Semantic Role Labeling and Constrained Conditional Models

Mausam

Slides by Ming-Wei Chang, Nick Rizzolo, Dan Roth, Dan Jurafsky

Nice to Meet You

n+ 

2  u

ILP & Constraints Conditional Models (CCMs)

- Making global decisions in which several local interdependent decisions play a role.
- Informally:
  - Everything that has to do with constraints (and learning models)
- Forr
  -
  - s:
  - ons

Issues to attend to:

- **While we formulate the problem as an ILP problem, Inference can be done multiple ways**
  - Search; sampling; dynamic programming; SAT; ILP
- **The focus is on joint global inference**
- **Learning may or may not be joint.**
  - Decomposing models is often beneficial

- CCMs make predictions in the presence of /guided by constraints

Constraints Driven Learning and Decision Making

- Why Constraints?
  - **The Goal: Building a good NLP systems <u>easily</u>**
  - **We have prior knowledge at our hand**
    - How can we use it?
    - We suggest that knowledge can often be injected directly
      - **Can use it to guide learning**
      - **Can use it to improve decision making**
      - **Can use it to simplify the models we need to learn**

- How useful are constraints?
  - **Useful for supervised learning**
  - **Useful for semi-supervised & other label-lean learning paradigms**
  - **Sometimes more efficient than labeling data directly**

Motivation: IE via Hidden Markov Models

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 .

Unsatisfactory 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
  - What if we only have a few labeled examples?

Increasing the model complexity

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

- Any other options?
  - Humans can <u>immediately</u> detect bad outputs
  - The output does not make sense

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 .

Violates lots of natural constraints!

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 Constraints

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

- [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 .

Constrained Conditional Models Allow:

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

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

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

How to solve?

This is an Integer Linear Program
Solving using ILP packages gives an exact solution.
Search techniques are also possible

How to train?

Training is learning the objective Function.
How to exploit the structure to minimize supervision?

$$f_{\Phi,C}(\mathbf{x},\mathbf{y}) = \sum w_i \phi_i(\mathbf{x},\mathbf{y}) - \sum \rho_i d_{C_i}(\mathbf{x},\mathbf{y}).$$

Features Versus Constraints

- $\phi_i : X \times Y \rightarrow R$; $C_i : X \times Y \rightarrow \{0,1\}$; $d: X \times Y \rightarrow R$;
  - In principle, constraints and features can encode the same propeties
  - In practice, they are very different

- Features
  - Local , short distance properties – to allow tractable inference
  - Propositional (grounded):
  - E.g. True if: "the" followed by a Noun occurs in the sentence"
- Constraints
  - Global properties
  - Quantified, first order logic expressions
  - E.g.True if: "all y_is in the sequence y are assigned different values."

Indeed, used differently

Encoding Prior Knowledge

- Consider encoding the knowledge that:
  - Entities of type A and B cannot occur simultaneously in a sentence
- The "Feature" Way
  - Results in higher order HMM, CRF
  - May require designing a model tailored to knowledge/constraints
  - Large number of new features: might require more labeled data
  - Wastes parameters to learn indirectly knowledge we have.

Need more training data

- The Constraints Way
  - Keeps the model simple; add expressive constraints directly
  - A small set of constraints
  - Allows for decision time incorporation of constraints

A form of supervision

CCMs are Optimization Problems

- We pose inference as an optimization problem
  - Integer Linear Programming (ILP)
- Advantages:
  - *Keep model small; easy to learn*
  - *Still allowing expressive, long-range constraints*
  - Mathematical optimization is well studied
  - Exact solution to the inference problem is possible
  - Powerful off-the-shelf solvers exist
- Disadvantage:
  - The inference problem could be NP-hard

CCM Examples

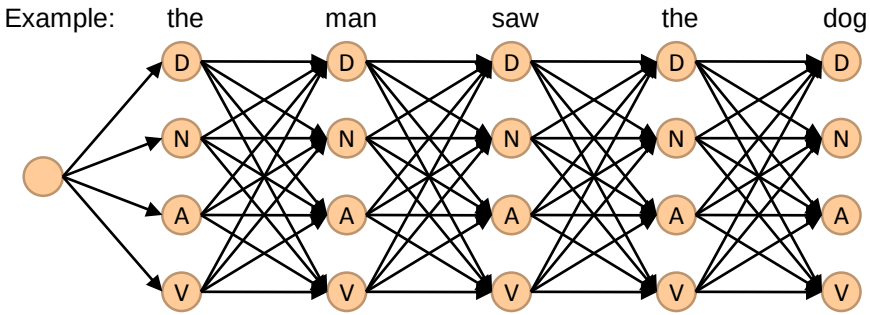
- Many works in NLP make use of constrained conditional models, implicitly or explicitly.
- Next we describe two examples in detail.

