constraints (L+I)
  - Jointly, in the presence of the constraints (IBT)
  - Decomposed to simpler models

Training Constrained Conditional Models

Decompose Model

$$\underset{y}{\operatorname{argmax}}\ \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Decompose Model from constraints

- Training:
  - Independently of the constraints (L+I)
  - Jointly, in the presence of the constraints (IBT)
  - Decomposed to simpler models
- There has been a lot of work, theoretical and experimental, on these issues, starting with [Punyakanok et. al IJCAI'05]
- Not surprisingly, decomposition is good. [Samdani et. al ICML'12]

Training Constrained Conditional Models

Decompose Model

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Decompose Model from constraints

- Training:
  - Independently of the constraints (L+I)
  - Jointly, in the presence of the constraints (IBT)
  - Decomposed to simpler models
- There has been a lot of work, theoretical and experimental, on these issues, starting with [Punyakanok et. al IJCAI'05]
- Not surprisingly, decomposition is good. [Samdani et. al ICML'12]
- There has been a lot of work on exploiting CCMs in learning structures with indirect supervision [Chang et. al, NAACL'10, ICML'10]

Training Constrained Conditional Models

Decompose Model

$$\operatorname*{argmax}_{y} \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Decompose Model from constraints

- Training:
  - Independently of the constraints (L+I)
  - Jointly, in the presence of the constraints (IBT)
  - Decomposed to simpler models
- There has been a lot of work, theoretical and experimental, on these issues, starting with [Punyakanok et. al IJCAI'05]
- Not surprisingly, decomposition is good. [Samdani et. al ICML'12]

- There has been a lot of work on exploiting CCMs in learning structures with indirect supervision [Chang et. al, NAACL'10, ICML'10]
- And Response based Learning [Goldwasser et. al'12, '14]

Information extraction without Prior Knowledge

Lars Ole Andersen . Program analysis and specialization for the C Programming language. PhD thesis. DIKU , University of Copenhagen, May 1994 .

Prediction result of a trained HMM

[AUTHOR]	Lars Ole Andersen . Program analysis and
[TITLE]	specialization for the
[EDITOR]	C
[BOOKTITLE]	Programming language
[TECH-REPORT]	. PhD thesis .
[INSTITUTION]	DIKU , University of Copenhagen , May
[DATE]	1994 .

Information extraction without Prior Knowledge

Lars Ole Andersen . Program analysis and specialization for the C Programming language. PhD thesis. DIKU , University of Copenhagen, May 1994 .

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x, y)$$

Prediction result of a trained HMM

[AUTHOR]	Lars Ole Andersen . Program analysis and
[TITLE]	specialization for the
[EDITOR]	C
[BOOKTITLE]	Programming language
[TECH-REPORT]	. PhD thesis .
[INSTITUTION]	DIKU , University of Copenhagen , May
[DATE]	1994 .

Information extraction without Prior Knowledge

Lars Ole Andersen . Program analysis and specialization for the C Programming language. PhD thesis. DIKU , University of Copenhagen, May 1994 .

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x, y)$$

Prediction result of a trained HMM

[AUTHOR]	Lars Ole Andersen . Program analysis and
[TITLE]	specialization for the
[EDITOR]	C
[BOOKTITLE]	Programming language
[TECH-REPORT]	. PhD thesis .
[INSTITUTION]	DIKU , University of Copenhagen , May
[DATE]	1994 .

Violates lots of natural constraints!

Strategies for Improving the Results

Strategies for Improving the Results

- (Pure) Machine Learning Approaches
  - Higher Order HMM/CRF?
  - Increasing the window size?
  - Adding a lot of new features
    - Requires a lot of labeled examples

Increasing the model complexity

Increase difficulty of Learning

Strategies for Improving the Results

- (Pure) Machine Learning Approaches
  - Higher Order HMM/CRF?
  - Increasing the window size?
  - Adding a lot of new features
    - Requires a lot of labeled examples
  - What if we only have a few labeled examples?

Increasing the model complexity

Increase difficulty of Learning

Strategies for Improving the Results

- (Pure) Machine Learning Approaches
  - Higher Order HMM/CRF?
  - Increasing the window size?
  - Adding a lot of new features
    - Requires a lot of labeled examples
  - What if we only have a few labeled examples?

Increasing the model complexity

Increase difficulty of Learning

Can we keep the learned model simple and still make expressive decisions?

Strategies for Improving the Results

- (Pure) Machine Learning Approaches
  - Higher Order HMM/CRF?
  - Increasing the window size?
  - Adding a lot of new features
    - Requires a lot of labeled examples
  - What if we only have a few labeled examples?

Increasing the model complexity

Increase difficulty of Learning

Can we keep the learned model simple and still make expressive decisions?

- Other options?
  - Constrain the output to make sense
  - Push the (simple) model in a direction that makes sense

Examples of Constraints

- Each field must be a consecutive list of words and can appear at most once in a citation.
- State transitions must occur on punctuation marks.
- The citation can only start with *AUTHOR* or *EDITOR*.
- The words *pp., pages* correspond to *PAGE.*
- Four digits starting with 20xx and 19xx are *DATE*.
- Quotations can appear only in *TITLE*
- …….

Examples of Constraints

- Each field must be a consecutive list of words and can appear at most once in a citation.
- State transitions must occur on punctuation marks.
- The citation can only start with *AUTHOR* or *EDITOR*.
- The words *pp., pages* correspond to *PAGE.*
- Four digits starting with 20xx and 19xx are *DATE*.
- Quotations can appear only in *TITLE*
- …….

Easy to express pieces of "knowledge"

Examples of Constraints

- Each field must be a consecutive list of words and can appear at most once in a citation.
- State transitions must occur on punctuation marks.
- The citation can only start with *AUTHOR* or *EDITOR*.
- The words *pp., pages* correspond to *PAGE.*
- Four digits starting with 20xx and 19xx are *DATE*.
- Quotations can appear only in *TITLE*
- …….

Easy to express pieces of "knowledge"

Non Propositional; May use Quantifiers

Information Extraction with "Expectation" Constraints

- Adding constraints, we get correct results!
  - **Without changing the model**

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x, y)$$

- *[AUTHOR]* Lars Ole Andersen .
  [TITLE] Program analysis and specialization for the C Programming language .
  [TECH-REPORT] PhD thesis .
  [INSTITUTION] DIKU , University of Copenhagen ,
  [DATE] May, 1994 .

Information Extraction with "Expectation" Constraints

- Adding constraints, we get correct results!
  - **Without changing the model**

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x, y)$$

- *[AUTHOR]* Lars Ole Andersen .
  [TITLE] Program analysis and specialization for the C Programming language .
  [TECH-REPORT] PhD thesis .
  [INSTITUTION] DIKU , University of Copenhagen ,
  [DATE] May, 1994 .

Information Extraction with “Expectation” Constraints

- Adding constraints, we get correct results!
  - **Without changing the model**

$$\arg\max_{y} \boldsymbol{\lambda} \cdot F(x,y) \boxed{- \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})}$$

- *[AUTHOR]* Lars Ole Andersen .
  [TITLE] Program analysis and specialization for the C Programming language .
  [TECH-REPORT] PhD thesis .
  [INSTITUTION] DIKU , University of Copenhagen ,
  [DATE] May, 1994 .

Information Extraction with "Expectation" Constraints

- Adding constraints, we get correct results!
  - **Without changing the model**

$$\arg\max_{y} \boldsymbol{\lambda} \cdot F(x,y) \boxed{- \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})}$$

- *[AUTHOR]* Lars Ole Andersen .
  [TITLE] Program analysis and specialization for the C Programming language .
  [TECH-REPORT] PhD thesis .
  [INSTITUTION] DIKU , University of Copenhagen ,
  [D

Constrained Conditional Models Allow:

- **Learning a simple model**
- **Make decisions with a more complex model**
- **Accomplished by directly incorporating constraints to bias/re-rank decisions made by the simpler model**

Guiding (Semi-Supervised) Learning with Constraints

Guiding (Semi-Supervised) Learning with Constraints

Seed examples → Model

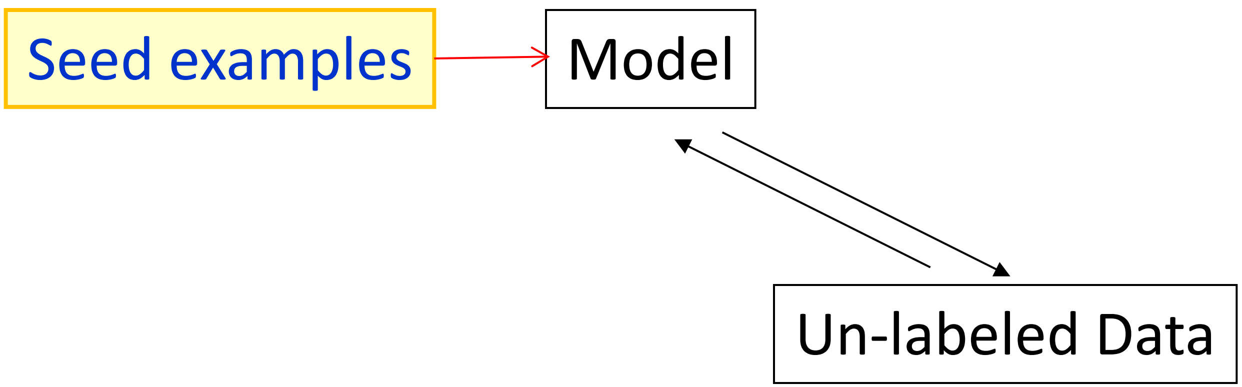
Un-labeled Data

Guiding (Semi-Supervised) Learning with Constraints

Seed examples → Model → Un-labeled Data

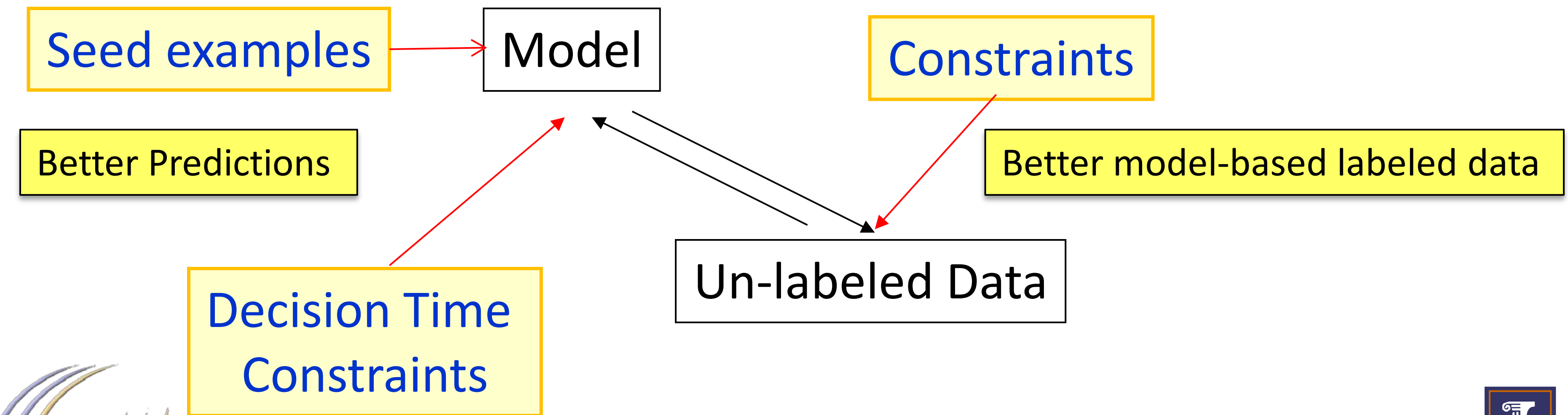
Guiding (Semi-Supervised) Learning with Constraints

- In traditional Semi-Supervised learning the model can drift away from the correct one.



Guiding (Semi-Supervised) Learning with Constraints

- In traditional Semi-Supervised learning the model can drift away from the correct one.
- Constraints can be used to generate better training data
  - At training to improve labeling of un-labeled data (and thus improve the model)
  - At decision time, to bias the objective function towards favoring constraint satisfaction.



Constraints Driven Learning (CoDL)

[Chang, Ratinov, Roth, ACL'07;ICML'08,MLJ'12]
See also: Ganchev et. al. 10 (PR)

(w,ρ)=learn(L)

For N iterations do

$T=\phi$

For each x in unlabeled dataset

$h \leftarrow \text{argmax}_y \, w^T \phi(x,y) - \sum \rho \, d_C(x,y)$

$T=T \cup \{(x, h)\}$

$(w,\rho) = \gamma \, (w,\rho) + (1- \gamma) \text{ learn}(T)$

Constraints Driven Learning (CoDL)

[Chang, Ratinov, Roth, ACL'07;ICML'08,MLJ'12]
See also: Ganchev et. al. 10 (PR)

(w,ρ)=learn(L)

Supervised learning algorithm parameterized by (w,ρ). Learning can be justified as an optimization procedure for an objective function

For N iterations do

$T=\phi$

For each x in unlabeled dataset

$h \leftarrow \text{argmax}_y\ w^T \phi(x,y) - \sum \rho\ d_C(x,y)$

$T=T \cup \{(x, h)\}$

$(w,\rho) = \gamma\ (w,\rho) + (1- \gamma)$ learn(T)

Constraints Driven Learning (CoDL)

[Chang, Ratinov, Roth, ACL'07;ICML'08,MLJ'12]
See also: Ganchev et. al. 10 (PR)

(w,ρ)=learn(L)

For N iterations do

$T=\phi$

For each x in unlabeled dataset

$h \leftarrow \text{argmax}_y \, w^T \phi(x,y) - \sum \rho \, d_C(x,y)$

$T = T \cup \{(x, h)\}$

$(w,\rho) = \gamma \, (w,\rho) + (1-\gamma) \, \text{learn}(T)$

Supervised learning algorithm parameterized by (w,ρ). Learning can be justified as an optimization procedure for an objective function

Inference with constraints: augment the training set

Constraints Driven Learning (CoDL)

[Chang, Ratinov, Roth, ACL'07;ICML'08,MLJ'12]
See also: Ganchev et. al. 10 (PR)

(w,ρ)=learn(L)

For N iterations do

$T=\phi$

For each x in unlabeled dataset

$h \leftarrow \text{argmax}_y \, w^T \phi(x,y) - \sum \rho \, d_C(x,y)$

$T = T \cup \{(x, h)\}$

$(w,\rho) = \gamma \, (w,\rho) + (1-\gamma) \text{ learn}(T)$

Supervised learning algorithm parameterized by (w,ρ). Learning can be justified as an optimization procedure for an objective function

Inference with constraints: augment the training set

Learn from new training data Weigh supervised & unsupervised models.

Constraints Driven Learning (CoDL)

Archetypical Semi/un-supervised learning: **A constrained EM**

[Chang, Ratinov, Roth, ACL'07;ICML'08,MLJ'12]
See also: Ganchev et. al. 10 (PR)

(w,ρ)=learn(L)

For N iterations do

$T=\phi$

For each x in unlabeled dataset

$h \leftarrow \mathrm{argmax}_y\, w^T \phi(x,y) - \sum \rho\, d_C(x,y)$

$T=T \cup \{(x, h)\}$

$(w,\rho) = \gamma\,(w,\rho) + (1-\gamma)$ learn(T)

Supervised learning algorithm parameterized by (w,ρ). Learning can be justified as an optimization procedure for an objective function

Inference with constraints: augment the training set

Learn from new training data
Weigh supervised & unsupervised models.

Constraints Driven Learning (CoDL)

Archetypical Semi/un-supervised learning: **A constrained EM**

[Chang, Ratinov, Roth, ACL'07;ICML'08,MLJ'12]
See also: Ganchev et. al. 10 (PR)

Supervised learning algorithm parameterized by (w,ρ). Learning can be justified as an optimization procedure for an objective function

(w,ρ)=learn(L)

For N iterations do

$T=\phi$

For each x in unlabeled dataset

$h \leftarrow \text{argmax}_y\ w^T \phi(x,y) - \sum \rho\, d_C(x,y)$

$T = T \cup \{(x, h)\}$

Inference with constraints: augment the training set

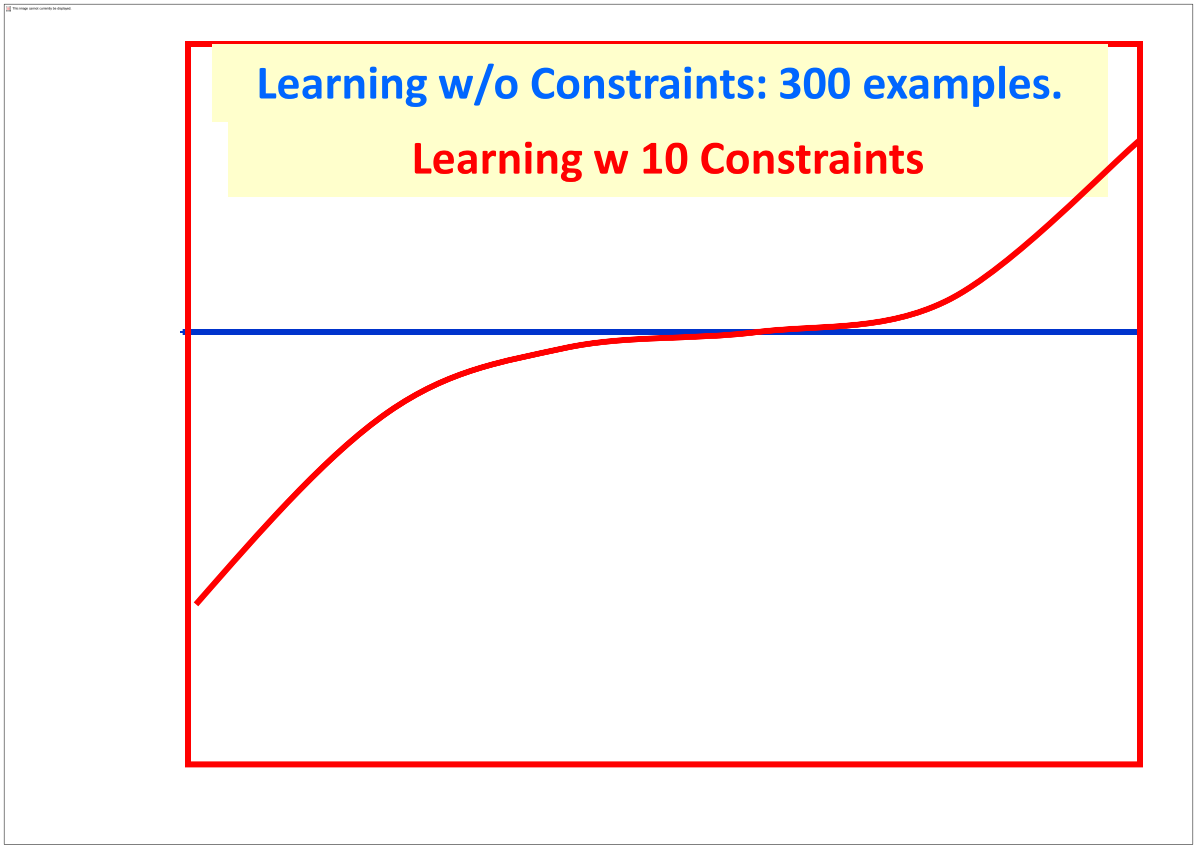
$(w,\rho) = \gamma\,(w,\rho) + (1-\gamma)$ learn(T)

Learn from new training data Weigh supervised & unsupervised models.

Excellent Experimental Results showing the advantages of using constraints, especially with small amounts of labeled data [Chang et. al, Others]

Value of Constraints in Semi-Supervised Learning

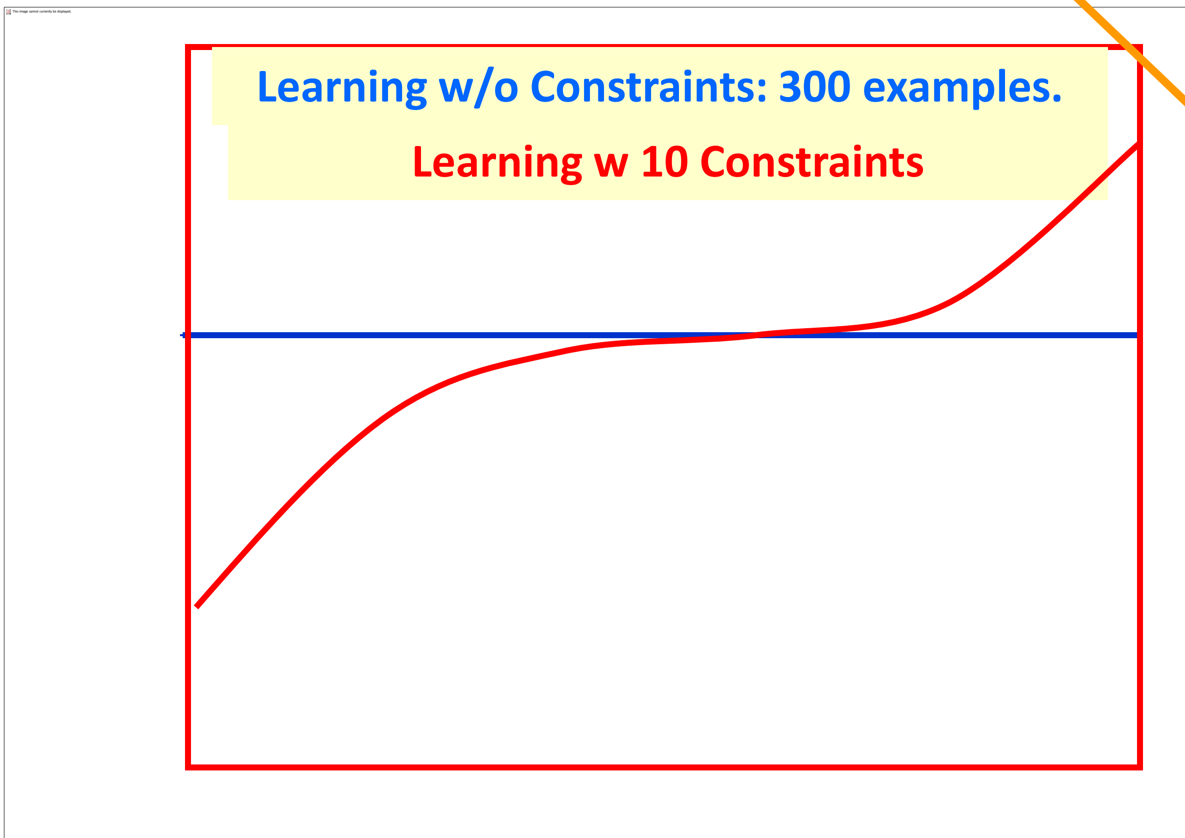
Objective function:



# of available labeled examples

Value of Constraints in Semi-Supervised Learning

Objective function:



# of available labeled examples

Constraints are used to Bootstrap a semi-supervised learner
Poor model + constraints used to annotate unlabeled data, which in turn is used to keep training the model.