- Example 1: Sequence Tagging
  - Adding long range constraints to a simple model
- Example 2: Semantic Role Labeling
  - The use of inference with constraints to improve semantic parsing

Example 1: Sequence Tagging

HMM / CRF:

$$\mathbf{y}^* = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} P(y_0)P(x_0|y_0)\prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i)$$



Inference Variables

As an ILP:

$$\text{maximize} \quad \sum_{y\in\mathcal{Y}} \lambda_{0,y} 1_{\{y_0=y\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}} \lambda_{i,y,y'} 1_{\{y_i=y \,\wedge\, y_{i-1}=y'\}}$$

$$\lambda_{0,y} = \log(P(y)) + \log(P(x_0|y))$$

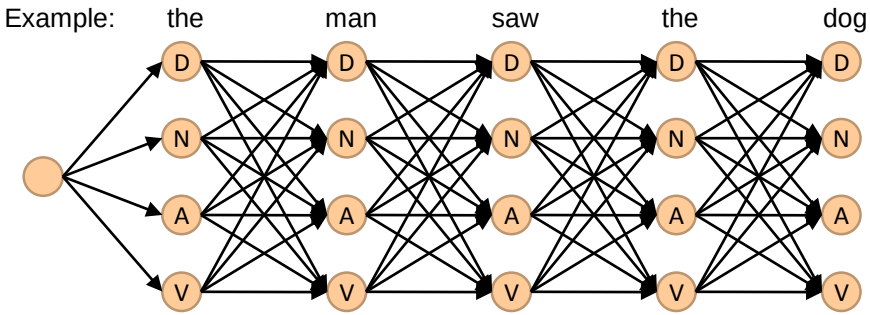
$$\lambda_{i,y,y'} = \log(P(y|y')) + \log(P(x_i|y))$$

subject to

Example 1: Sequence Tagging

HMM / CRF:

$$\mathbf{y}^* = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} P(y_0)P(x_0|y_0)\prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i)$$



As an ILP:

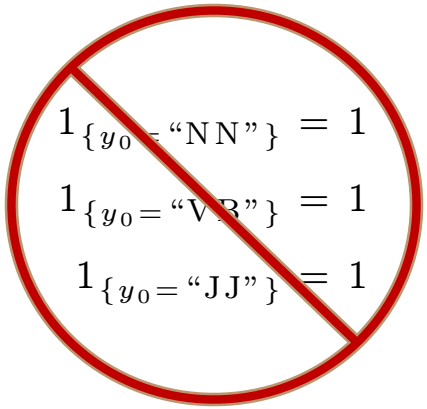
$$\text{maximize} \quad \sum_{y\in\mathcal{Y}} \lambda_{0,y} 1_{\{y_0=y\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}} \lambda_{i,y,y'} 1_{\{y_i=y \,\wedge\, y_{i-1}=y'\}}$$

$$\lambda_{0,y} = \log(P(y)) + \log(P(x_0|y))$$

$$\lambda_{i,y,y'} = \log(P(y|y')) + \log(P(x_i|y))$$

$$\text{subject to} \quad \sum_{y\in\mathcal{Y}} 1_{\{y_0=y\}} = 1$$

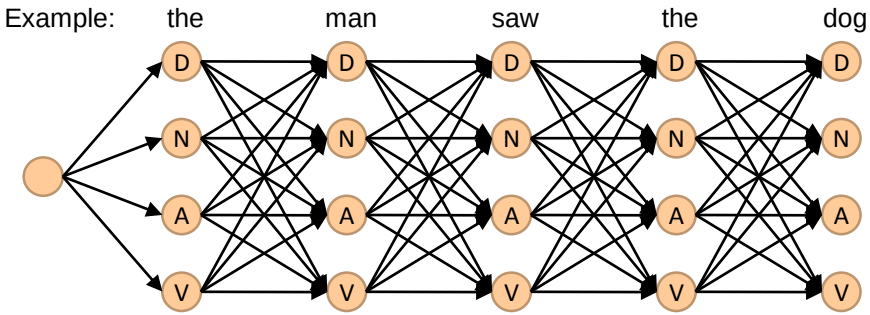
Discrete predictions



Example 1: Sequence Tagging

HMM / CRF:

$$\mathbf{y}^* = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} P(y_0)P(x_0|y_0)\prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i)$$



As an ILP:

$$\text{maximize} \quad \sum_{y\in\mathcal{Y}} \lambda_{0,y} 1_{\{y_0=y\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}} \lambda_{i,y,y'} 1_{\{y_i=y \,\wedge\, y_{i-1}=y'\}}$$

$$\lambda_{0,y} = \log(P(y)) + \log(P(x_0|y))$$

$$\lambda_{i,y,y'} = \log(P(y|y')) + \log(P(x_i|y))$$

subject to

$$\sum_{y\in\mathcal{Y}} 1_{\{y_0=y\}} = 1$$

Discrete predictions

$$\forall y, \quad 1_{\{y_0=y\}} = \sum_{y'\in\mathcal{Y}} 1_{\{y_0=y \,\wedge\, y_1=y'\}}$$

$$\forall y, i > 1 \quad \sum_{y'\in\mathcal{Y}} 1_{\{y_{i-1}=y' \,\wedge\, y_i=y\}} = \sum_{y''\in\mathcal{Y}} 1_{\{y_i=y \,\wedge\, y_{i+1}=y''\}}$$

Feature consistency

$$1_{\{y_0=\text{“NN”}\}} = 1$$

$$1_{\{y_0=\text{“DT”} \,\wedge\, y_1=\text{“JJ”}\}} = 1$$

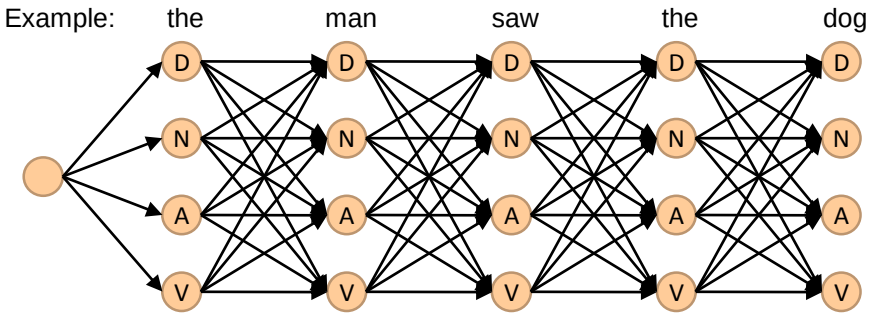
$$1_{\{y_0=\text{“DT”} \,\wedge\, y_1=\text{“JJ”}\}} = 1$$

$$1_{\{y_1=\text{“NN”} \,\wedge\, y_2=\text{“VB”}\}} = 1$$

Example 1: Sequence Tagging

HMM / CRF:

$$\mathbf{y}^* = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} P(y_0)P(x_0|y_0)\prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i)$$



As an ILP:

$$\text{maximize} \quad \sum_{y\in\mathcal{Y}} \lambda_{0,y} 1_{\{y_0=y\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}} \lambda_{i,y,y'} 1_{\{y_i=y \,\wedge\, y_{i-1}=y'\}}$$

$$\lambda_{0,y} = \log(P(y)) + \log(P(x_0|y))$$

$$\lambda_{i,y,y'} = \log(P(y|y')) + \log(P(x_i|y))$$

subject to

$$\sum_{y\in\mathcal{Y}} 1_{\{y_0=y\}} = 1$$

Discrete predictions

$$\forall y, \quad 1_{\{y_0=y\}} = \sum_{y'\in\mathcal{Y}} 1_{\{y_0=y \,\wedge\, y_1=y'\}}$$

$$\forall y, i > 1 \quad \sum_{y'\in\mathcal{Y}} 1_{\{y_{i-1}=y' \,\wedge\, y_i=y\}} = \sum_{y''\in\mathcal{Y}} 1_{\{y_i=y \,\wedge\, y_{i+1}=y''\}}$$

Feature consistency

$$1_{\{y_0=\text{“V”}\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} 1_{\{y_{i-1}=y \,\wedge\, y_i=\text{“V”}\}} \geq 1$$

There must be a verb!

CCM Examples: (Add Constraints; Solve as ILP)

- Many works in NLP make use of constrained conditional models, implicitly or explicitly.
- Next we describe two examples in detail.

- Example 1: Sequence Tagging
  - Adding long range constraints to a simple model
- Example 2: Semantic Role Labeling
  - The use of inference with constraints to improve semantic parsing

Semantic Role Labeling

Semantic Role Labeling

Who did *what* to *whom* at *where*?

The police officer detained the suspect at the scene of the crime

Agent Predicate Theme Location

Can we figure out that these have the same meaning?

XYZ corporation **bought** the stock.

They **sold** the stock to XYZ corporation.

The stock was **bought** by XYZ corporation.

The **purchase** of the stock by XYZ corporation...

The stock **purchase** by XYZ corporation...

A Shallow Semantic Representation: Semantic Roles

Predicates (bought, sold, purchase) represent an **event**

semantic roles express the abstract role that arguments of a predicate can take in the event

Thematic roles

- A typical set:

Thematic Role	Definition	Example
AGENT	The volitional causer of an event	*The waiter* spilled the soup.
EXPERIENCER	The experiencer of an event	*John* has a headache.
FORCE	The non-volitional causer of the event	*The wind* blows debris from the mall into our yards.
THEME	The participant most directly affected by an event	Only after Benjamin Franklin broke *the ice*...
RESULT	The end product of an event	The city built a *regulation-size baseball diamond*...
CONTENT	The proposition or content of a propositional event	Mona asked *"You met Mary Ann at a supermarket?"*
INSTRUMENT	An instrument used in an event	He poached catfish, stunning them *with a shocking device*...
BENEFICIARY	The beneficiary of an event	Whenever Ann Callahan makes hotel reservations *for her boss*...
SOURCE	The origin of the object of a transfer event	I flew in *from Boston*.
GOAL	The destination of an object of a transfer event	I drove *to Portland*.

Thematic grid, case frame, θ-grid

Example usages of "break"

John broke the window.
AGENT THEME

John broke the window with a rock.
AGENT THEME INSTRUMENT

The rock broke the window.
INSTRUMENT THEME

The window broke.
THEME

The window was broken by John.
THEME AGENT

thematic grid, case frame, θ-grid

Break:

AGENT, THEME, INSTRUMENT.