CoDL as Constrained Hard EM

- Hard EM is a popular variant of EM
- While EM estimates a distribution over all y variables in the E-step,
- … Hard EM predicts the best output in the E-step

$$y^* = \mathbf{argmax}_{\mathrm{y}}\ P_w(\mathbf{y}\,|\,\mathbf{x})$$

- Alternatively, hard EM predicts a peaked distribution

$$q(y) = \delta_{y=y^*}$$

- Constrained-Driven Learning (CODL) – can be viewed as a constrained version of hard EM:

$$y^* = \mathbf{argmax}_{\mathrm{y:Uy \leq b}}\ P_w(y|x)$$

CoDL as Constrained Hard EM

- Hard EM is a popular variant of EM
- While EM estimates a distribution over all y variables in the E-step,
- … Hard EM predicts the best output in the E-step

$$y^* = \mathbf{argmax}_{\mathrm{y}}\ P_w(\mathbf{y}\,|\,\mathbf{x})$$

- Alternatively, hard EM predicts a peaked distribution

$$q(y) = \delta_{y=y^*}$$

- Constrained-Driven Learning (CODL) – can be viewed as a constrained version of hard EM:

Constraining the feasible set

$$y^* = \mathbf{argmax}_{\mathrm{y:Uy} \leq \mathrm{b}}\ P_w(y\,|\,x)$$

Constrained EM: Two Versions

- While Constrained-Driven Learning [CODL; Chang et al, 07,12] is a constrained version of hard EM:

Constraining the feasible set

$$y^* = \text{argmax}_{\mathbf{y:Uy \leq b}} P_w(y|x)$$

- … It is possible to derive a constrained version of EM:

Constrained EM: Two Versions

- While Constrained-Driven Learning [CODL; Chang et al, 07,12] is a constrained version of hard EM:

Constraining the feasible set

$$y^* = \mathrm{argmax}_{\mathbf{y}:\mathbf{Uy}\leq\,\mathbf{b}}\; P_w(y|x)$$

- ... It is possible to derive a constrained version of EM:
- To do that, constraints are relaxed into expectation constraints on the posterior probability q:

$$\mathrm{E}_q[\boldsymbol{U}\boldsymbol{y}] \leq \boldsymbol{b}$$

Constrained EM: Two Versions

- While Constrained-Driven Learning [CODL; Chang et al, 07,12] is a constrained version of hard EM:

Constraining the feasible set

$$y^* = \mathrm{argmax}_{\mathbf{y}:\mathbf{U}\mathbf{y} \leq \mathbf{b}} \; P_w(y|x)$$

- … It is possible to derive a constrained version of EM:
- To do that, constraints are relaxed into expectation constraints on the posterior probability q:

$$\mathrm{E}_q[\boldsymbol{U}\boldsymbol{y}] \leq \boldsymbol{b}$$

- The E-step now becomes: [Neal & Hinton '99 view of EM]

$$q' = \underset{q: q(\mathbf{y}) \geq 0, E_q[\mathbf{U}\mathbf{y}] \leq \mathbf{b}, \sum_{\mathbf{y}} q(\mathbf{y}) = 1}{\arg\min} KL(q(\mathbf{y}) || P(\mathbf{y}|\mathbf{x}, \mathbf{w}))$$

Constrained EM: Two Versions

- While Constrained-Driven Learning [CODL; Chang et al, 07,12] is a constrained version of hard EM:

Constraining the feasible set

$$y^* = \mathrm{argmax}_{\mathbf{y}:\mathbf{U}\mathbf{y}\leq\,\mathbf{b}}\; P_w(y|x)$$

- … It is possible to derive a constrained version of EM:
- To do that, constraints are relaxed into expectation constraints on the posterior probability q:

$$\mathrm{E}_q[\boldsymbol{U}\boldsymbol{y}] \leq \boldsymbol{b}$$

- The E-step now becomes: [Neal & Hinton '99 view of EM]

q' = $$\underset{q:q(\mathbf{y})\geq 0, E_q[\mathbf{U}\mathbf{y}]\leq \mathbf{b}, \sum_{\mathbf{y}} q(\mathbf{y})=1}{\arg\min} KL(q(\mathbf{y})||P(\mathbf{y}|\mathbf{x},\mathbf{w}))$$

- This is the Posterior Regularization model [PR; Ganchev et al, 10]

Which (Constrained) EM to use?

- that

Which (Constrained) EM to use?

- There is a lot of literature on EM vs hard EM
  - Experimentally, the bottom line is that with a good enough (???) initialization point, hard EM is probably better (and more efficient).
    - **E.g., EM vs hard EM (Spitkovsky et al, 10)**
  -

that

Which (Constrained) EM to use?

- There is a lot of literature on EM vs hard EM
  - Experimentally, the bottom line is that with a good enough (???) initialization point, hard EM is probably better (and more efficient).
    - **E.g., EM vs hard EM (Spitkovsky et al, 10)**
- Similar issues exist in the constrained case: CoDL vs. PR
  -

that

Which (Constrained) EM to use?

- that
- There is a lot of literature on EM vs hard EM
  - Experimentally, the bottom line is that with a good enough (???) initialization point, hard EM is probably better (and more efficient).
    - **E.g., EM vs hard EM (Spitkovsky et al, 10)**
- Similar issues exist in the constrained case: CoDL vs. PR

- Unified EM (UEM) [Samdani et. al., NAACL-12]
  - Provides a continuum of algorithms – from EM to hard EM, and infinitely many new EM algorithms in between.
  - Implementation wise, not more complicated than EM

Unifying Existing EM Algorithms

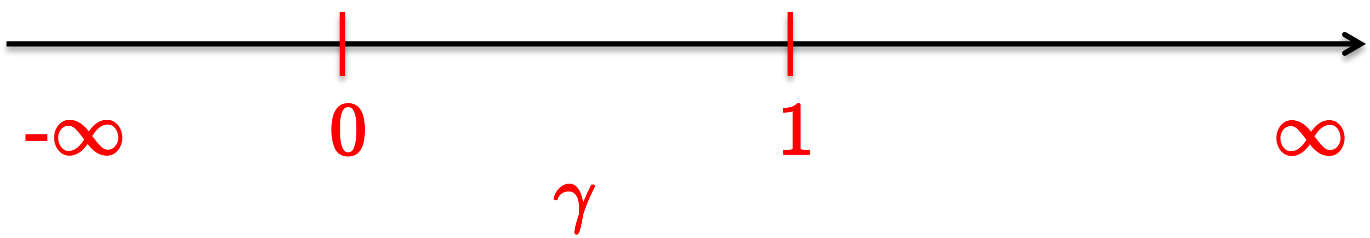
$$KL(q\ ,\ p; \gamma) = \sum_y \gamma\, q(y) \log q(y) - q(y) \log p(y)$$

Changing γ values results in different existing EM algorithms

Unifying Existing EM Algorithms

$$KL(q\,,\,p;\gamma) = \textstyle\sum_y \gamma\, q(y) \log q(y) - q(y) \log p(y)$$

Changing γ values results in different existing EM algorithms



Unifying Existing EM Algorithms

$$KL(q\ ,\ p;\ \gamma) = \sum_y \gamma\ q(y) \log q(y) - q(y) \log p(y)$$

Changing γ values results in different existing EM algorithms

No
Constraints

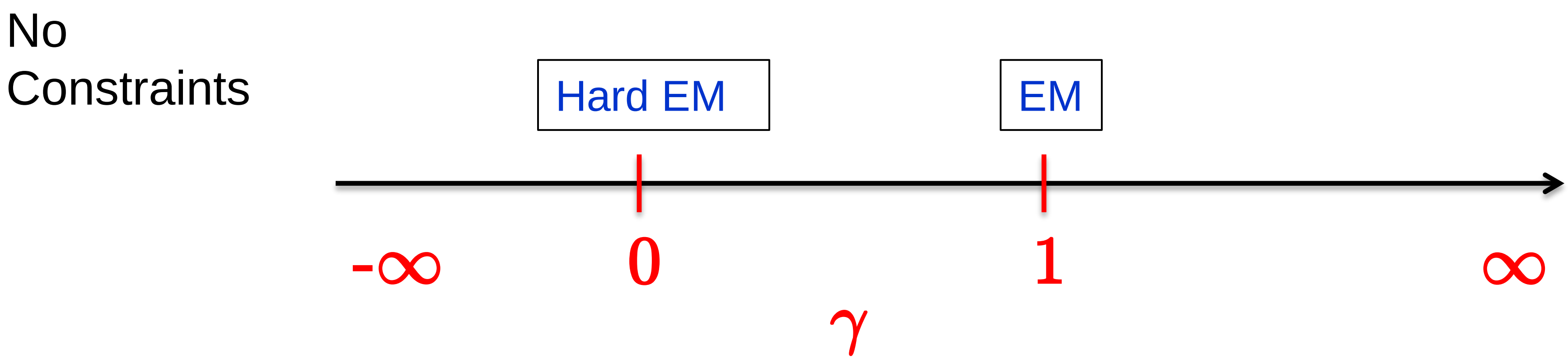
$-\infty$ 0 1 ∞

γ

Unifying Existing EM Algorithms

$$KL(q\ ,\ p;\ \gamma) = \textstyle\sum_y \gamma\ q(y) \log q(y) - q(y) \log p(y)$$

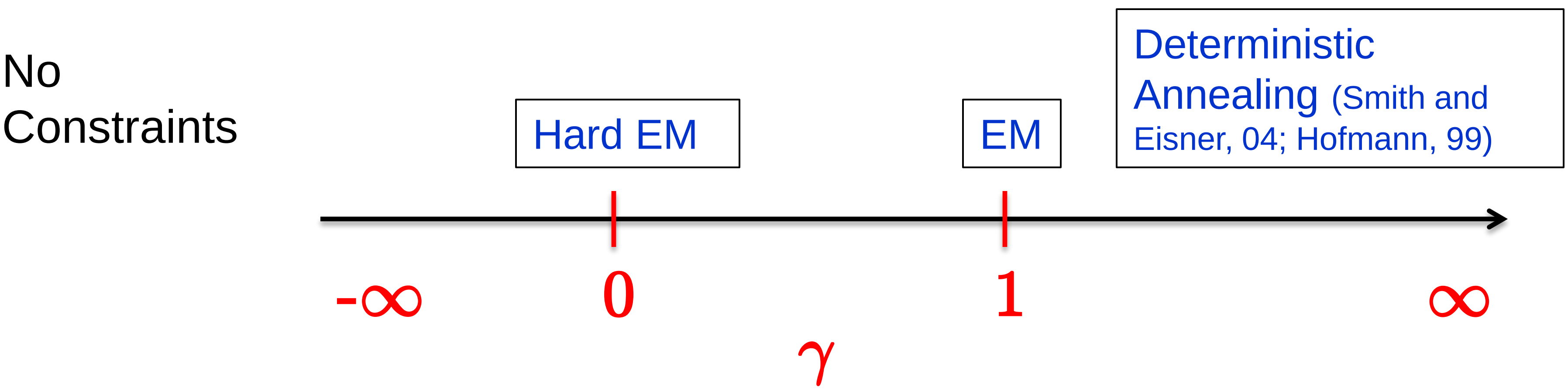
Changing γ values results in different existing EM algorithms



Unifying Existing EM Algorithms

$$KL(q\ ,\ p;\ \gamma) = \textstyle\sum_y \gamma\, q(y) \log q(y) - q(y) \log p(y)$$

Changing γ values results in different existing EM algorithms

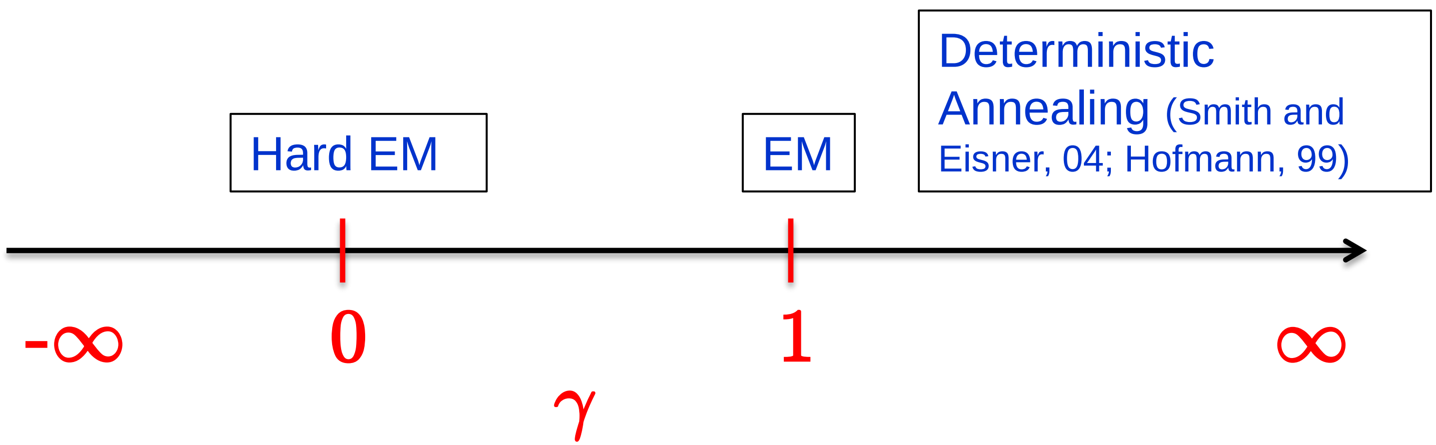


Unifying Existing EM Algorithms

$$KL(q\ ,\ p;\ \gamma) = \sum_y \gamma\ q(y)\ \log\ q(y) - q(y)\ \log\ p(y)$$

Changing γ values results in different existing EM algorithms

No Constraints

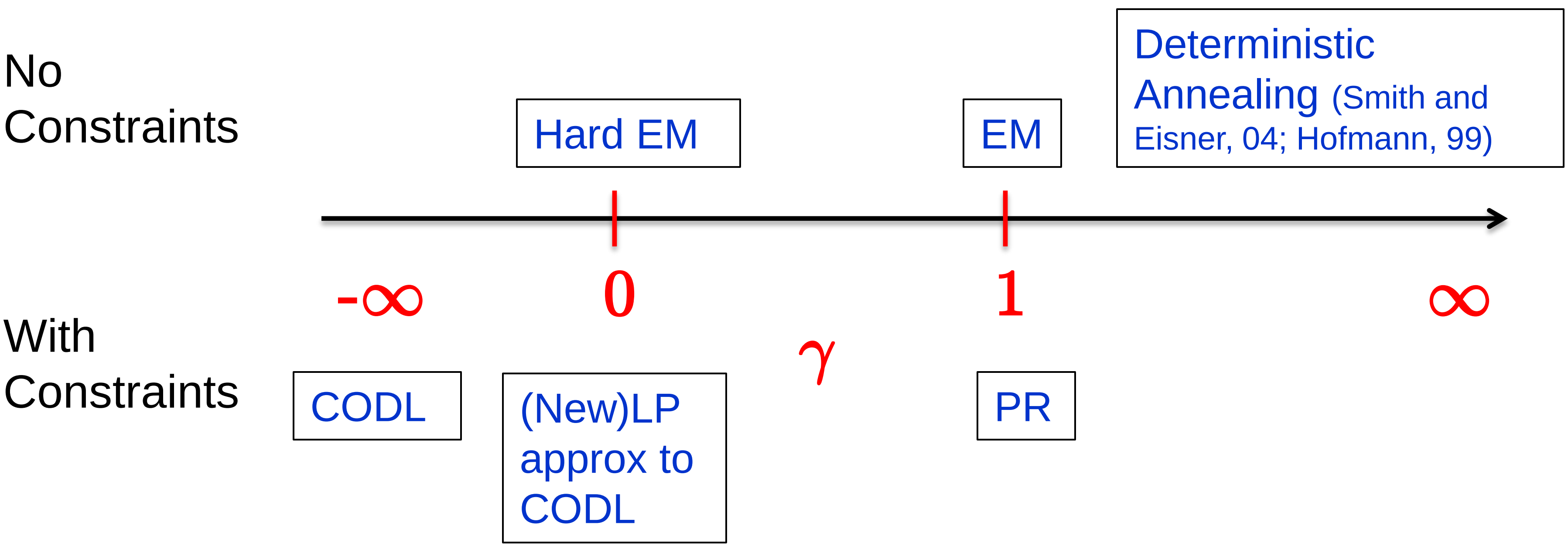


With Constraints

Unifying Existing EM Algorithms

$$KL(q\ ,\ p;\ \gamma) = \sum_y \gamma\ q(y) \log q(y) - q(y) \log p(y)$$

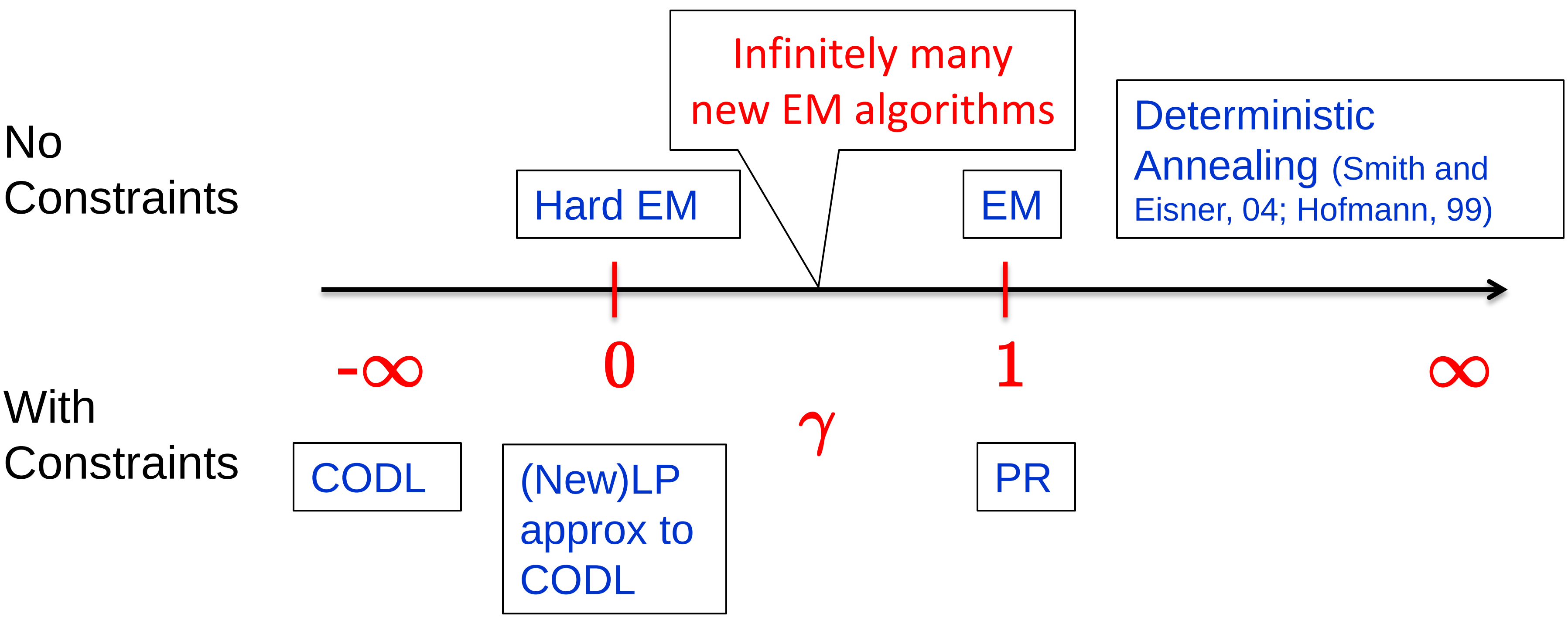
Changing γ values results in different existing EM algorithms