Some realizations:

AGENT/Subject, THEME/Object
AGENT/Subject, THEME/Object, INSTRUMENT/PP_{with}
INSTRUMENT/Subject, THEME/Object
THEME/Subject

Problems with Thematic Roles

Hard to create standard set of roles or formally define them

Often roles need to be fragmented to be defined.

Levin and Rappaport Hovav (2015): two kinds of INSTRUMENTS

intermediary instruments that can appear as subjects

The cook opened the jar with the new gadget.

The new gadget opened the jar.

enabling instruments that cannot

Shelly ate the sliced banana with a fork.

*The fork ate the sliced banana.

Alternatives to thematic roles

1. **Fewer roles**: generalized semantic roles, defined as prototypes (Dowty 1991)

   PROTO-AGENT

   PROTO-PATIENT

   PropBank

2. **More roles**: Define roles specific to a group of predicates

   FrameNet

PropBank

- Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106

PropBank Roles

Following Dowty 1991

Proto-Agent

- Volitional involvement in event or state
- Sentience (and/or perception)
- Causes an event or change of state in another participant
- Movement (relative to position of another participant)

Proto-Patient

- Undergoes change of state
- Causally affected by another participant
- Stationary relative to movement of another participant

PropBank Roles

- Following Dowty 1991
  - Role definitions determined verb by verb, with respect to the other roles
  - Semantic roles in PropBank are thus verb-sense specific.
- Each verb sense has numbered argument: Arg0, Arg1, Arg2,…

Arg0: PROTO-AGENT

Arg1: PROTO-PATIENT

Arg2: usually: benefactive, instrument, attribute, or end state

Arg3: usually: start point, benefactive, instrument, or attribute

Arg4 the end point

(Arg2-Arg5 are not really that consistent, causes a problem for labeling)

PropBank Frame Files

agree.01

Arg0: Agreer
Arg1: Proposition
Arg2: Other entity agreeing

Ex1: [$_{Arg0}$ The group] *agreed* [$_{Arg1}$ it wouldn't make an offer].
Ex2: [$_{ArgM\text{-}TMP}$ Usually] [$_{Arg0}$ John] *agrees* [$_{Arg2}$ with Mary] [$_{Arg1}$ on everything].

fall.01

Arg1: Logical subject, patient, thing falling
Arg2: Extent, amount fallen
Arg3: start point
Arg4: end point, end state of arg1
Ex1: [$_{Arg1}$ Sales] *fell* [$_{Arg4}$ to $25 million] [$_{Arg3}$ from $27 million].
Ex2: [$_{Arg1}$ The average junk bond] *fell* [$_{Arg2}$ by 4.2%].

Advantage of a ProbBank Labeling

` **increase.01** "go up incrementally"

Arg0: causer of increase
Arg1: thing increasing
Arg2: amount increased by, EXT, or MNR
Arg3: start point
Arg4: end point

This would allow us to see the commonalities in these 3 sentences:

[$_{Arg0}$ Big Fruit Co.] increased [$_{Arg1}$ the price of bananas].
[$_{Arg1}$ The price of bananas] was increased again [$_{Arg0}$ by Big Fruit Co.]
[$_{Arg1}$ The price of bananas] increased [$_{Arg2}$ 5%].

Modifiers or adjuncts of the predicate: Arg-M

ArgM-TMP	when?	yesterday evening, now
LOC	where?	at the museum, in San Francisco
DIR	where to/from?	down, to Bangkok
MNR	how?	clearly, with much enthusiasm
PRP/CAU	why?	because ... , in response to the ruling
REC		themselves, each other
ADV	miscellaneous	
PRD	secondary predication	...ate the meat raw

Annotated PropBank Data

- Penn English TreeBank, OntoNotes 5.0.
  - Total ~2 million words
- Penn Chinese TreeBank
- Hindi/Urdu PropBank
- Arabic PropBank

2013 Verb Frames Coverage
Count of word sense (lexical units)

Language	*Final Count*
English	**10,615***
Chinese	**24, 642**
Arabic	**7,015**

From Martha Palmer 2013 Tutorial

Example 2: Semantic Role Labeling

Who did what to whom, when, where, why,…

Semantic Role Labeling Output

Demo:http://L2R.cs.uiuc.edu/~cogcomp

Input Text:

A car bomb that exploded outside the U.S. military base in Beniji killed 11 Iraqi citizens.

Result: Complete!

⊞ General Explanation of Argument Labels

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		
.		

Approach :

1) Reveals several relations.

2) Produces a very good semantic parser. F1~90%

3) Easy and fast: ~7 Sent/Sec (using Xpress-MP)

Top ranked system in CoNLL'05 shared task

Key difference is the Inference

Simple sentence:

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 .