Unifying Existing EM Algorithms

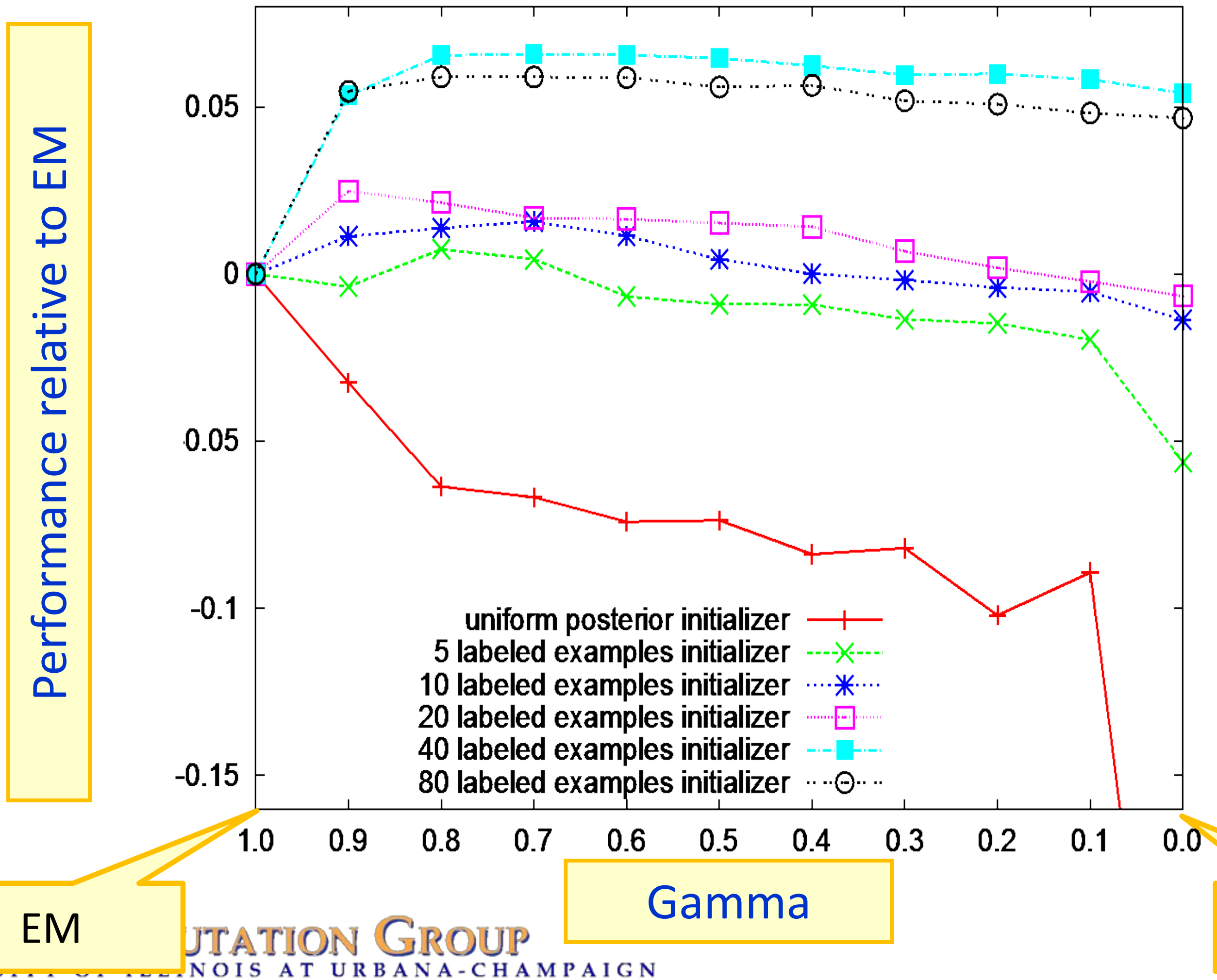
$$KL(q\ ,\ p;\ \gamma) = \sum_y \gamma\ q(y) \log q(y) - q(y) \log p(y)$$

Changing γ values results in different existing EM algorithms



Unsupervised POS tagging: Different EM instantiations

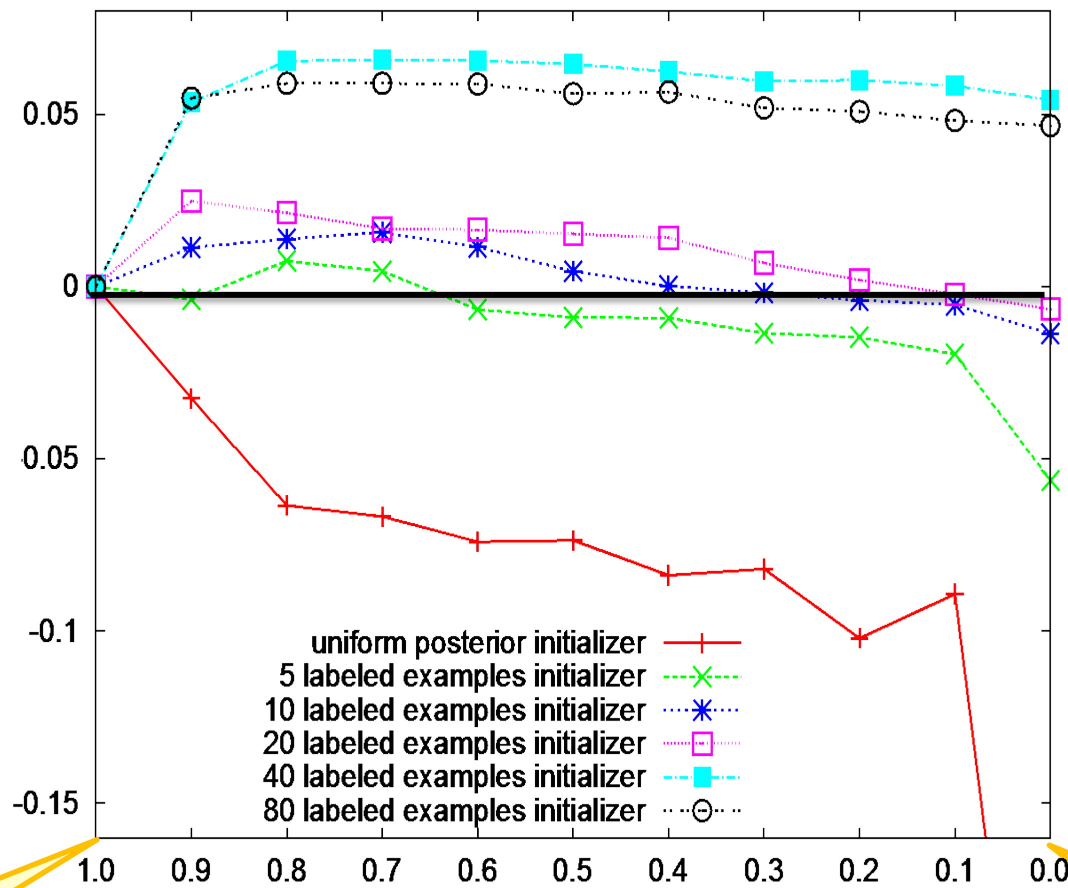
- Measure percentage accuracy relative to EM



Unsupervised POS tagging: Different EM instantiations

- Measure percentage accuracy relative to EM

Performance relative to EM



EM

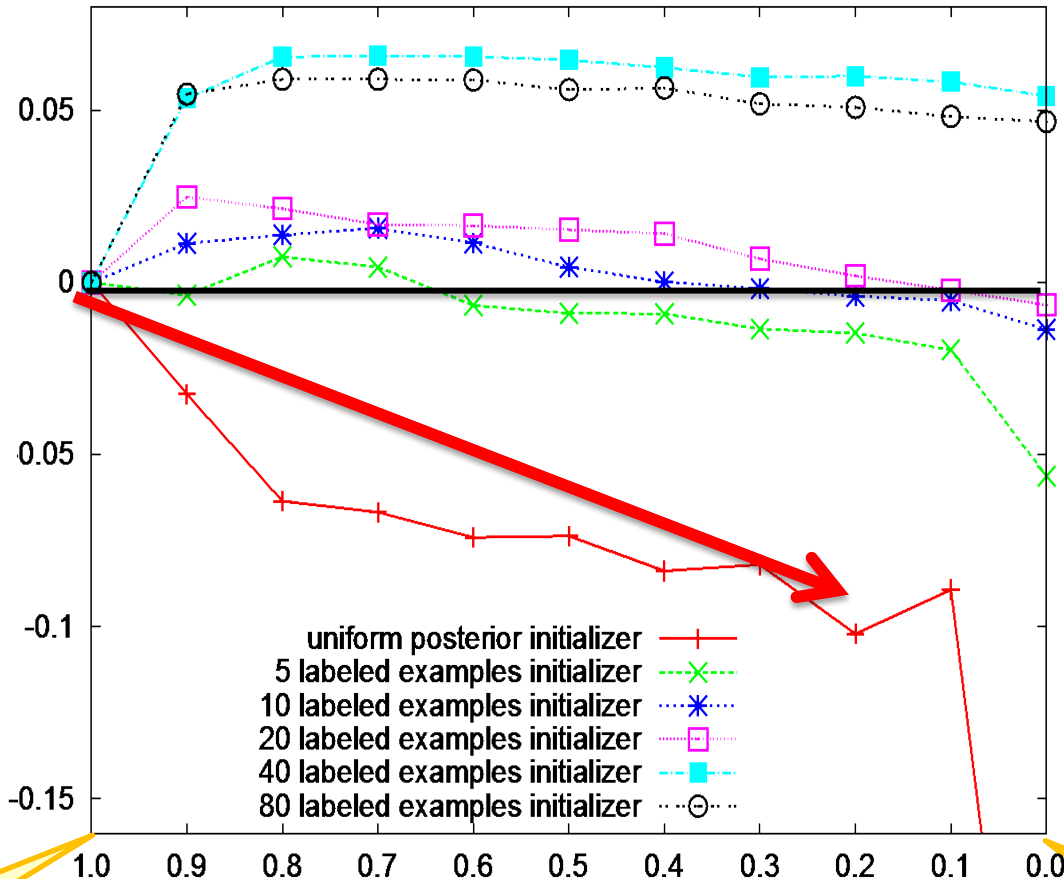
Gamma

Hard EM

Unsupervised POS tagging: Different EM instantiations

- Measure percentage accuracy relative to EM

Performance relative to EM



Uniform Initialization

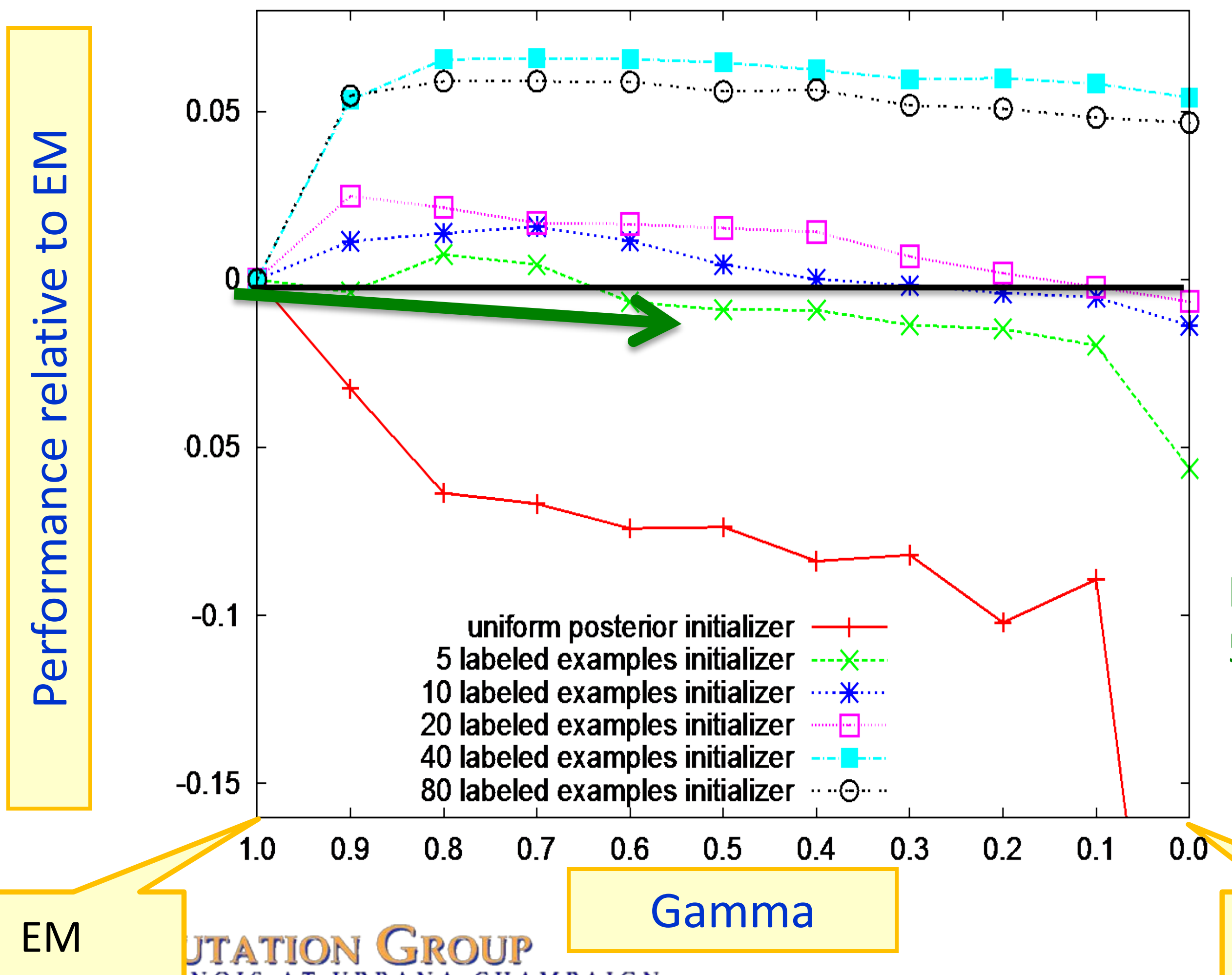
EM

Gamma

Hard EM

Unsupervised POS tagging: Different EM instantiations

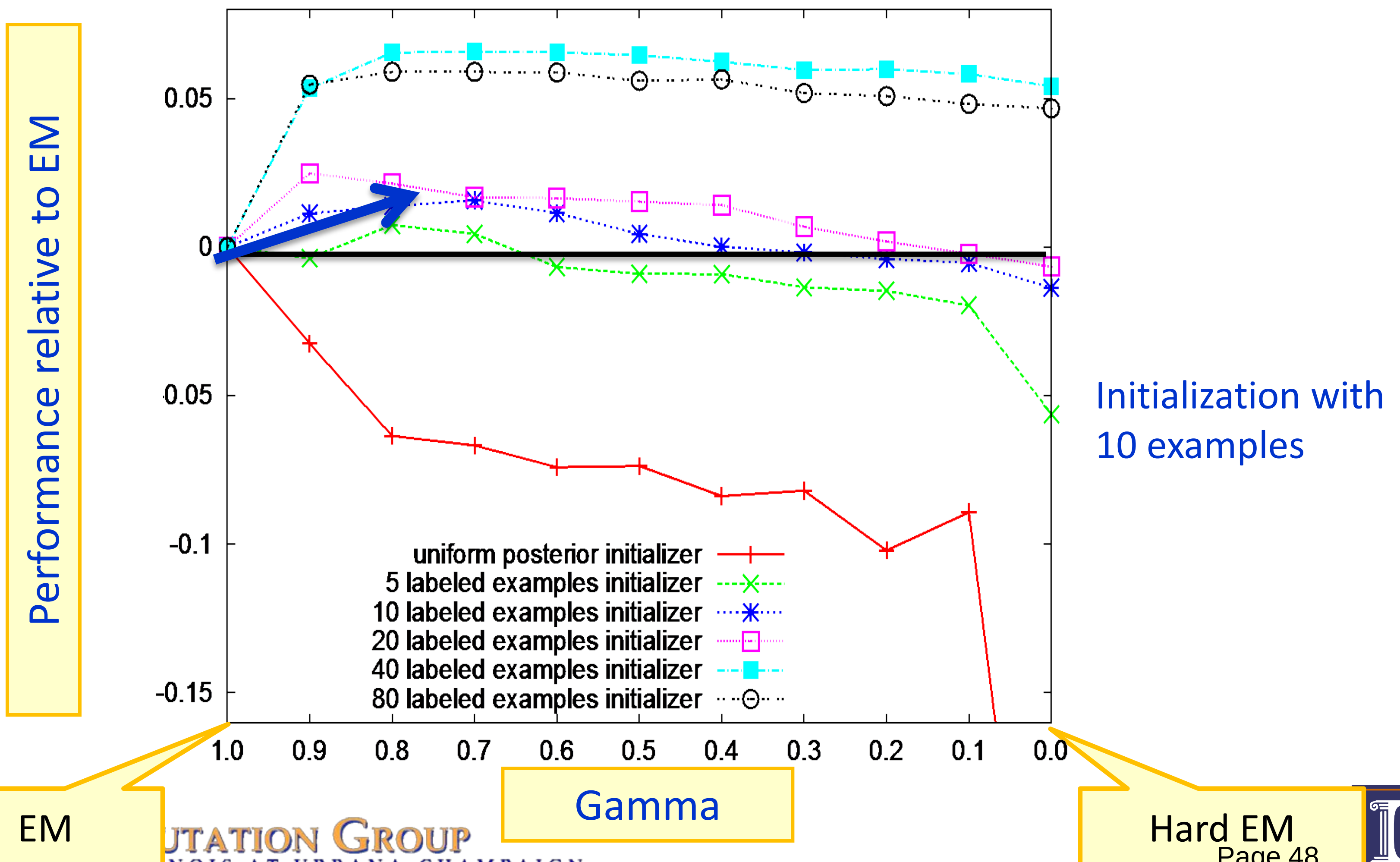
- Measure percentage accuracy relative to EM



Initialization with 5 examples

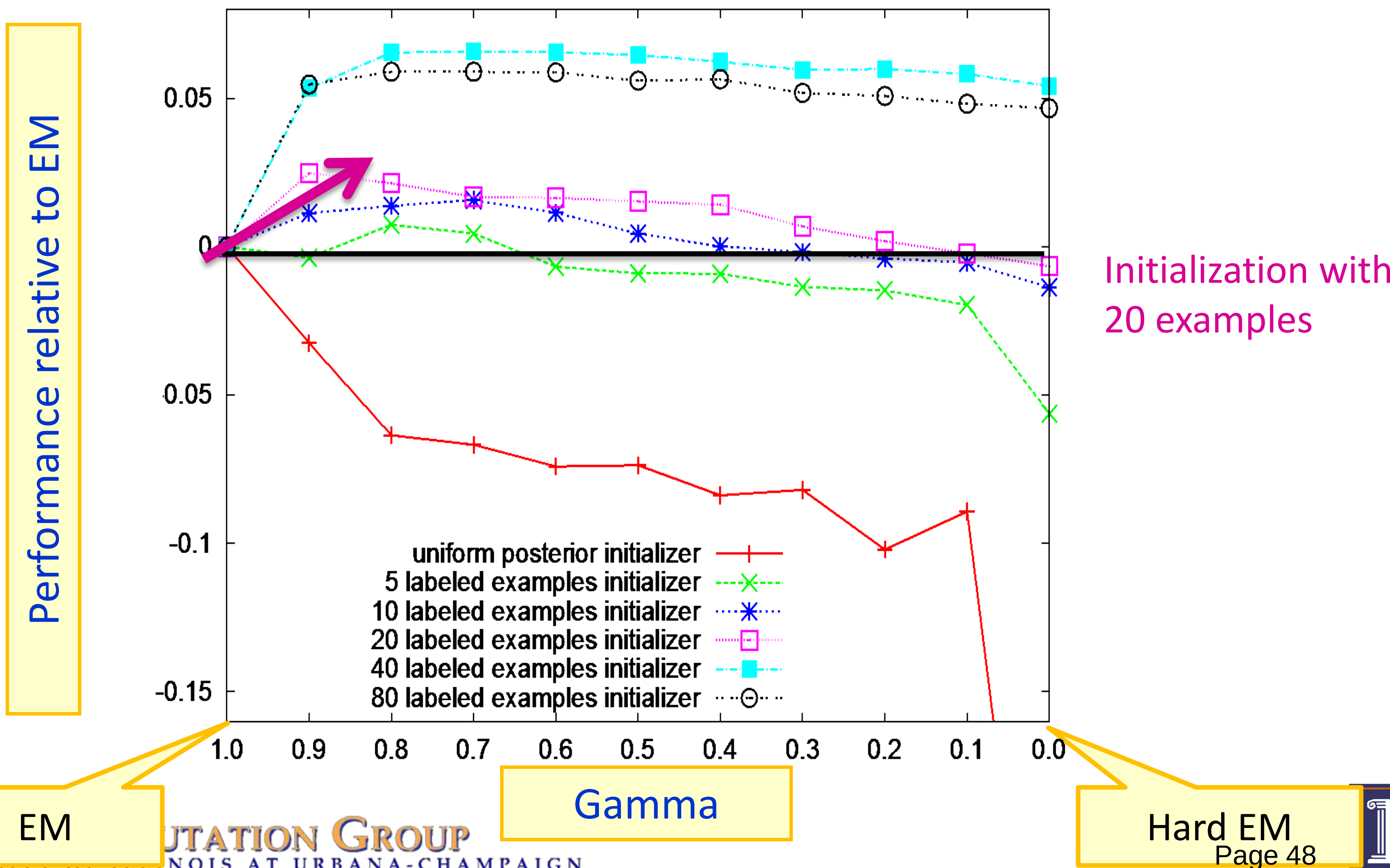
Unsupervised POS tagging: Different EM instantiations

- Measure percentage accuracy relative to EM



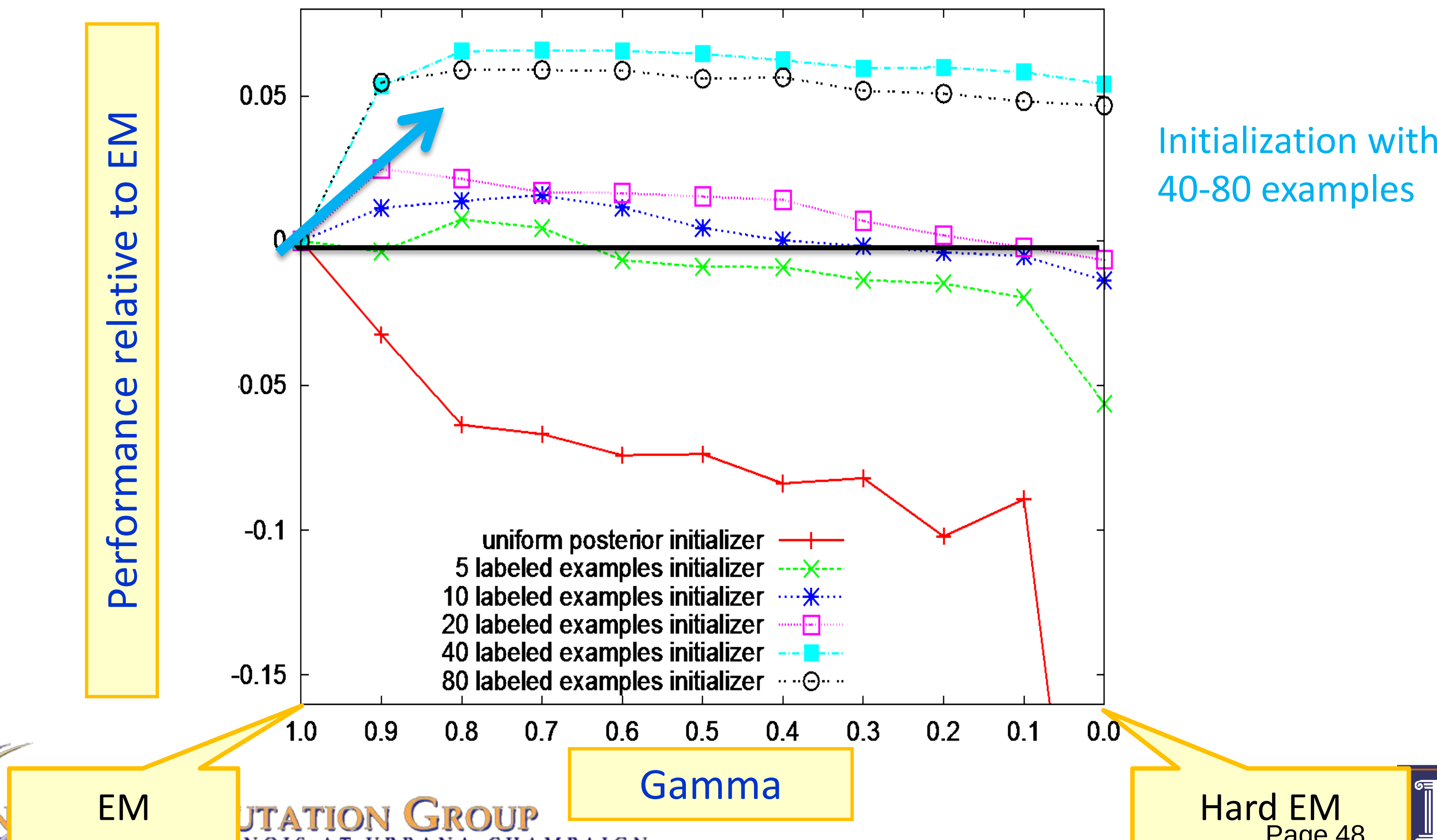
Unsupervised POS tagging: Different EM instantiations

- Measure percentage accuracy relative to EM



Unsupervised POS tagging: Different EM instantiations

- Measure percentage accuracy relative to EM



Summary: Constraints as Supervision

Summary: Constraints as Supervision

- Introducing domain knowledge-based constraints can help guiding semi-supervised learning
  - E.g. “the sentence must have at least one verb”, “a field of type y appears once in a citation”

Summary: Constraints as Supervision

- Introducing domain knowledge-based constraints can help guiding semi-supervised learning
  - E.g. “the sentence must have at least one verb”, “a field of type y appears once in a citation”
- Constrained Driven Learning (CoDL) : Constrained hard EM
- PR: Constrained soft EM
- UEM : Beyond “hard” and “soft”

Summary: Constraints as Supervision

- Introducing domain knowledge-based constraints can help guiding semi-supervised learning
  - E.g. "the sentence must have at least one verb", "a field of type y appears once in a citation"
- Constrained Driven Learning (CoDL) : Constrained hard EM
- PR: Constrained soft EM
- UEM : Beyond "hard" and "soft"
- Related literature:
  - Constraint-driven Learning (Chang et al, 07; MLJ-12),
  - Posterior Regularization (Ganchev et al, 10),
  - Generalized Expectation Criterion (Mann & McCallum, 08),
  - Learning from Measurements (Liang et al, 09)
  - **Unified EM (Samdani et al 2012: NAACL-12)**

Outline

- Constrained Conditional Models
  - A formulation for global inference with knowledge modeled as expressive structural constraints
  - Some examples
- Learning with Constrained Latent Representation
- Constraints Driven Learning
  - Training Paradigms for Constrained Conditional Models
  - Constraints Driven Learning (CoDL)
  - Unified (Constrained) Expectation Maximization
- Amortized Integer Linear Programming Inference
  - Exploiting Previous Inference Results
    - **In Inference and in Structured Learning**

Amortized ILP based Inference

- Imagine that you already solved many structured output inference problems
  - Co-reference resolution; Semantic Role Labeling; Parsing citations; Summarization; dependency parsing; image segmentation,…
  - Your solution method doesn't matter either

Amortized ILP based Inference

- Imagine that you already solved many structured output inference problems
  - Co-reference resolution; Semantic Role Labeling; Parsing citations; Summarization; dependency parsing; image segmentation,…
  - Your solution method doesn't matter either
- **How can we exploit this fact to save inference cost**?

> After solving **n** inference problems, can we make the **(n+1)**th one faster?

Amortized ILP based Inference

- Imagine that you already solved many structured output inference problems
  - Co-reference resolution; Semantic Role Labeling; Parsing citations; Summarization; dependency parsing; image segmentation,…
  - Your solution method doesn't matter either
- **How can we exploit this fact to save inference cost**?

> After solving **n** inference problems, can we make the **(n+1)**th one faster?

- We will show how to do it when your problem is formulated as a 0-1 LP, Max $\mathbf{cx}$

$$A\mathbf{x} \leq \mathbf{b}$$

Amortized ILP based Inference

- Imagine that you already solved many structured output inference problems
  - Co-reference resolution; Semantic Role Labeling; Parsing citations; Summarization; dependency parsing; image segmentation,…
  - Your solution method doesn't matter either
- **How can we exploit this fact to save inference cost**?

After solving **n** inference problems, can we make the **(n+1)**th one faster?

- We will show how to do it when your problem is formulated as a 0-1 LP, Max $\mathbf{cx}$

$$A\mathbf{x} \leq \mathbf{b}$$

- Very general: All discrete MAP problems can be formulated as 0-1 LPs
- We only care about inference formulation, not algorithmic solution

Inference for BIG TEXT

- In NLP, we typically don't solve a single inference problem.
- We solve one or more per sentence.
- Beyond improving the inference algorithm, what can be done?

Inference for BIG TEXT

- In NLP, we typically don’t solve a single inference problem.
- We solve one or more per sentence.
- Beyond improving the inference algorithm, what can be done?

S1	S2	POS
He	She	PRP
is	is	VBZ
reading	watching	VBG
a	a	DT
book	movie	NN

S1 & S2 look very different but their output structures are the same

The inference outcomes are the same

Inference for BIG TEXT

- In NLP, we typically don't solve a single inference problem.
- We solve one or more per sentence.
- Beyond improving the inference algorithm, what can be done?

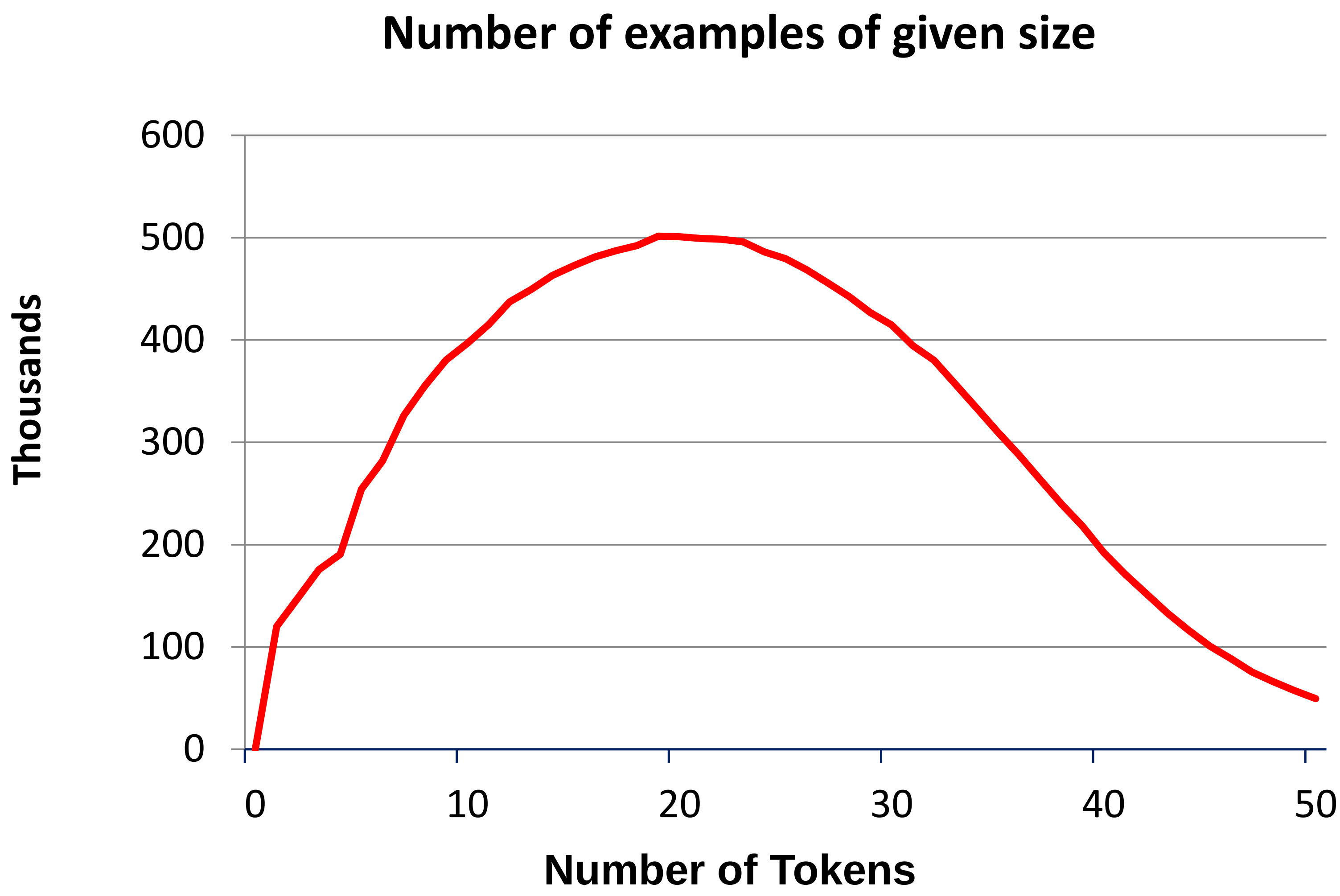
S1	S2	POS
He	She	PRP
is	is	VBZ
reading	watching	VBG
a	a	DT
book	movie	NN

S1 & S2 look very different but their output structures are the same

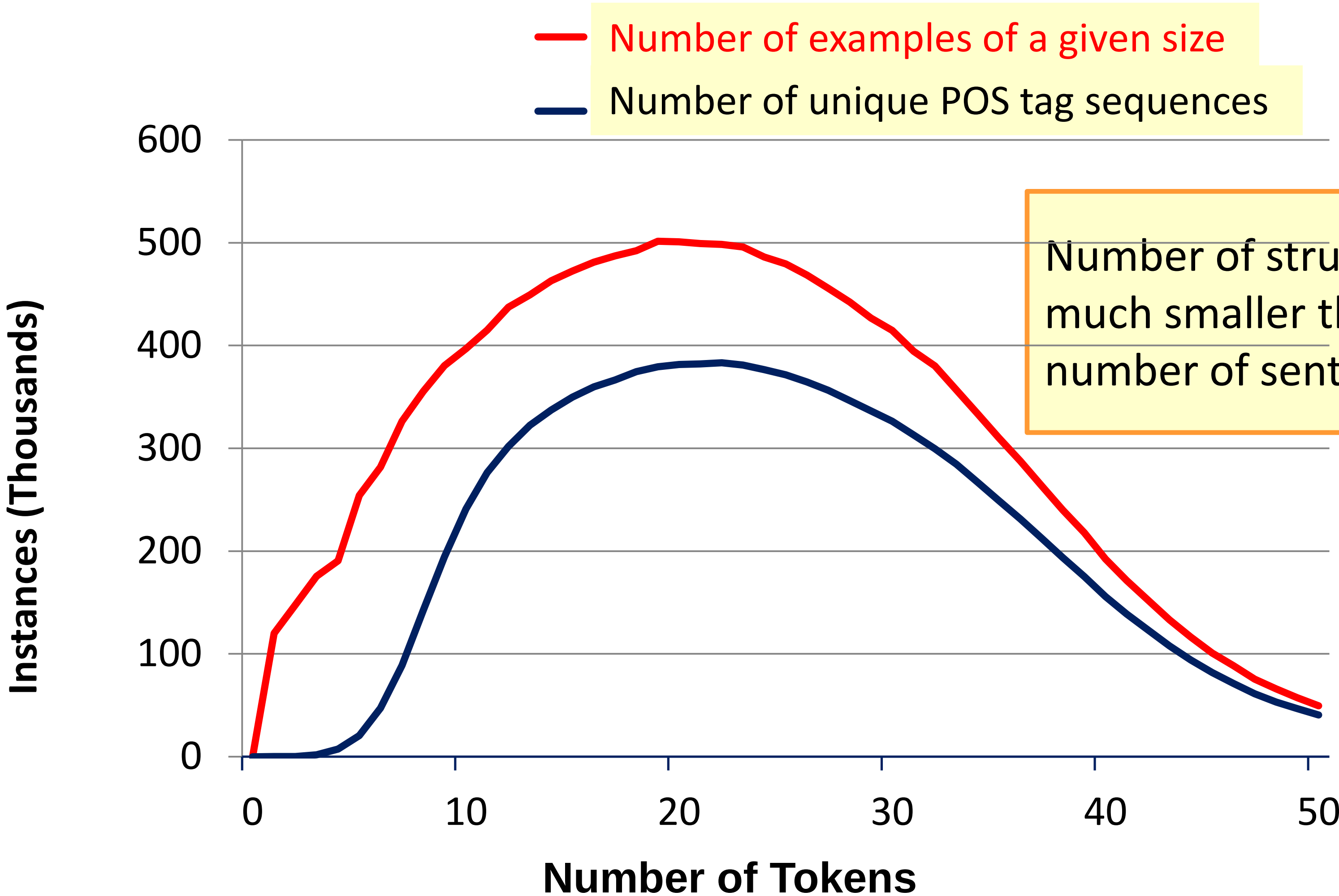
The inference outcomes are the same

After inferring the POS structure for S1,
Can we speed up inference for S2 ?

The Hope: POS Tagging on Gigaword



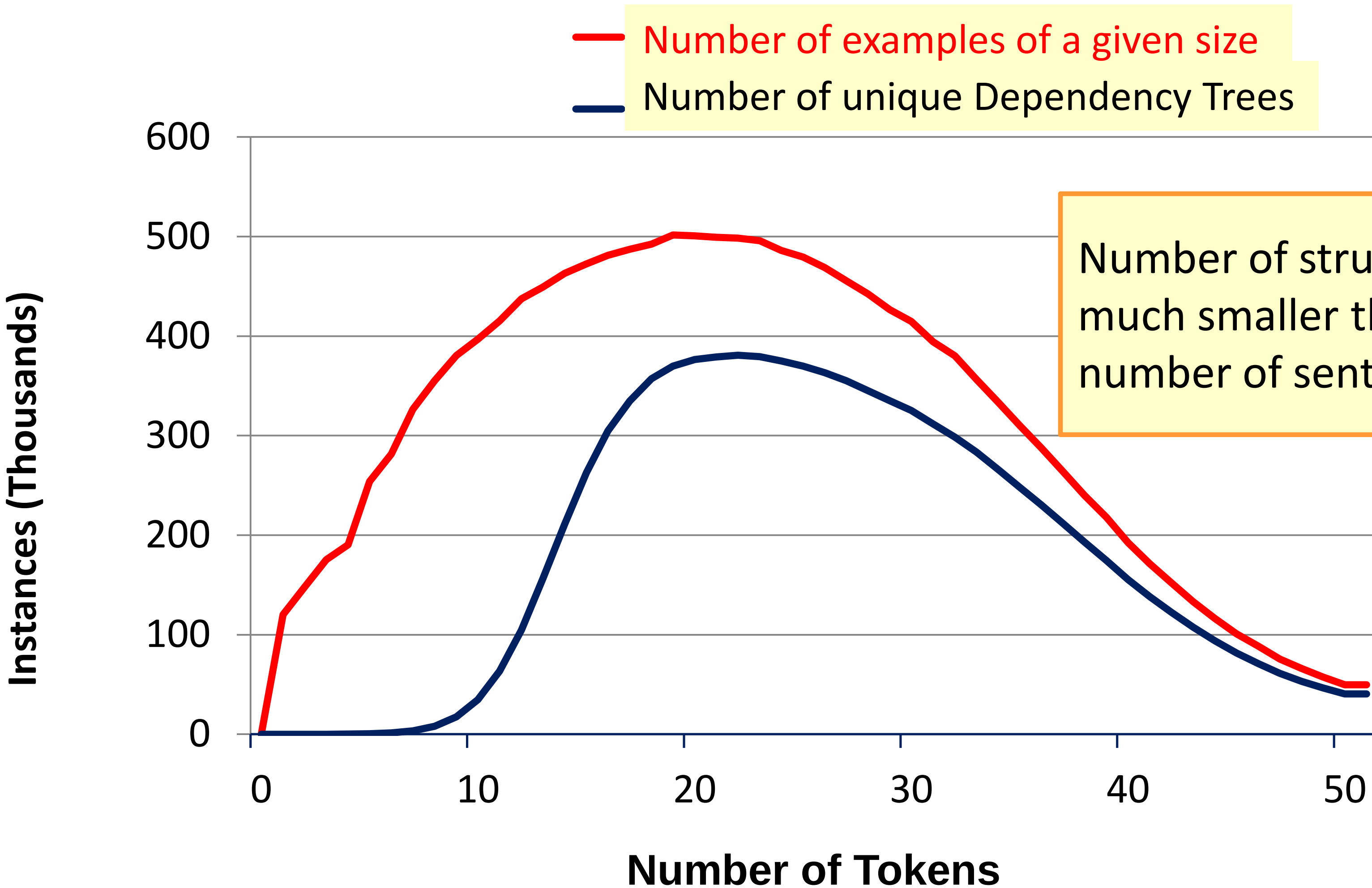
The Hope: POS Tagging on Gigaword



Number of structures is much smaller than the number of sentences

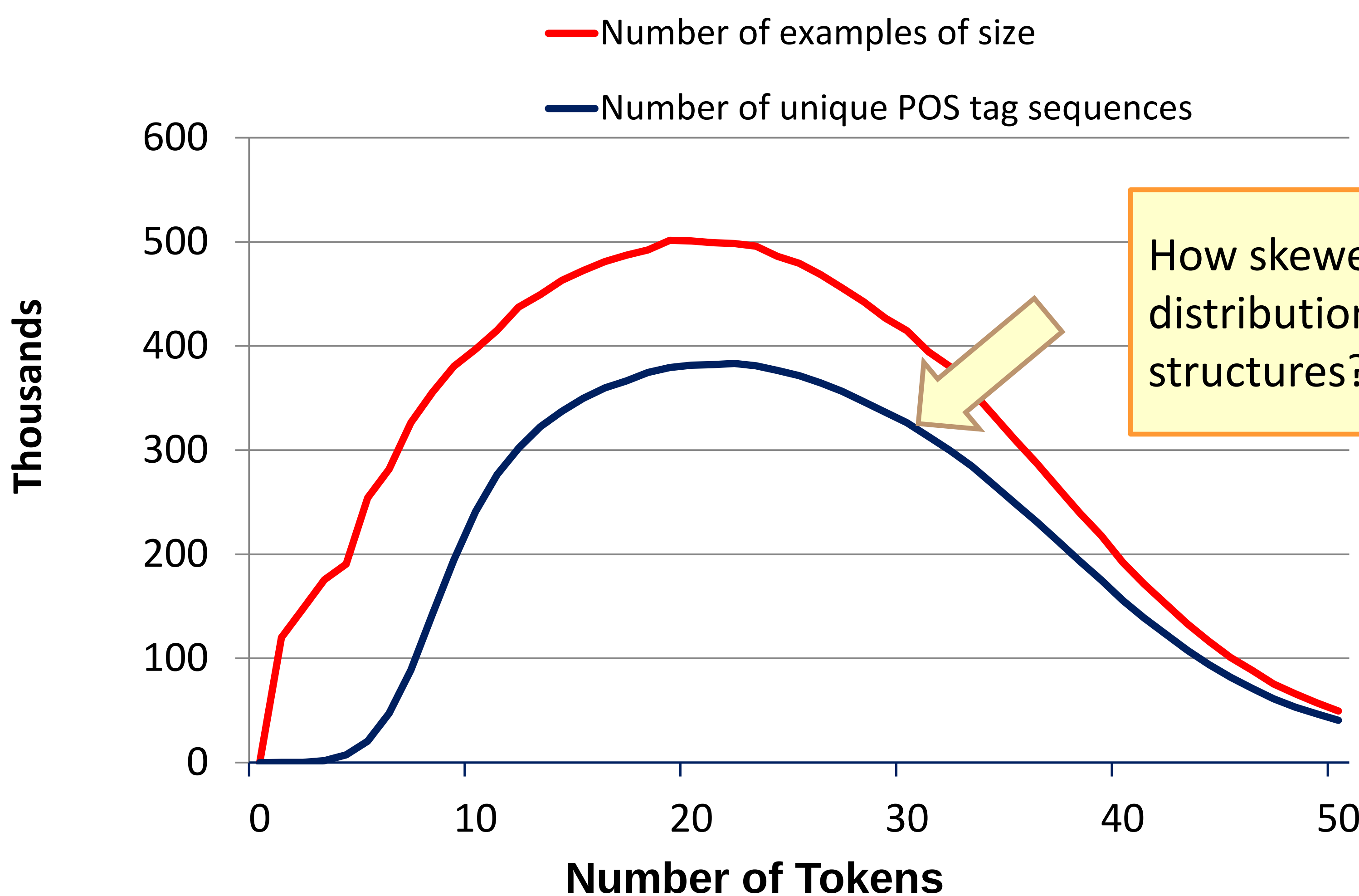
COGNITIVE COMPUTATION GROUP
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