A simple modern algorithm

```
function SEMANTICROLELABEL(words) returns labeled tree

    parse ← PARSE(words)
    for each predicate in parse do
        for each node in parse do
            featurevector ← EXTRACTFEATURES(node, predicate, parse)
            CLASSIFYNODE(node, featurevector, parse)
```

- How do we decide what is a predicate
  - Choose all verbs
  - Possibly removing light verbs (from a list)

3-step version of SRL algorithm

1. **Pruning**: use simple heuristics to prune unlikely constituents.
2. **Identification**: a binary classification of each node as an argument to be labeled or a NONE.
3. **Classification**: a 1-of-*N* classification of all the constituents that were labeled as arguments by the previous stage

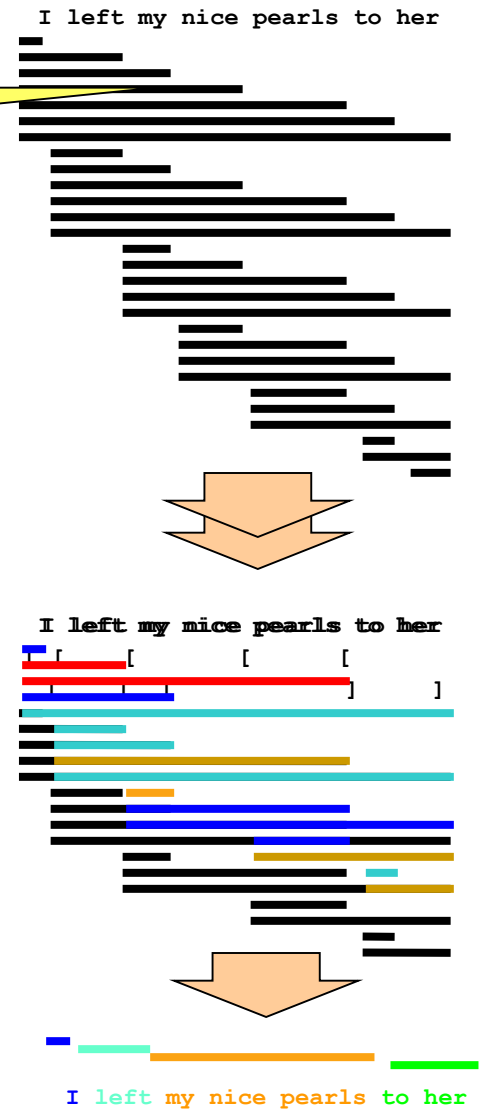
Why add Pruning and Identification steps?

- Algorithm is looking at one predicate at a time
- Very few of the nodes in the tree could possible be arguments of that one predicate
- Imbalance between
  - positive samples (constituents that are arguments of predicate)
  - negative samples (constituents that are not arguments of predicate)
- Imbalanced data can be hard for many classifiers
- So we prune the **very** unlikely constituents first, and then use a classifier to get rid of the rest.