The Hope: Dependency Parsing on Gigaword



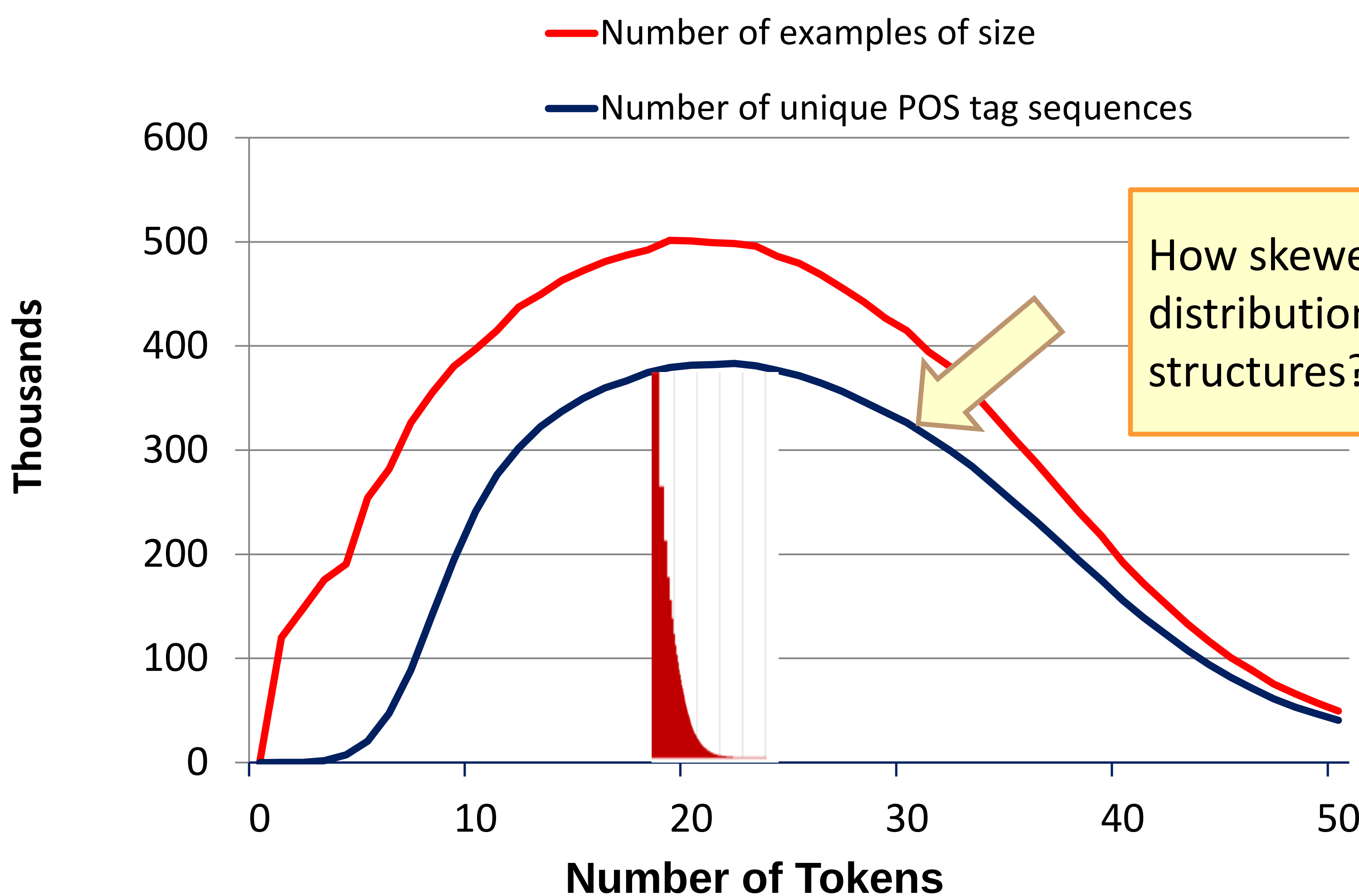
Number of structures is much smaller than the number of sentences

POS Tagging on Gigaword



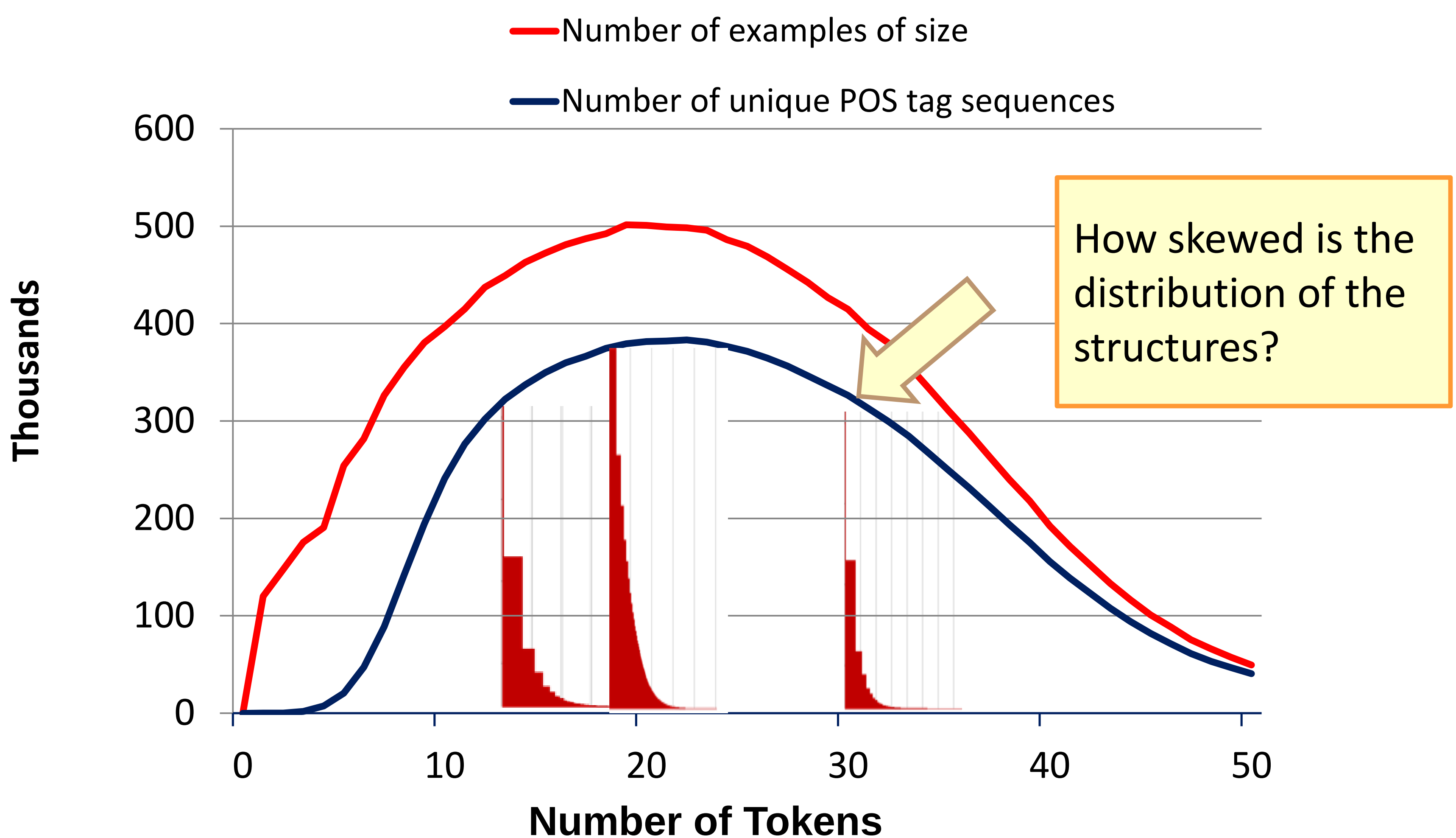
How skewed is the distribution of the structures?

POS Tagging on Gigaword

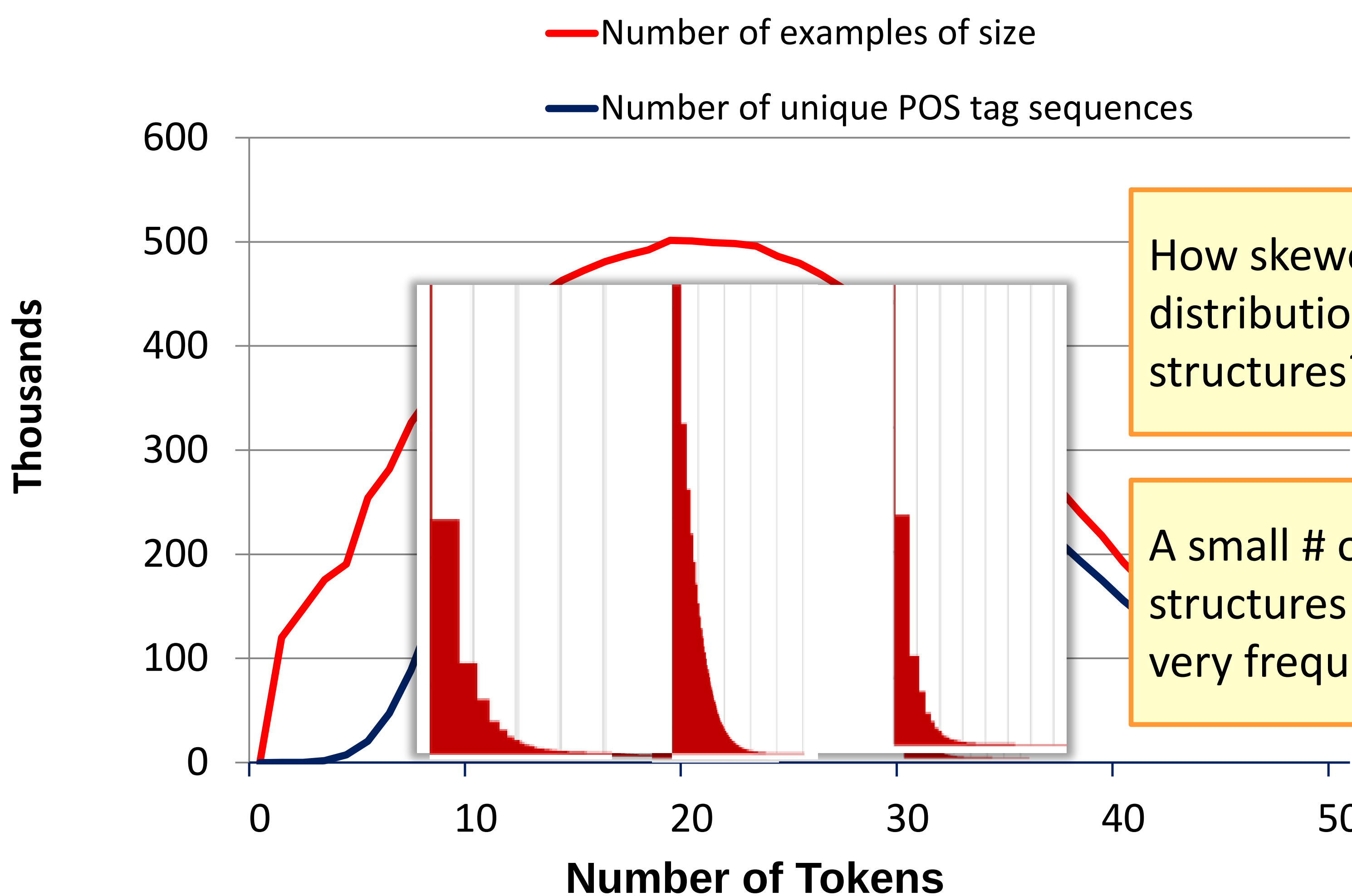


How skewed is the distribution of the structures?

POS Tagging on Gigaword



POS Tagging on Gigaword



How skewed is the distribution of the structures?

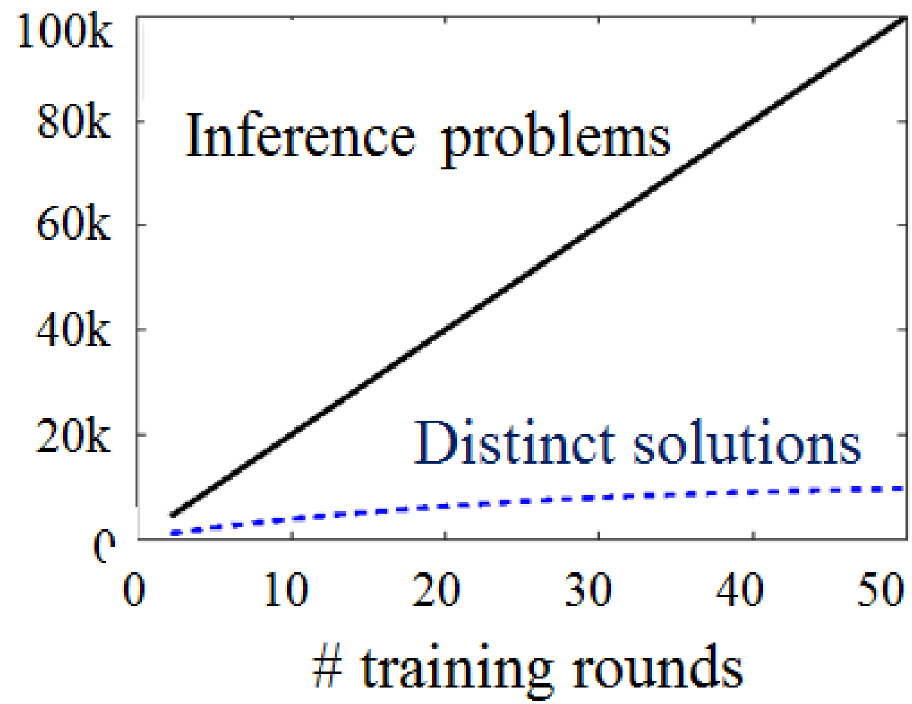
A small # of structures occur very frequently

Redundancy in Inference and Learning

- This redundancy is clearly important since in all NLP tasks there is a need to solve many inferences, at least one per sentence.
- However, it is as important in structured learning, where algorithms cycle between
- performing inference and
- updating the model.

Redundancy in Inference and Learning

- This redundancy is clearly important since in all NLP tasks there is a need to solve many inferences, at least one per sentence.
- However, it is as important in structured learning, where algorithms cycle between
- performing inference and
- updating the model.



Amortized ILP Inference

- These statistics show that many different instances are mapped into identical inference outcomes.
  - Pigeon Hole Principle

Amortized ILP Inference

- These statistics show that many different instances are mapped into identical inference outcomes.
  - Pigeon Hole Principle
- **How can we exploit this fact to save inference cost over the life time of the agent?**

Amortized ILP Inference

- These statistics show that many different instances are mapped into identical inference outcomes.
  - Pigeon Hole Principle
- **How can we exploit this fact to save inference cost over the life time of the agent?**

We give conditions on the objective functions
(for all objectives with the same # or variables and same feasible set),
under which the solution of a new problem Q is the same as the one of P (which we already cached)

Amortized ILP Inference

We argue here that the inference formulation provides a new level of abstraction.

- These statistics show that many different instances are mapped into identical inference outcomes.
  - Pigeon Hole Principle
- **How can we exploit this fact to save inference cost over the life time of the agent?**

We give conditions on the objective functions
(for all objectives with the same # or variables and same feasible set),
under which the solution of a new problem Q is the same as the one of P (which we already cached)

Amortized ILP Inference

We argue here that the inference formulation provides a new level of abstraction.

- These statistics show that many different instances are mapped into identical inference outcomes.
  - Pigeon Hole Principle
- **How can we exploit this fact to save inference cost over the life time of the agent?**

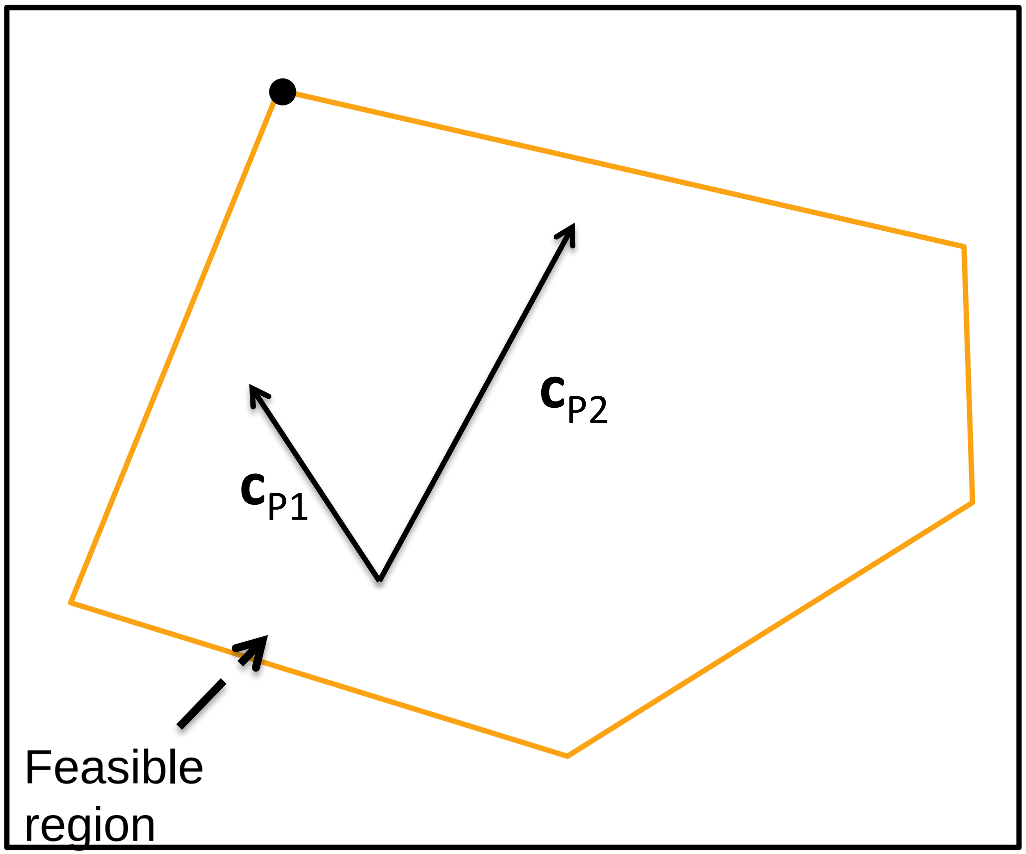
We give conditions on the objective functions
(for all objectives with the same # or variables and same feasible set),
under which the solution of a new problem Q is the same as the one of P (which we already cached)

If **CONDITION** (problem ***cache***, *new problem*)
then (no need to call the solver)
SOLUTION(*new problem*) = old solution
Else
Call **base solver** and update ***cache***
End