Algorithmic Approach

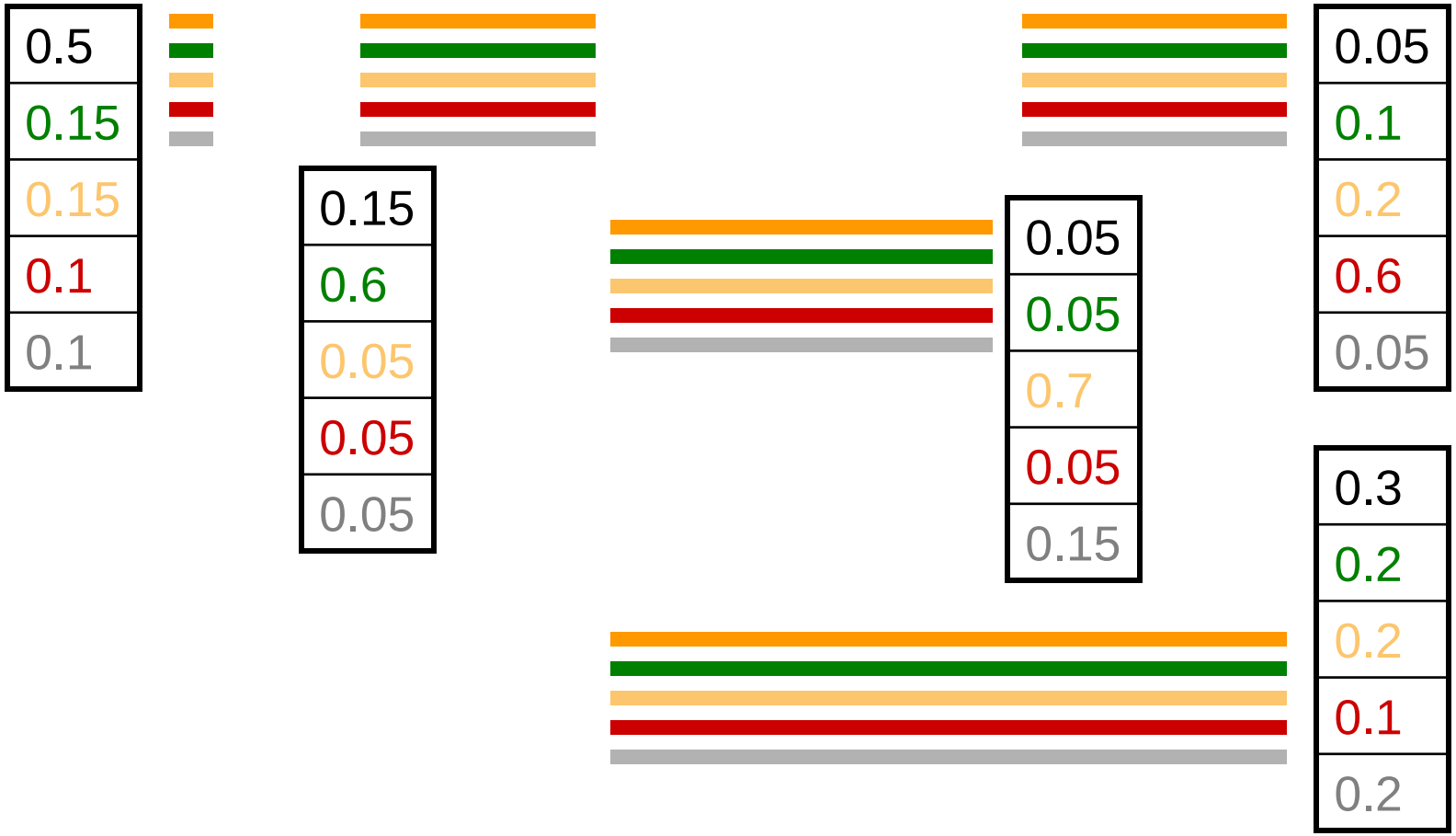
candidate arguments

- 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



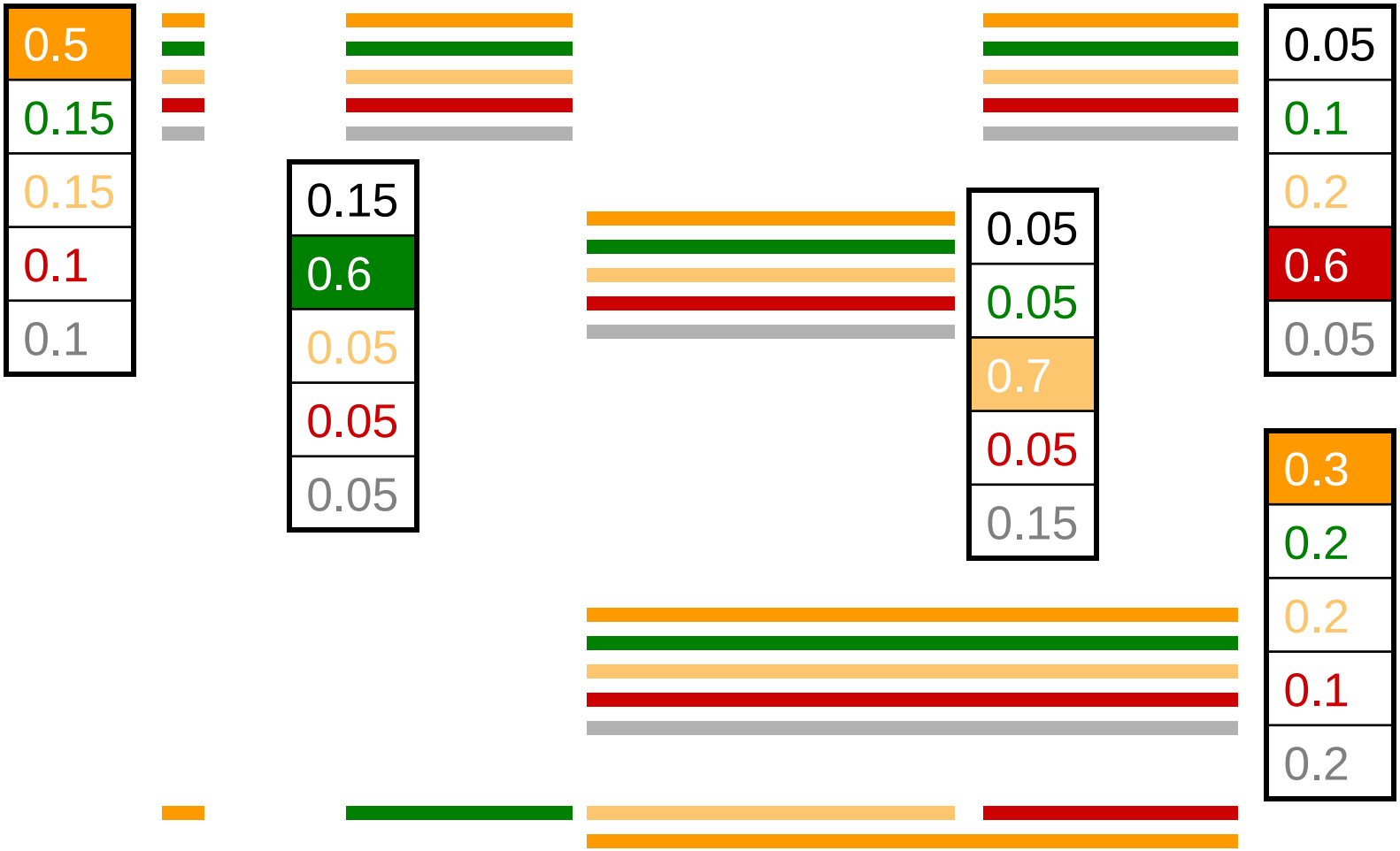
Semantic Role Labeling (SRL)

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



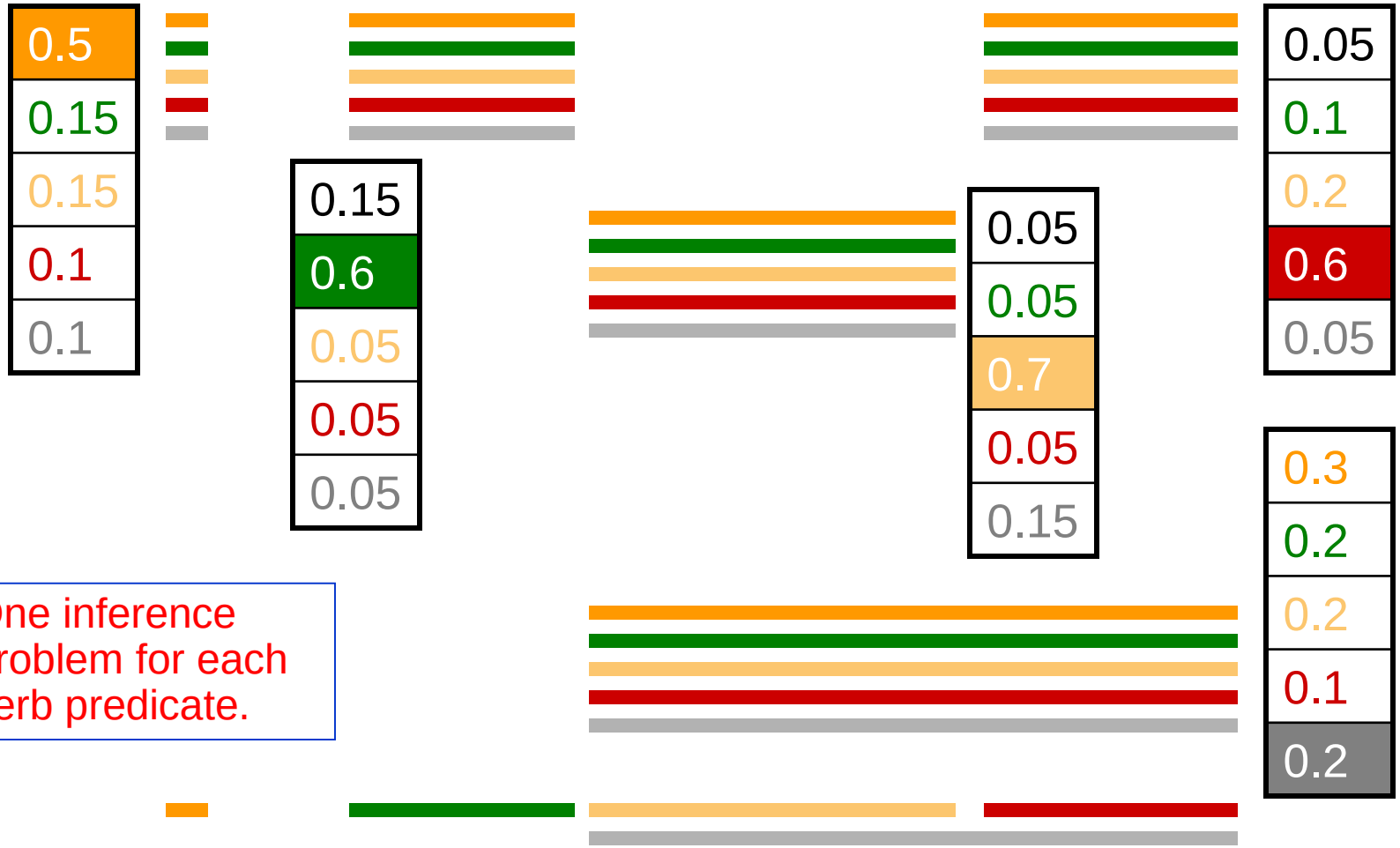
Semantic Role Labeling (SRL)

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



Semantic Role Labeling (SRL)

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



One inference problem for each verb predicate.

Constraints

Any Boolean rule can be encoded as a set of linear inequalities.

- No duplicate argument classes

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

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

- R-Ax

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

If there is an C-x phrase, there is an Ax [illegible] e it

- C-Ax

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

Universally quantified rules

LBJ: allows a developer to encode constraints in FOL; these are compiled into linear inequalities automatically.

- Many other possible constraints:
  - **Unique labels**
  - **No overlapping or embedding**
  - **Relations between number of arguments; order constraints**
  - **If each is of type A, no argument of type B**

Joint inference can be used also to combine different SRL Systems.