0.04 ms

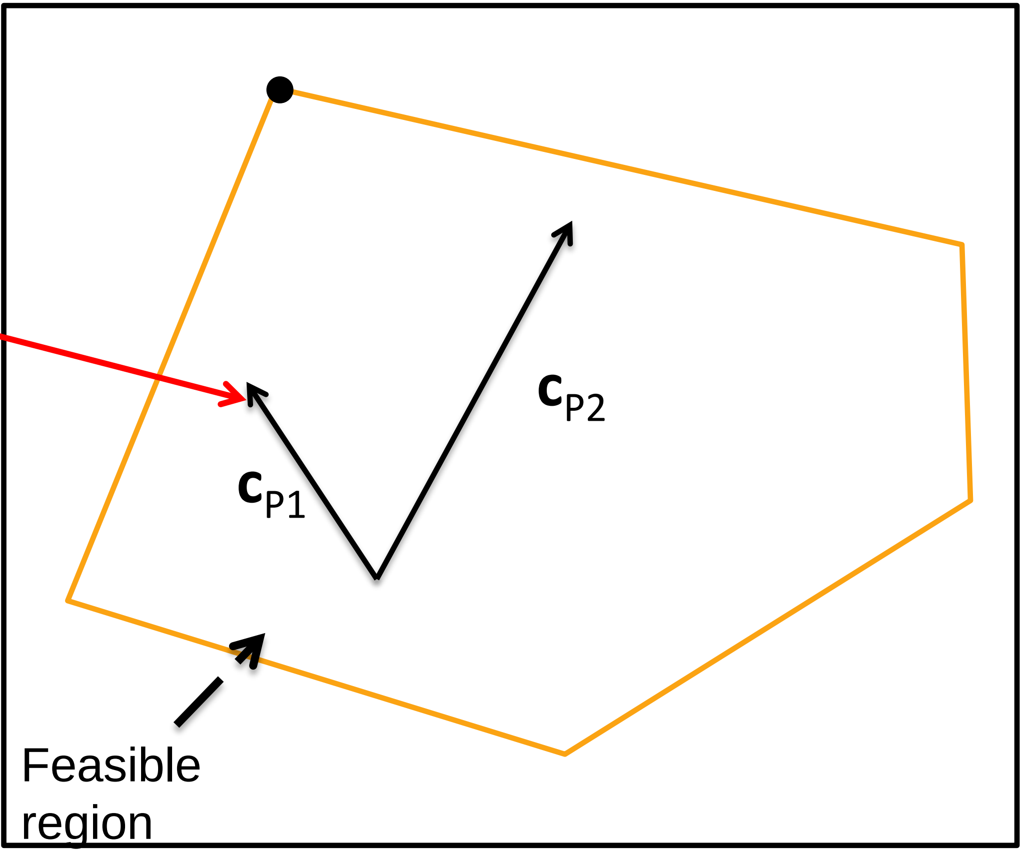
2 ms

Theorem II (Geometric Interpretation)



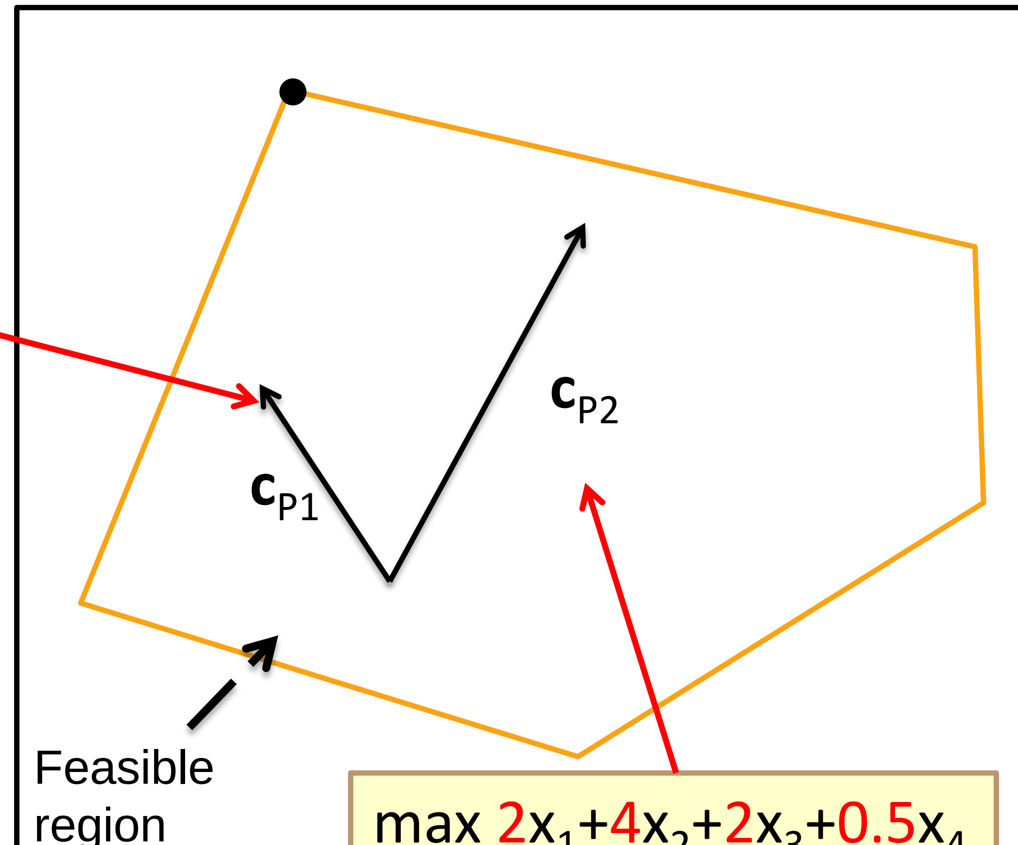
Theorem II (Geometric Interpretation)

$$\max\ 2x_1+3x_2+2x_3+1x_4$$
$$x_1 + x_2 \leq 1$$
$$x_3 + x_4 \leq 1$$



Theorem II (Geometric Interpretation)

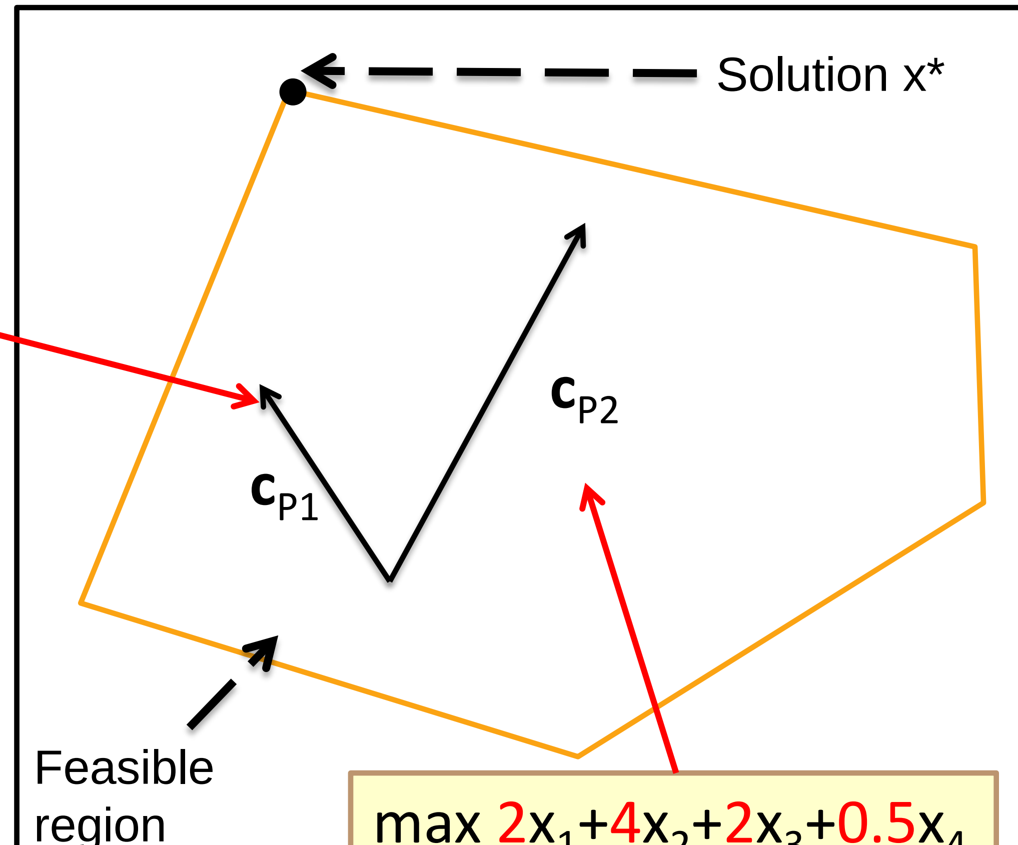
$$\max\ 2x_1+3x_2+2x_3+1x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$



$$\max\ 2x_1+4x_2+2x_3+0.5x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

Theorem II (Geometric Interpretation)

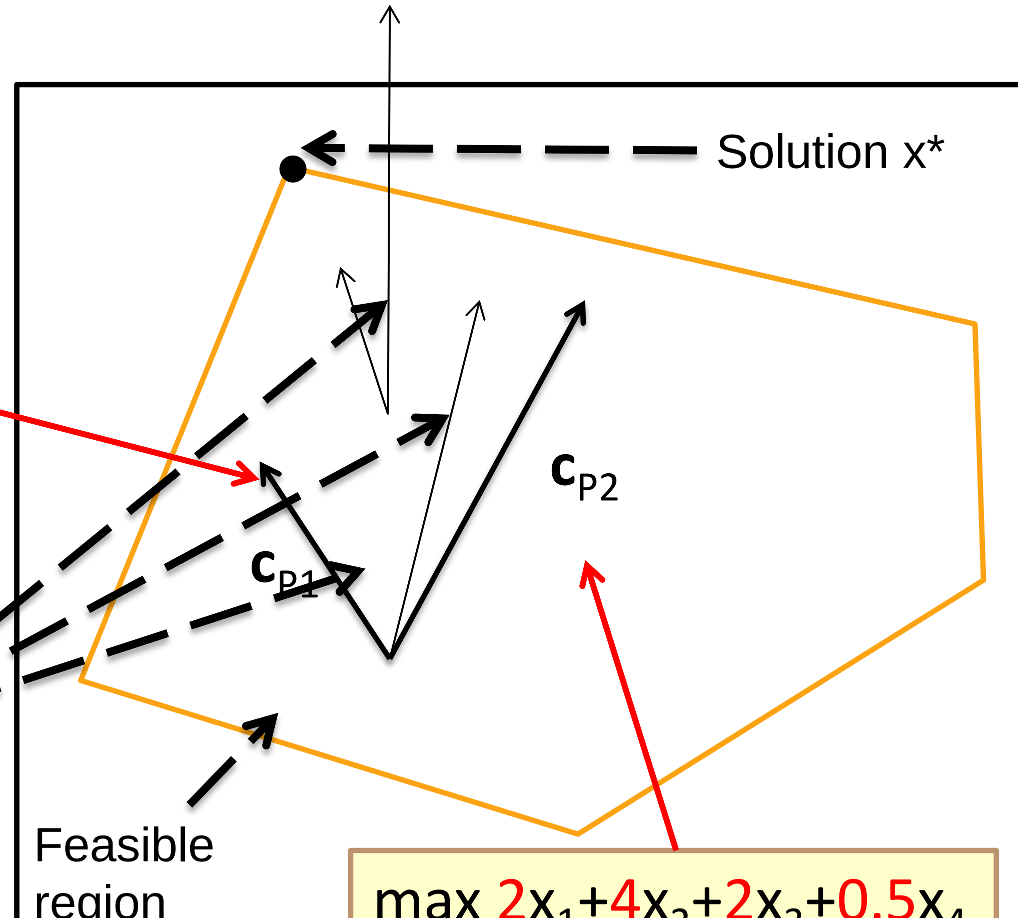
$$\max\ 2x_1+3x_2+2x_3+1x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$



$$\max\ 2x_1+4x_2+2x_3+0.5x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

Theorem II (Geometric Interpretation)

$$\max\ 2x_1+3x_2+2x_3+1x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$



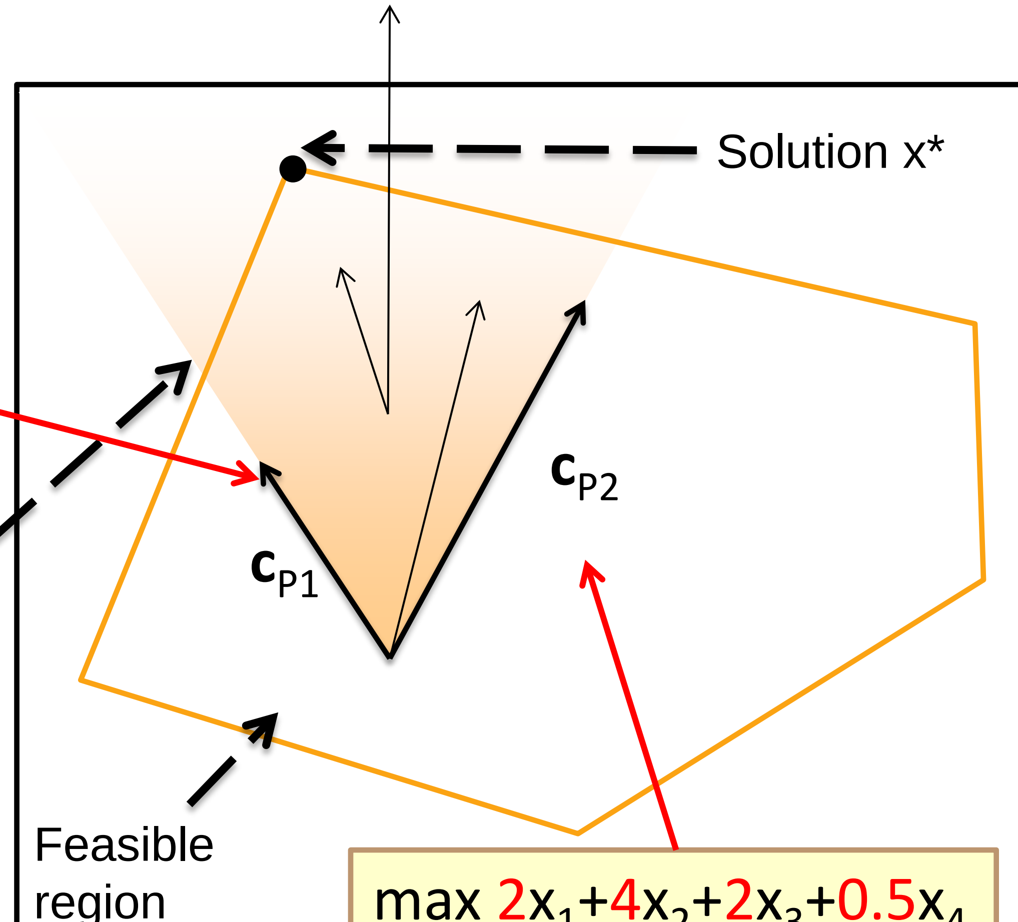
ILPs corresponding to all these objective vectors will share the same maximizer *for this feasible region*

$$\max\ 2x_1+4x_2+2x_3+0.5x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

Theorem II (Geometric Interpretation)

$$\max\ 2x_1+3x_2+2x_3+1x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

All ILPs in the ***cone*** will share the maximizer



$$\max\ 2x_1+4x_2+2x_3+0.5x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

Theorem I

P

max $2x_1+3x_2+2x_3+x_4$
$x_1 + x_2 \leq 1$
$x_3 + x_4 \leq 1$

Q

max $2x_1+4x_2+2x_3+0.5x_4$
$x_1 + x_2 \leq 1$
$x_3 + x_4 \leq 1$

$\mathbf{x}^*_P$: <0, **1, 1,** 0>

$\mathbf{c}_P$: <2, **3, 2,** 1>

$\mathbf{c}_Q$: <2, **4, 2,** 0.5>

Theorem I

P

max $2x_1+3x_2+2x_3+x_4$

$x_1 + x_2 \leq 1$

$x_3 + x_4 \leq 1$

Q

max $2x_1+4x_2+2x_3+0.5x_4$

$x_1 + x_2 \leq 1$

$x_3 + x_4 \leq 1$

If

The objective coefficients of active variables **did not decrease** from P to Q

x^*_P: <0, **1, 1,** 0>

c_P: <2, **3, 2,** 1>

$\Downarrow \quad \Downarrow$

c_Q: <2, **4, 2,** 0.5>

Theorem I

P

$$\max\ 2x_1+3x_2+2x_3+x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

Q

$$\max\ 2x_1+4x_2+2x_3+0.5x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

If

The objective coefficients of active variables **did not decrease** from P to Q

$\mathbf{x}^*_P$: <**0**, 1, 1, **0**>

$\mathbf{c}_P$: <**2**, 3, 2, **1**>

$\mathbf{c}_Q$: <**2**, 4, 2, **0.5**>

And

The objective coefficients of inactive variables **did not increase** from P to Q

Theorem I

P

$$\max\ 2x_1+3x_2+2x_3+x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

Q

$$\max\ 2x_1+4x_2+2x_3+0.5x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

If

The objective coefficients of active variables **did not decrease** from P to Q

x^*_P: <**0**, 1, 1, **0**>

c_P: <**2**, 3, 2, **1**>

c_Q: <**2**, 4, 2, **0.5**>

And

The objective coefficients of inactive variables **did not increase** from P to Q

Theorem I

Then: The optimal solution of Q is the same as P's

$\mathbf{x}^*_P = \mathbf{x}^*_Q$

P

$$\max\ 2x_1+3x_2+2x_3+x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

Q

$$\max\ 2x_1+4x_2+2x_3+0.5x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

If

The objective coefficients of active variables **did not decrease** from P to Q

$\mathbf{x}^*_P$: <**0**, 1, 1, **0**>

$\mathbf{c}_P$: <**2**, 3, 2, **1**>

$\mathbf{c}_Q$: <**2**, 4, 2, **0.5**>

And

The objective coefficients of inactive variables **did not increase** from P to Q

Theorem I

Then: The optimal solution of Q is the same as P's

$\mathbf{x}^*_P = \mathbf{x}^*_Q$

P

$$\max\ 2x_1+3x_2+2x_3+x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

Q

$$\max\ 2x_1+4x_2+2x_3+0.5x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

If

The objective coefficients of active variables **did not decrease** from P to Q

$\mathbf{x}^*_P$: <**0**, 1, 1, **0**>

$\mathbf{c}_P$: <**2**, 3, 2, **1**>

$\mathbf{c}_Q$: <**2**, 4, 2, **0.5**>

And

The objective coefficients of inactive variables **did not increase** from P to Q

$$\forall i, \left(2\boldsymbol{y}^*_{p,i} - 1\right)\left(c_{Q,i} - c_{P,i}\right) \ge 0$$

Theorem I

Then: The optimal solution of Q is the same as P's

$x^*_P = x^*_Q$

P

max $2x_1+3x_2+2x_3+x_4$
$x_1 + x_2 \leq 1$
$x_3 + x_4 \leq 1$

Q

max $2x_1+4x_2+2x_3+0.5x_4$
$x_1 + x_2 \leq 1$
$x_3 + x_4 \leq 1$

If

The objective coefficients of active variables **did not decrease** from P to Q

x^*_P: <**0**, 1, 1, **0**>

c_P: <**2**, 3, 2, **1**>

c_Q: <**2**, 4, 2, **0.5**>

And

The objective coefficients of inactive variables **did not increase** from P to Q

$$\forall i, \left(2\boldsymbol{y}^*_{p,i} - 1\right)(c_{Q,i} - c_{P,i}) \geq 0$$

$$\forall i, \left(2\boldsymbol{y}^*_{p,i} - 1\right)(c_{Q,i} - c_{P,i}) \geq -\epsilon |c_{Q,i}|$$

Theorem I

Then: The optimal solution of Q is the same as P's

Structured Learning: Dual coordinate descent for structured SVM still returns an exact model even if approx. amortized inference is used.

P

$$\max\ 2x_1+3x_2+2x_3+x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

$\mathbf{x}^*_P = \mathbf{x}^*_Q$

$$\max\ 2x_1+4x_2+2x_3+0.5x_4$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$

If

The objective coefficients of active variables **did not decrease** from P to Q

$\mathbf{x}^*_P$: <**0**, 1, 1, **0**>

$\mathbf{c}_P$: <**2**, 3, 2, **1**>

$\mathbf{c}_Q$: <**2**, 4, 2, **0.5**>

And

The objective coefficients of inactive variables **did not increase** from P to Q

$$\forall i, \left(2\boldsymbol{y}^*_{p,i} - 1\right)(c_{Q,i} - c_{P,i}) \ge 0$$

$$\forall i, \left(2\boldsymbol{y}^*_{p,i} - 1\right)(c_{Q,i} - c_{P,i}) \ge -\epsilon |c_{Q,i}|$$

Amortized Inference Experiments

- Setup
  - Verb semantic role labeling; Entity and Relations
  - Speedup & Accuracy are measured over WSJ test set (Section 23) and Test of E & R
  - Baseline: solving ILPs using the Gurobi solver.

Amortized Inference Experiments

- Setup
  - Verb semantic role labeling; Entity and Relations
  - Speedup & Accuracy are measured over WSJ test set (Section 23) and Test of E & R
  - Baseline: solving ILPs using the Gurobi solver.

- For amortization
  - Cache 250,000 inference problems (objective, solution) from Gigaword
  - For each problem in test set either call the inference engine or re-use a solution from the cache, if our theorems hold.

Amortized Inference Experiments

- Setup
  - Verb semantic role labeling; Entity and Relations
  - Speedup & Accuracy are measured over WSJ test set (Section 23) and Test of E & R
  - Baseline: solving ILPs using the Gurobi solver.

- For amortization
  - Cache 250,000 inference problems (objective, solution) from Gigaword
  - For each problem in test set either call the inference engine or re-use a solution from the cache, if our theorems hold.

No training data is needed for this method.
Once you have a model, you can generate a large cache that will be then used to save you time at evaluation time.

Speedup & Accuracy

$$\text{Speedup} = \frac{\text{number of inference calls without amortization}}{\text{number of inference calls with amortization}}$$

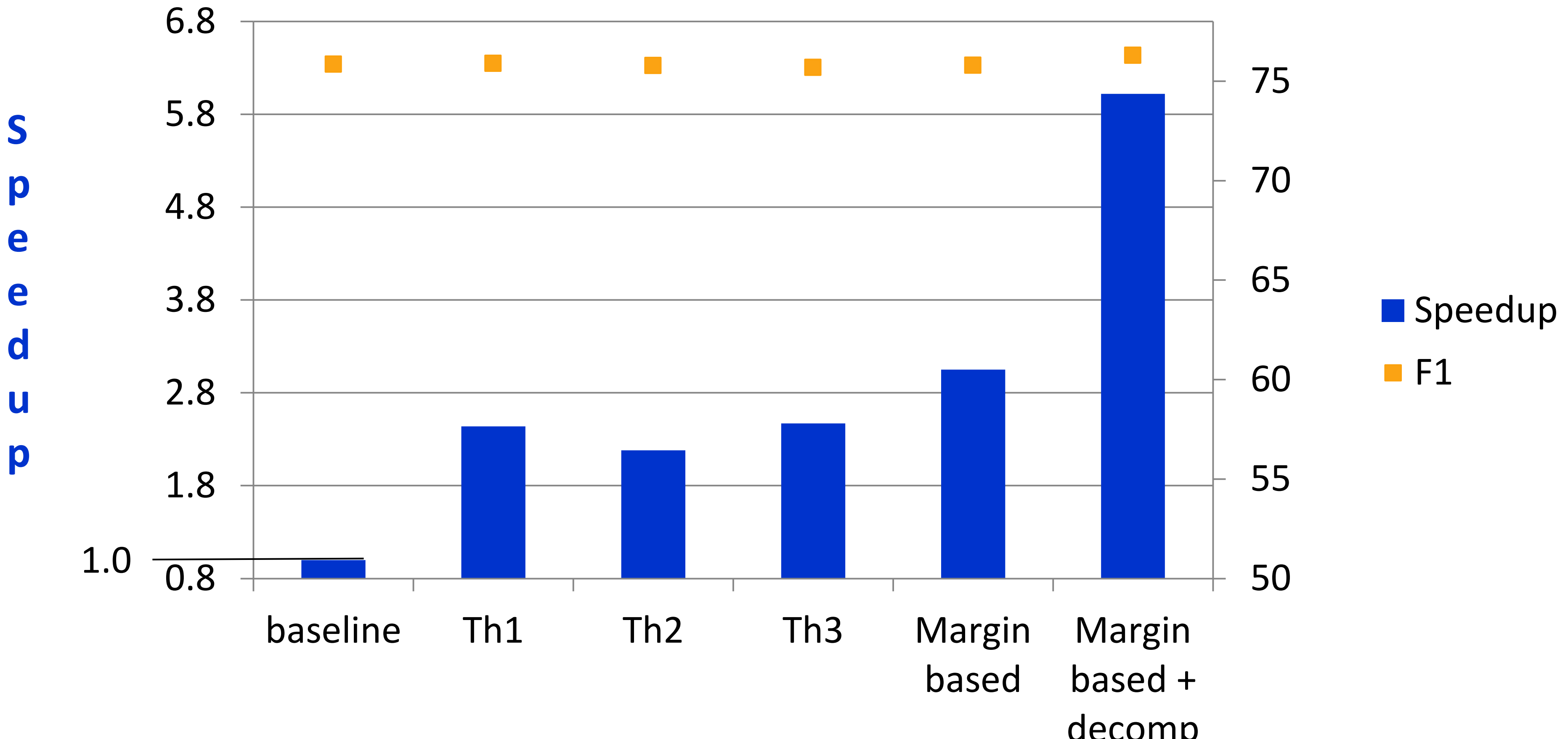
S
p
e
e
d
u
p

Amortization schemes [EMNLP'12, ACL'13]

Speedup & Accuracy

By decomposing the objective function, building on the fact that "smaller structures" are more redundant, it is possible to get even better results.

$$\text{Speedup} = \frac{\text{number of inference calls without amortization}}{\text{number of inference calls with amortization}}$$

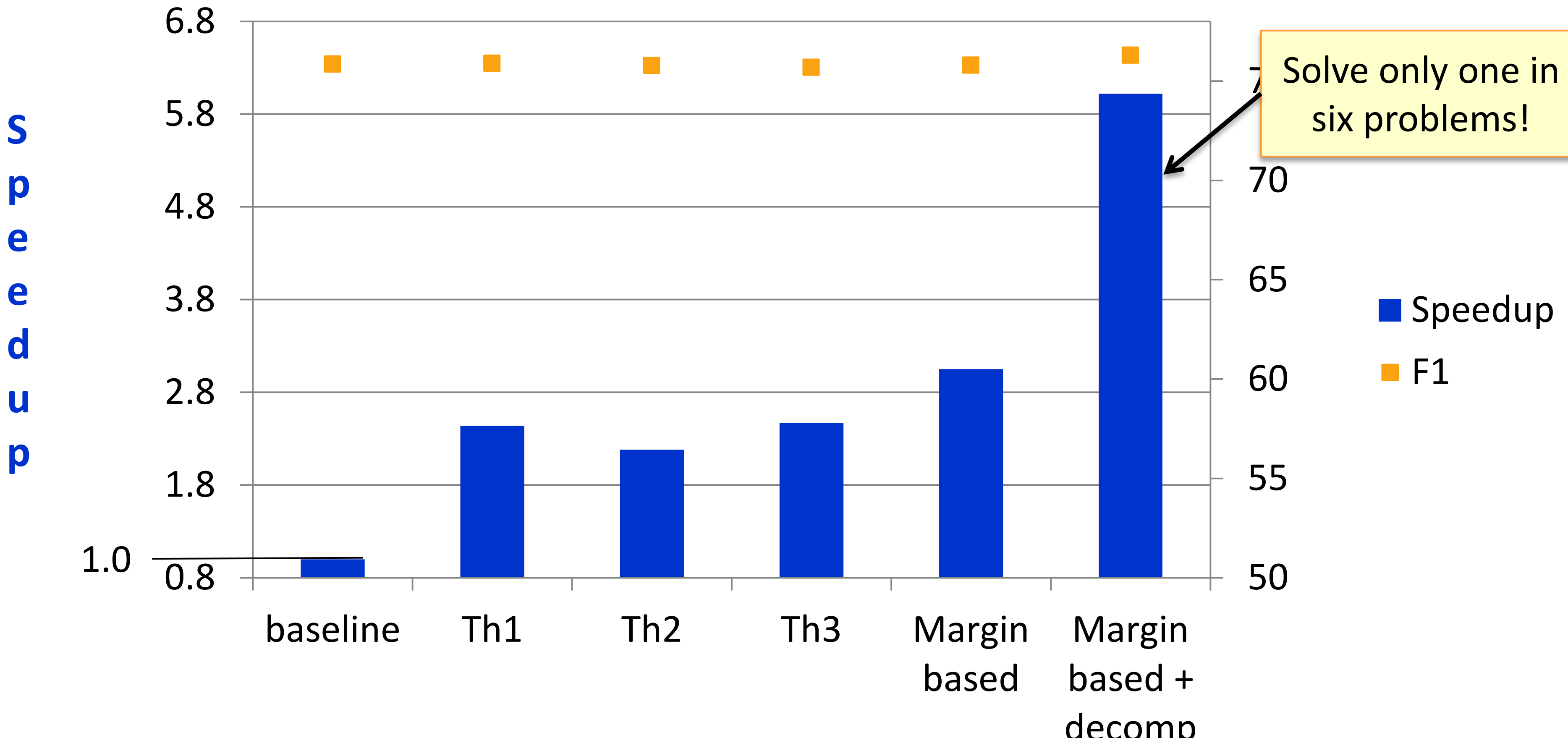


Amortization schemes [EMNLP'12, ACL'13]

Speedup & Accuracy

By decomposing the objective function, building on the fact that "smaller structures" are more redundant, it is possible to get even better results.

$$\text{Speedup} = \frac{\text{number of inference calls without amortization}}{\text{number of inference calls with amortization}}$$

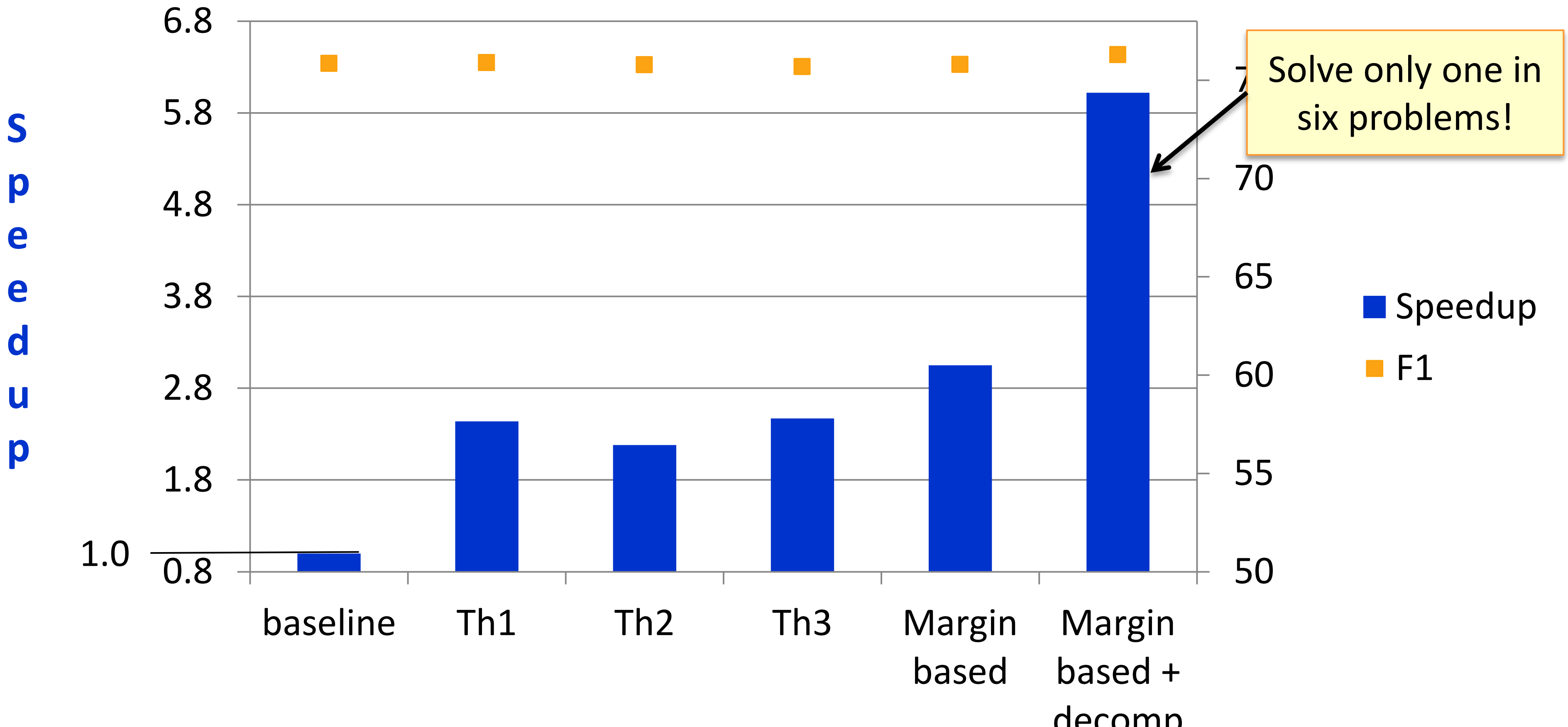


Amortization schemes [EMNLP'12, ACL'13]

Speedup & Accuracy

The results show that, indeed, the inference formulation provides a new level of abstraction that can be exploited to re-use solutions

$$\text{Speedup} = \frac{\text{number of inference calls without amortization}}{\text{number of inference calls with amortization}}$$



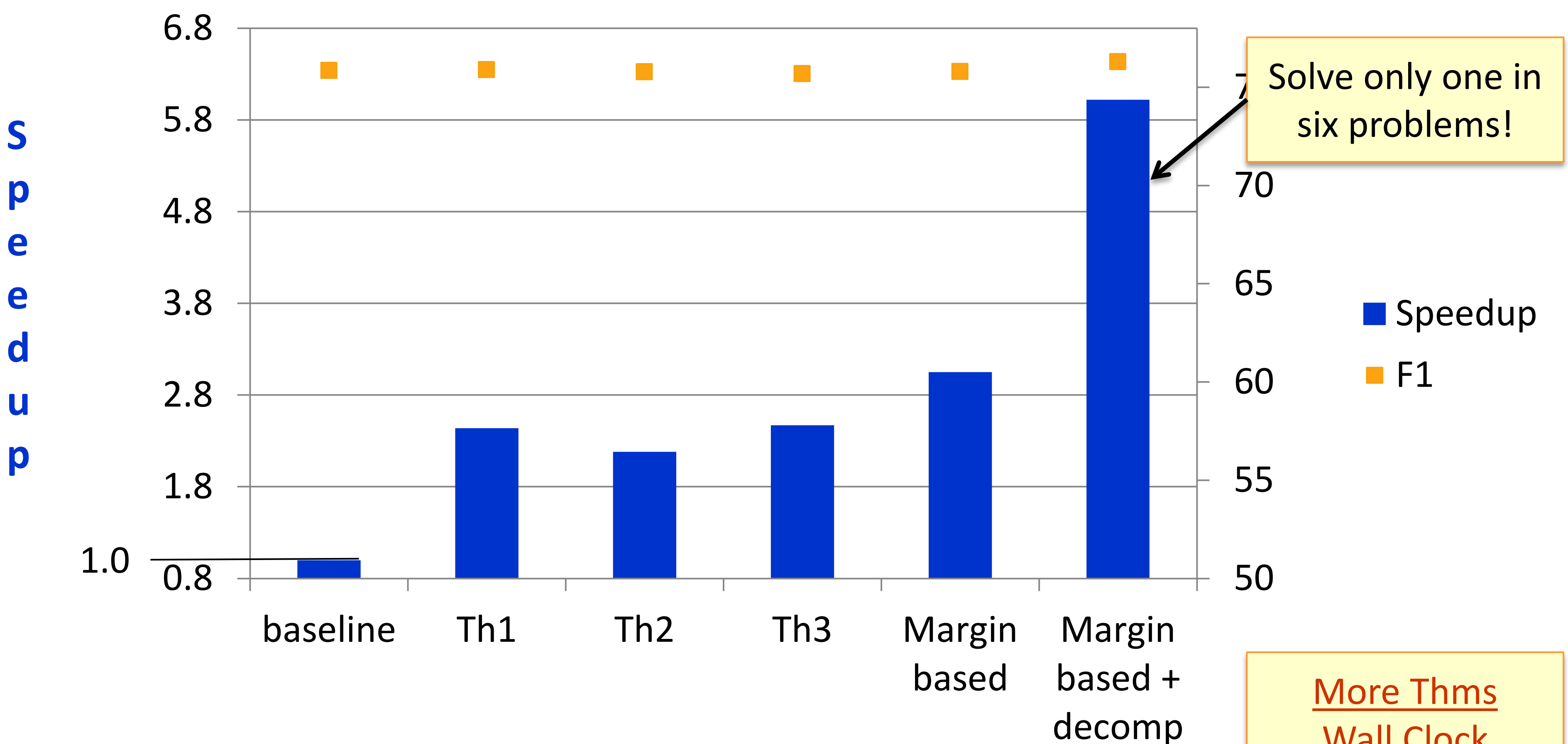
Amortization schemes [EMNLP'12, ACL'13]

Cognitive Computation Group
University of Illinois at Urbana-Champaign

Speedup & Accuracy

The results show that, indeed, the inference formulation provides a new level of abstraction that can be exploited to re-use solutions

$$\text{Speedup} = \frac{\text{number of inference calls without amortization}}{\text{number of inference calls with amortization}}$$

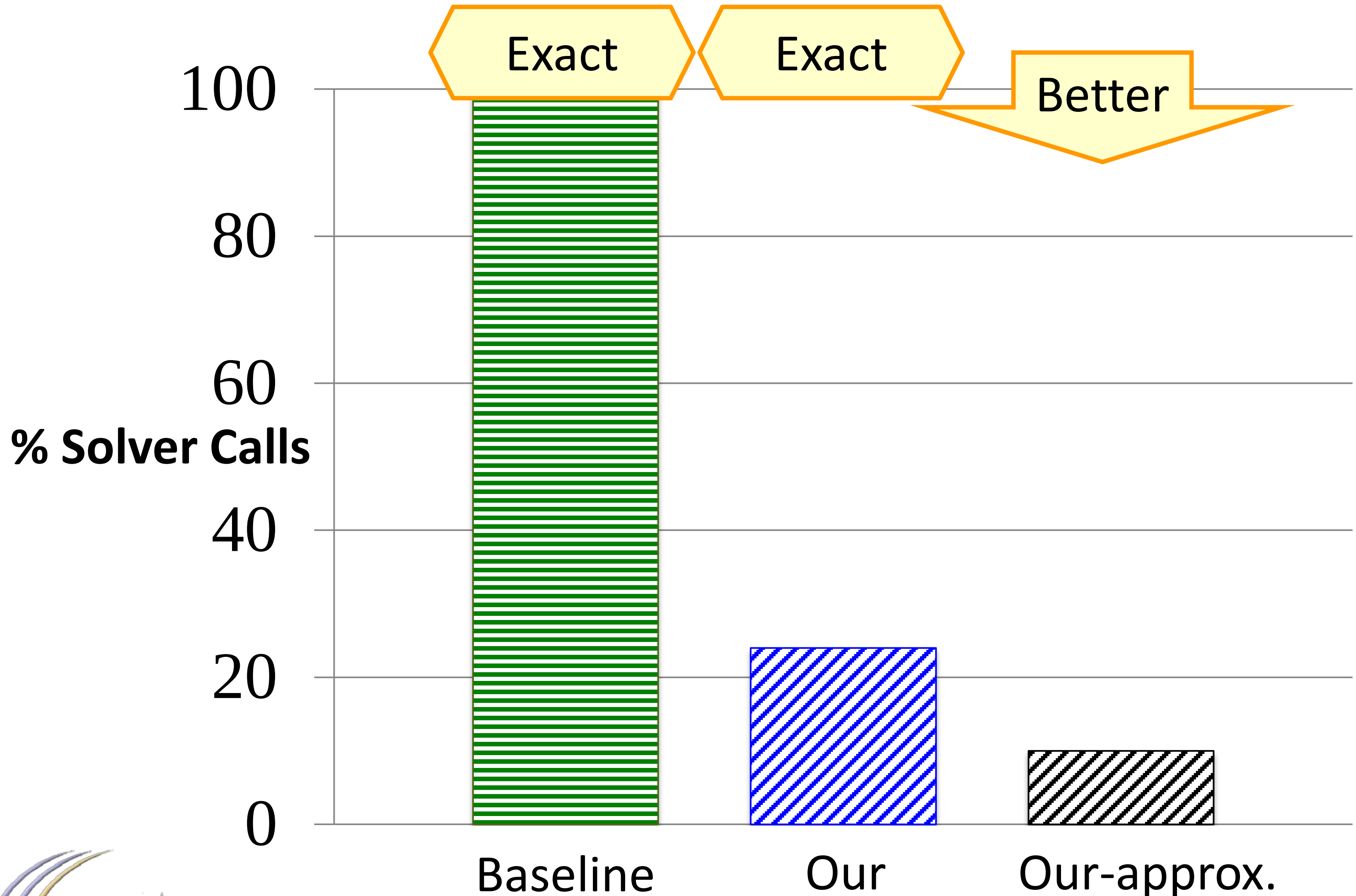


Amortization schemes [EMNLP'12, ACL'13]

More Thms
Wall Clock

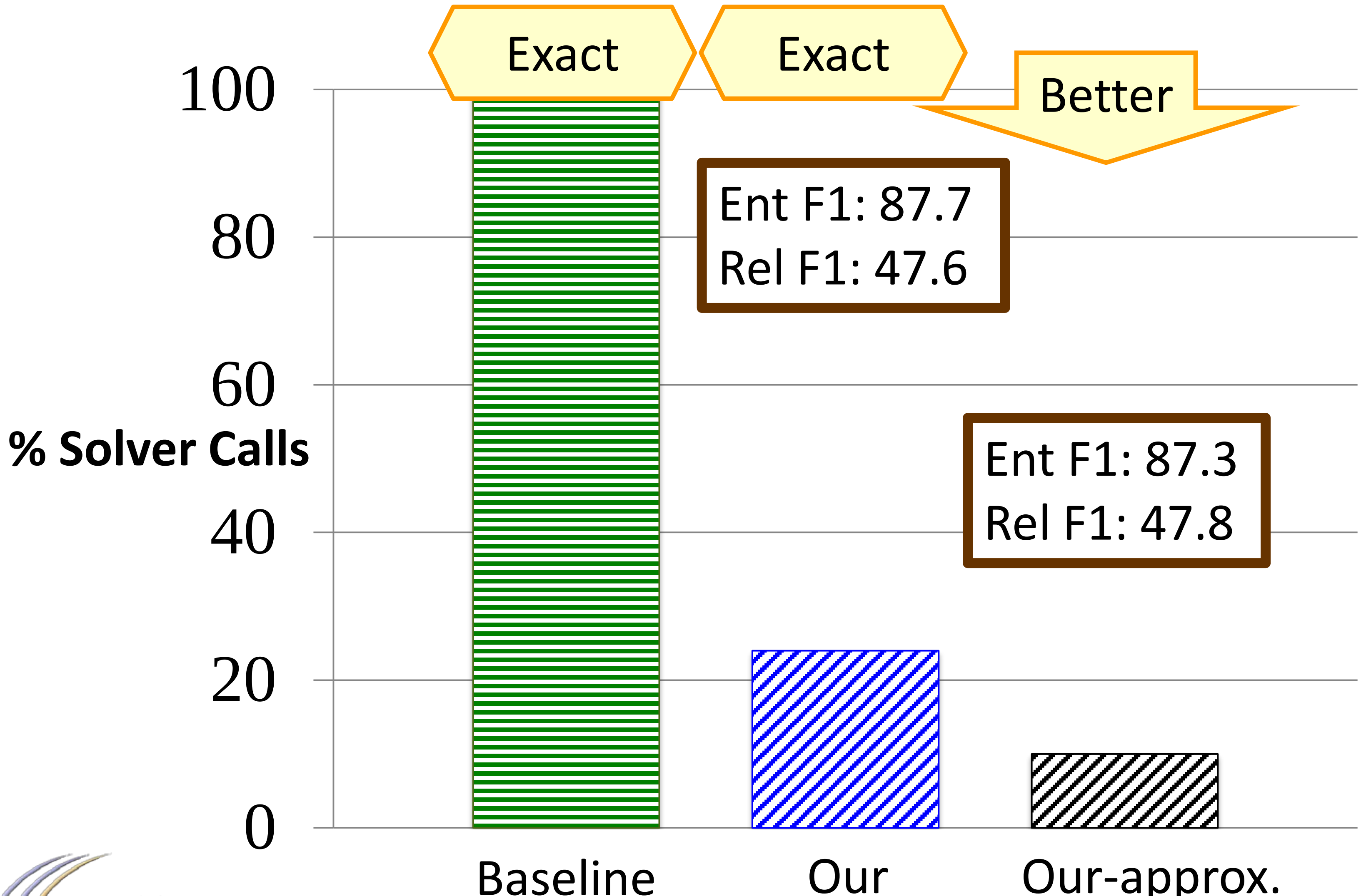
Cognitive Computation Group
University of Illinois at Urbana-Champaign

Solver Calls (Entity-Relation Extraction)



Solver Calls (Entity-Relation Extraction)

Recent results [AAAI'15] on how to exploit amortized ILP in faster Structured Learning



Conclusion

- Presented Constrained Conditional Models:
  - An ILP based computational framework that augments statistically learned linear models with declarative constraints as a way to incorporate knowledge and support decisions in an expressive output spaces
  - Maintains modularity and tractability of training
- A powerful & modular learning and inference paradigm for high level tasks.