SRL: Posing the Problem

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

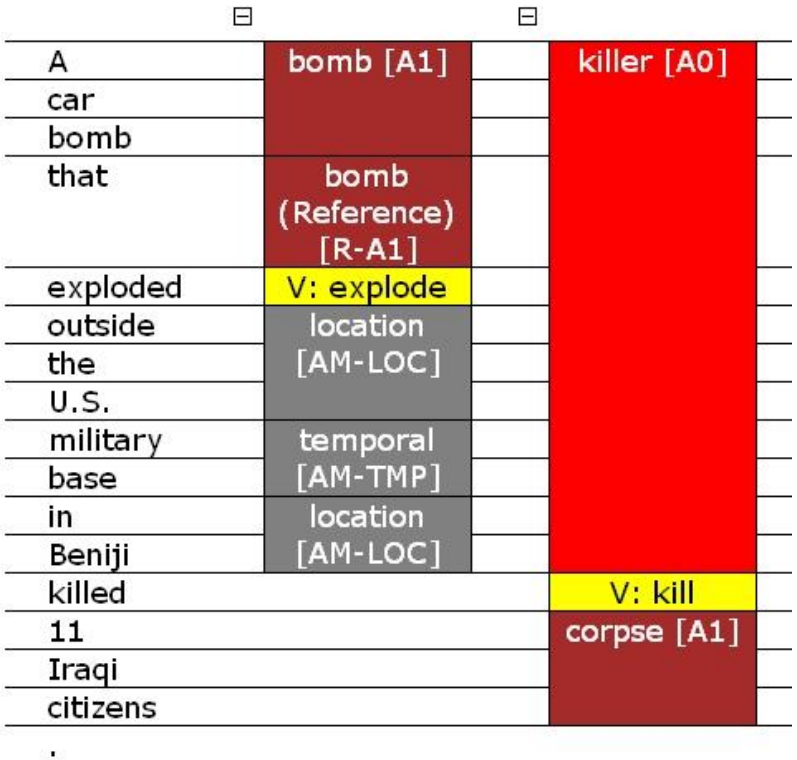
$$\text{where} \quad \lambda_{\mathbf{x}, y} = \lambda \cdot F(\mathbf{x}, y) = \lambda_y \cdot F(\mathbf{x})$$

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

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

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

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



Solvers

- All applications presented so far used ILP for inference.
- People used different solvers
  - Xpress-MP
  - GLPK
  - lpsolve
  - R
  - Mosek
  - CPLEX
- Other search-based algorithms can also be used

Training Constrained Conditional Models

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

- Learning model
  - Independently of the constraints (L+I)
  - Jointly, in the presence of the constraints (IBT)
  - Decomposed to simpler models
- Learning constraints' penalties
  - Independently of learning the model
  - Jointly, along with learning the model
- Dealing with lack of supervision
  - Constraints Driven Semi-Supervised learning (CODL)
  - Indirect Supervision
- Learning Constrained Latent Representations

Soft Constraints

$$-\sum_{i=1}^{K} \rho_k \underline{d(y, 1_{C_i(x)})}$$

- Hard Versus Soft Constraints
  - Hard constraints: Fixed Penalty $\rho_i = \infty$
  - Soft constraints: Need to set the penalty

- Why soft constraints?
  - Constraints might be violated by gold data
  - Some constraint violations are more serious
  - An example can violate a constraint multiple times!
  - <u>Degree of violation is only meaningful when constraints are soft!</u>

Learning the penalty weights

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

- **Strategy 1: Independently of learning the model**
  - Handle the learning parameters λ and the penalty ρ separately
  - Learn a feature model and a constraint model
  - Similar to L+I, but also learn the penalty weights
  - Keep the model simple

- **Strategy 2: Jointly, along with learning the model**
  - Handle the learning parameters λ and the penalty ρ together
  - Treat soft constraints as high order features
  - Similar to IBT, but also learn the penalty weights

Strategy 1: Independently of learning the model

- Model: (First order) Hidden Markov Model $P_\theta(x, y)$

- Constraints: long distance constraints
  - The i-th the constraint: C_i
  - The probability that the i-th constraint is violated $P(C_i = 1)$

- The learning problem
  - Given labeled data, estimate θ and $P(C_i = 1)$
  - For one labeled example,

$$\textsc{Score}(x, y) = \text{HMM Probability} \times \text{Constraint Violation Score}$$

  - Training: Maximize the score of all labeled examples!

Strategy 1: Independently of learning the model (cont.)

$$\textsc{Score}(x, y) = \text{HMM Probability} \times \text{ Constraint Violation Score}$$

- The new score function is a CCM!
  - Setting $\rho_i = -\log \frac{P(C_i = 1)}{P(C_i = 0)}$
  - New score:

$$\log \textsc{Score}(x, y) = \lambda \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)}) + c$$

- Maximize this new scoring function on labeled data
  - Learn a HMM separately
  - Estimate $P(C_i = 1)$ separately by counting how many times the constraint is violated by the training data!
- A formal justification for optimizing the model and the penalty weights separately!

Summary

- Constrained Conditional Models: Computational Framework for global inference and a vehicle for incorporating knowledge
- Direct supervision for structured NLP tasks is **expensive**
- Indirect supervision is cheap and easy to obtain
- Constrained Conditional Models combine
  - Learning conditional models with using declarative expressive constraints
  - Within a constrained optimization framework
- diverse usage CCMs have already found in NLP
  - Significant success on several NLP and IE tasks (often, with ILP)