- Learning issues:
  - Constraints driven learning, constrained EM
  - Many other issues have been and should be studied
- Inference:
  - The power of ILP formulations is shown via the amortized inference results: how to use previous inference outcomes to reduce inference and, consequently, learning cost

Conclusion

- Presented Constrained Conditional Models:
  - An ILP based computational framework that augments statistically learned linear models with declarative constraints as a way to incorporate knowledge and support decisions in an expressive output spaces
  - Maintains modularity and tractability of training
- A powerful & modular learning and inference paradigm for high level tasks.

- Learning issues:
  - Constraints driven learning, constrained EM
  - Many other issues have been and should be studied
- Inference:
  - The power of ILP formulations is shown via the amortized inference results: how to use previous inference outcomes to reduce inference and, consequently, learning cost

Check out our tools, demos, tutorials

Conclusion

Thank You!

- Presented Constrained Conditional Models:
  - An ILP based computational framework that augments statistically learned linear models with declarative constraints as a way to incorporate knowledge and support decisions in an expressive output spaces
  - Maintains modularity and tractability of training
- A powerful & modular learning and inference paradigm for high level tasks.

- Learning issues:
  - Constraints driven learning, constrained EM
  - Many other issues have been and should be studied
- Inference:
  - The power of ILP formulations is shown via the amortized inference results: how to use previous inference outcomes to reduce inference and, consequently, learning cost

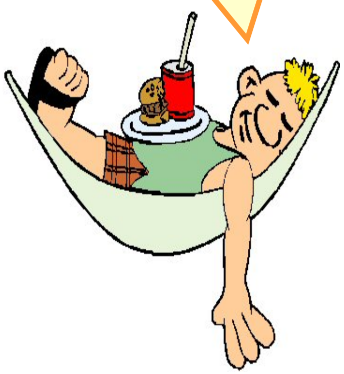
Check out our tools, demos, tutorials

Bonus Slides

- Response Based Learning
  - [From Clarke et. al. CoNLL'10 to Goldwasser & Roth MLJ'14]

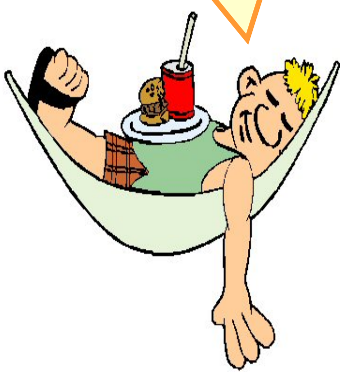
Understanding Language Requires Supervision

Can I get a coffee with lots of sugar and no milk



Understanding Language Requires Supervision

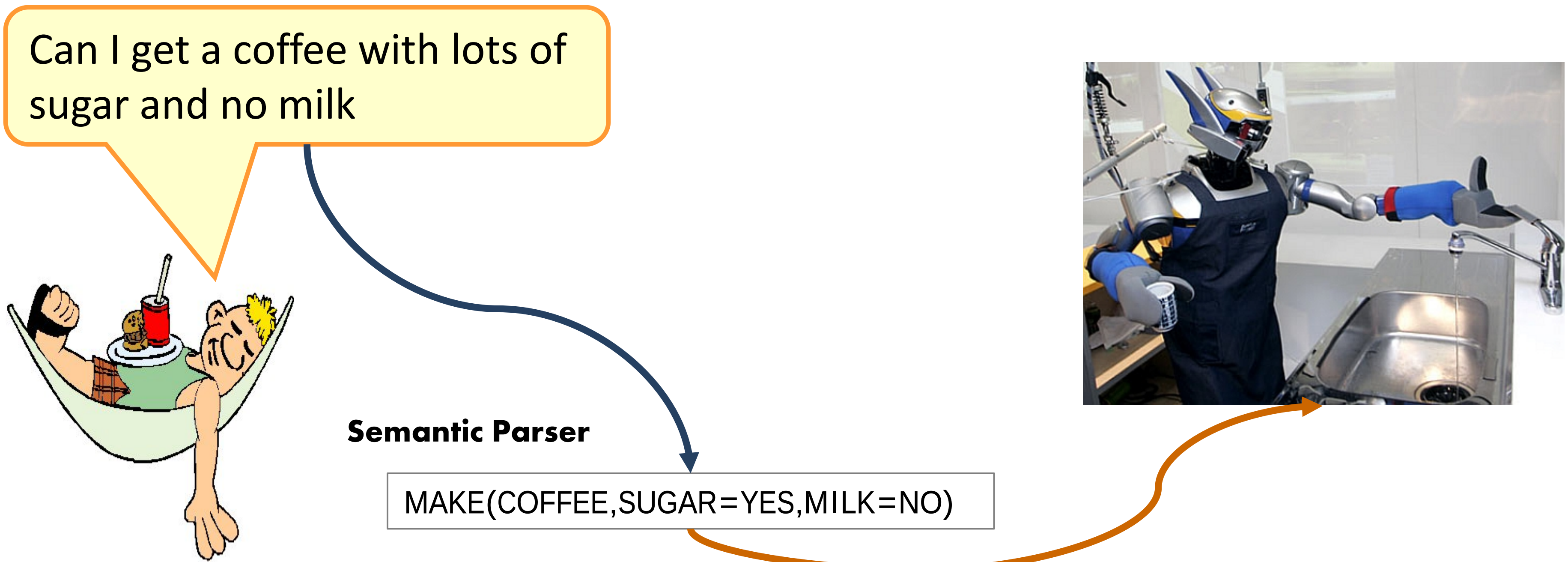
Can I get a coffee with lots of sugar and no milk



Semantic Parser

MAKE(COFFEE,SUGAR=YES,MILK=NO)

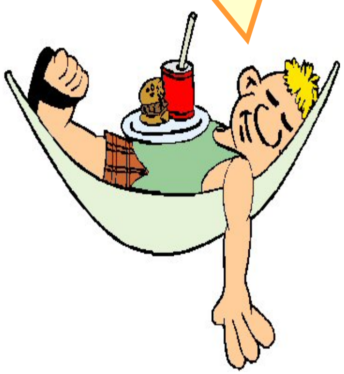
Understanding Language Requires Supervision



- How to recover meaning from text?

Understanding Language Requires Supervision

Can I get a coffee with lots of sugar and no milk



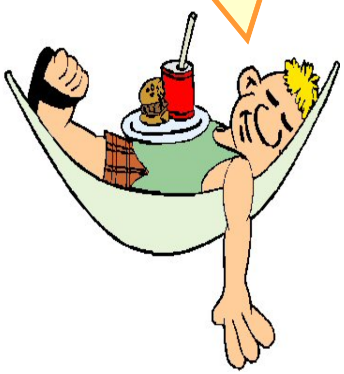
Semantic Parser

MAKE(COFFEE,SUGAR=YES,MILK=NO)

- How to recover meaning from text?
- Standard "example based" ML: annotate text with meaning representation
  - Teacher needs deep understanding of the learning agent ; not scalable.

Understanding Language Requires Supervision

Can I get a coffee with lots of sugar and no milk

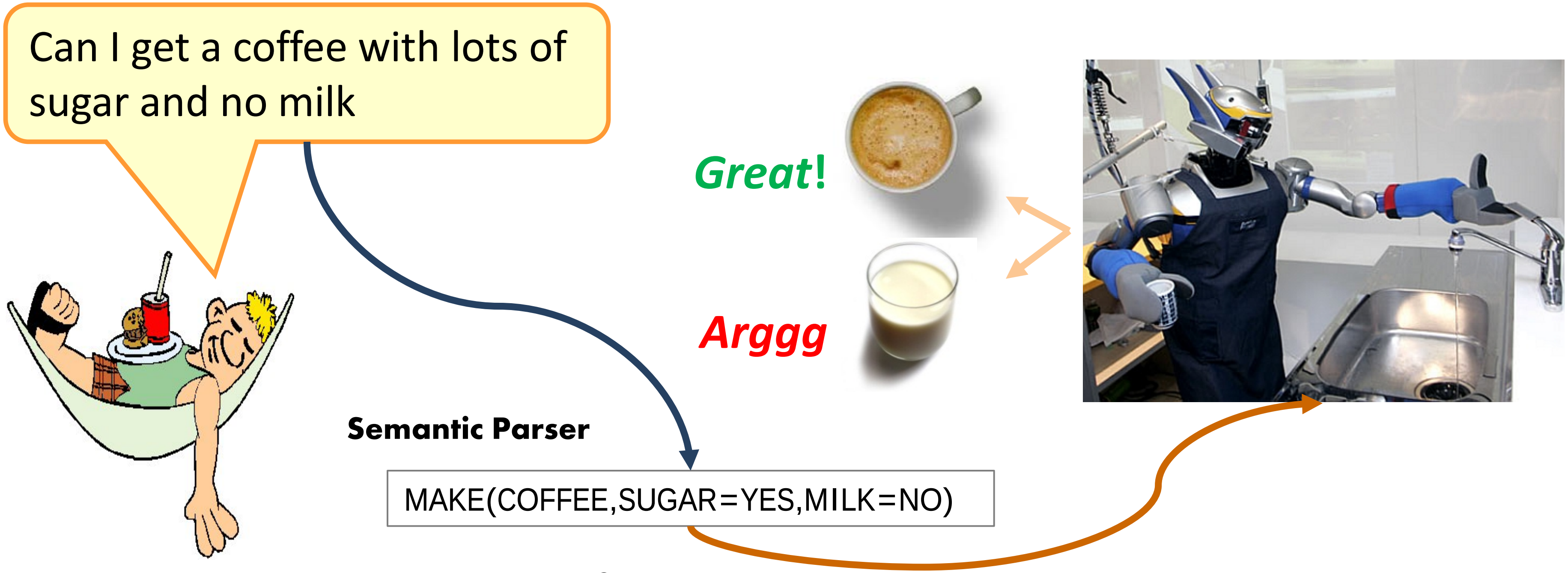


Semantic Parser

MAKE(COFFEE,SUGAR=YES,MILK=NO)

- How to recover meaning from text?
- Standard "example based" ML: annotate text with meaning representation
  - Teacher needs deep understanding of the learning agent ; not scalable.
- Response Driven Learning: Exploit indirect signals in the interaction between the learner and the teacher/environment

Understanding Language Requires Supervision

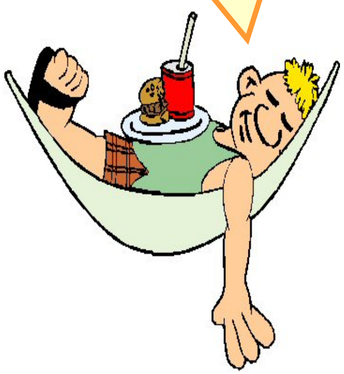


- How to recover meaning from text?
- Standard "example based" ML: annotate text with meaning representation
  - Teacher needs deep understanding of the learning agent ; not scalable.
- Response Driven Learning: Exploit indirect signals in the interaction between the learner and the teacher/environment

Understanding Language Requires Supervision

Can we rely on this interaction to provide supervision (and eventually, recover meaning) ?

Can I get a coffee with lots of sugar and no milk



Great!

Arggg

Semantic Parser

MAKE(COFFEE,SUGAR=YES,MILK=NO)

- How to recover meaning from text?
- Standard "example based" ML: annotate text with meaning representation
  - Teacher needs deep understanding of the learning agent ; not scalable.
- Response Driven Learning: Exploit indirect signals in the interaction between the learner and the teacher/environment

Response Based Learning

- We want to learn a model that transforms a natural language sentence to some meaning representation.

English Sentence → **Model** → Meaning Representation

- Instead of training with (Sentence, Meaning Representation) pairs

Response Based Learning

- We want to learn a model that transforms a natural language sentence to some meaning representation.

English Sentence → **Model** → Meaning Representation

- Instead of training with (Sentence, Meaning Representation) pairs
- Think about some simple derivatives of the models outputs,
  - Supervise the derivative [verifier] (easy!) and
  - Propagate it to learn the complex, structured, transformation model

Scenario I: Freecell with Response Based Learning

- We want to learn a model to transform a natural language sentence to some meaning representation.

English Sentence → **Model** → Meaning Representation

English Sentence	Meaning Representation
A top card can be moved to the tableau if it has a different color than the color of the top tableau card, and the cards have successive values.	Move (a1,a2) top(a1,x1) card(a1) tableau(a2) top(x2,a2) color(a1,x3) color(x2,x4) not-equal(x3,x4) value(a1,x5) value(x2,x6) successor(x5,x6)

Scenario I: Freecell with Response Based Learning

- We want to learn a model to transform a natural language sentence to some meaning representation.

English Sentence → **Model** → Meaning Representation

English Sentence	Meaning Representation
A top card can be moved to the tableau if it has a different color than the color of the top tableau card, and the cards have successive values.	Move (a1,a2) top(a1,x1) card(a1) tableau(a2) top(x2,a2) color(a1,x3) color(x2,x4) not-equal(x3,x4) value(a1,x5) value(x2,x6) successor(x5,x6)

- Simple derivatives of the models outputs
  - Supervise the derivative and
  - Propagate it to learn the transformation model

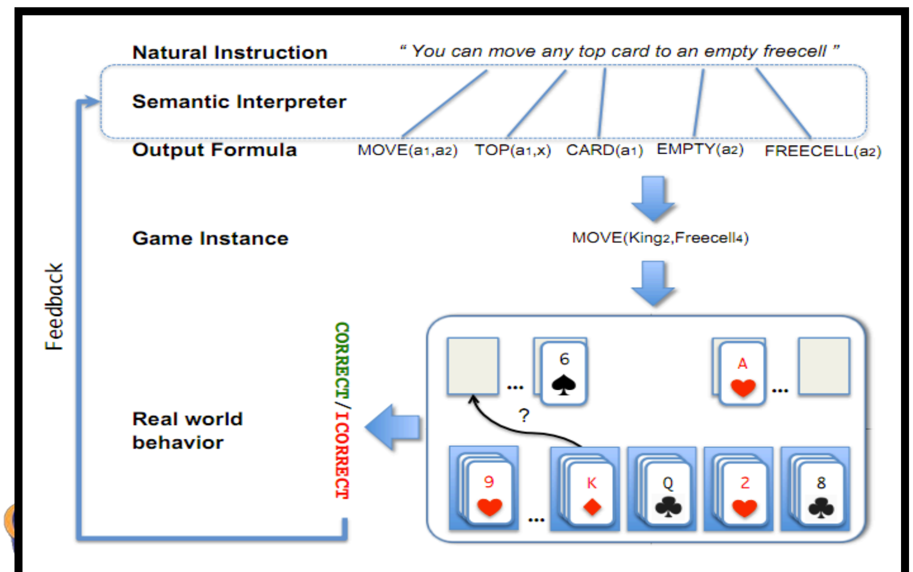
Scenario I: Freecell with Response Based Learning

- We want to learn a model to transform a natural language sentence to some meaning representation.

English Sentence → **Model** → Meaning Representation

A top card can be moved to the tableau if it has a different color than the color of the top tableau card, and the cards have successive values.

Move (a1,a2) top(a1,x1) card(a1) tableau(a2) top(x2,a2) color(a1,x3) color(x2,x4) not-equal(x3,x4) value(a1,x5) value(x2,x6) successor(x5,x6)



← Play Freecell (solitaire)

- Simple derivatives of the models outputs
  - Supervise the derivative and
  - Propagate it to learn the transformation model

Scenario II: Geoquery with Response based Learning

- We want to learn a model to transform a natural language sentence to some formal representation.

English Sentence → **Model** → Meaning Representation

What is the largest state that borders NY? → largest(state(next_to(const(NY))))

Scenario II: Geoquery with Response based Learning

- We want to learn a model to transform a natural language sentence to some formal representation.

English Sentence → **Model** → Meaning Representation

What is the largest state that borders NY?

largest(state(next_to(const(NY))))

- Simple derivatives of the models outputs

Scenario II: Geoquery with Response based Learning

- We want to learn a model to transform a natural language sentence to some formal representation.

English Sentence → **Model** → Meaning Representation

What is the largest state that borders NY?

largest(state(next_to(const(NY))))

- Query a GeoQuery Database.

- Simple derivatives of the models outputs

Scenario II: Geoquery with Response based Learning

- We want to learn a model to transform a natural language sentence to some formal representation.

English Sentence → **Model** → Meaning Representation

What is the largest state that borders NY? → largest(state(next_to(const(NY))))

- Query a GeoQuery Database. ← Simple derivatives of the models outputs

- "Guess" a semantic parse. **Is [DB response == Expected response] ?**
  - **Expected**: Pennsylvania **DB Returns**: Pennsylvania→ **Positive Response**
  - **Expected**: Pennsylvania **DB Returns**: NYC, or ???? → **Negative Response**

Response Based Learning: Using a Simple Feedback

- We want to learn a model to transform a natural language sentence to some formal representation.

English Sentence → **Model** → Meaning Representation

- Instead of training with (Sentence, Meaning Representation) pairs
- Think about some simple derivatives of the models outputs,
  - Supervise the derivative (easy!) and
  - Propagate it to learn the complex, structured, transformation model

Response Based Learning: Using a Simple Feedback

- We want to learn a model to transform a natural language sentence to some formal representation.

English Sentence → **Model** → Meaning Representation

- Instead of training with (Sentence, Meaning Representation) pairs
- Think about some simple derivatives of the models outputs,
  - Supervise the derivative (easy!) and
  - Propagate it to learn the complex, structured, transformation model

LEARNING:

- Train a structured predictor (semantic parse) with this binary supervision
  - Many challenges: e.g., how to make a better use of a negative response?
- Learning with a constrained latent representation, making used of CCM inference, exploiting knowledge on the structure of the meaning representation.

Geoquery: Response based Competitive with Supervised

Clarke, Goldwasser, Chang, Roth CoNLL'10; Goldwasser, Roth IJCAI'11, MLJ'14

Algorithm	Training Accuracy	Testing Accuracy	# Training Examples
NoLearn	22	--	-
Supervised	--	86.07	**600 structs.**

NoLearn :Initialization point

Supervised : Trained with annotated data

Response based Learning is gathering momentum:

- Liang, M.I. Jordan, D. Klein, Learning Dependency-Based Compositional Semantics, ACL'11.
- Berant et-al ' Semantic Parsing on Freebase from Question-Answer Pairs, EMNLP'13

Supervised: Y.-W. Wong and R. Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. ACL'07

Geoquery: Response based Competitive with Supervised

Clarke, Goldwasser, Chang, Roth CoNLL'10; Goldwasser, Roth IJCAI'11, MLJ'14

Algorithm	Training Accuracy	Testing Accuracy	# Training Examples
NoLearn	22	--	-
Response-based (2010)	82.4	73.2	250 answers
Liang et-al 2011	--	78.9	250 answers
Response-based (2012)	**86.8**	**81.6**	250 answers
Supervised	--	86.07	**600 structs**.

NoLearn :Initialization point

Supervised : Trained with annotated data

Response based Learning is gathering momentum:

- Liang, M.I. Jordan, D. Klein, Learning Dependency-Based Compositional Semantics, ACL'11.
- Berant et-al ' Semantic Parsing on Freebase from Question-Answer Pairs, EMNLP'13

Supervised: Y.-W. Wong and R. Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. ACL'07