Novel Estimation Methods for Unsupervised Discovery of Latent Structure in Natural Language Text

Noah A. Smith

Hertz Foundation Fellow

Department of Computer Science /
Center for Language and Speech Processing
Johns Hopkins University

Advisor: Jason Eisner

→

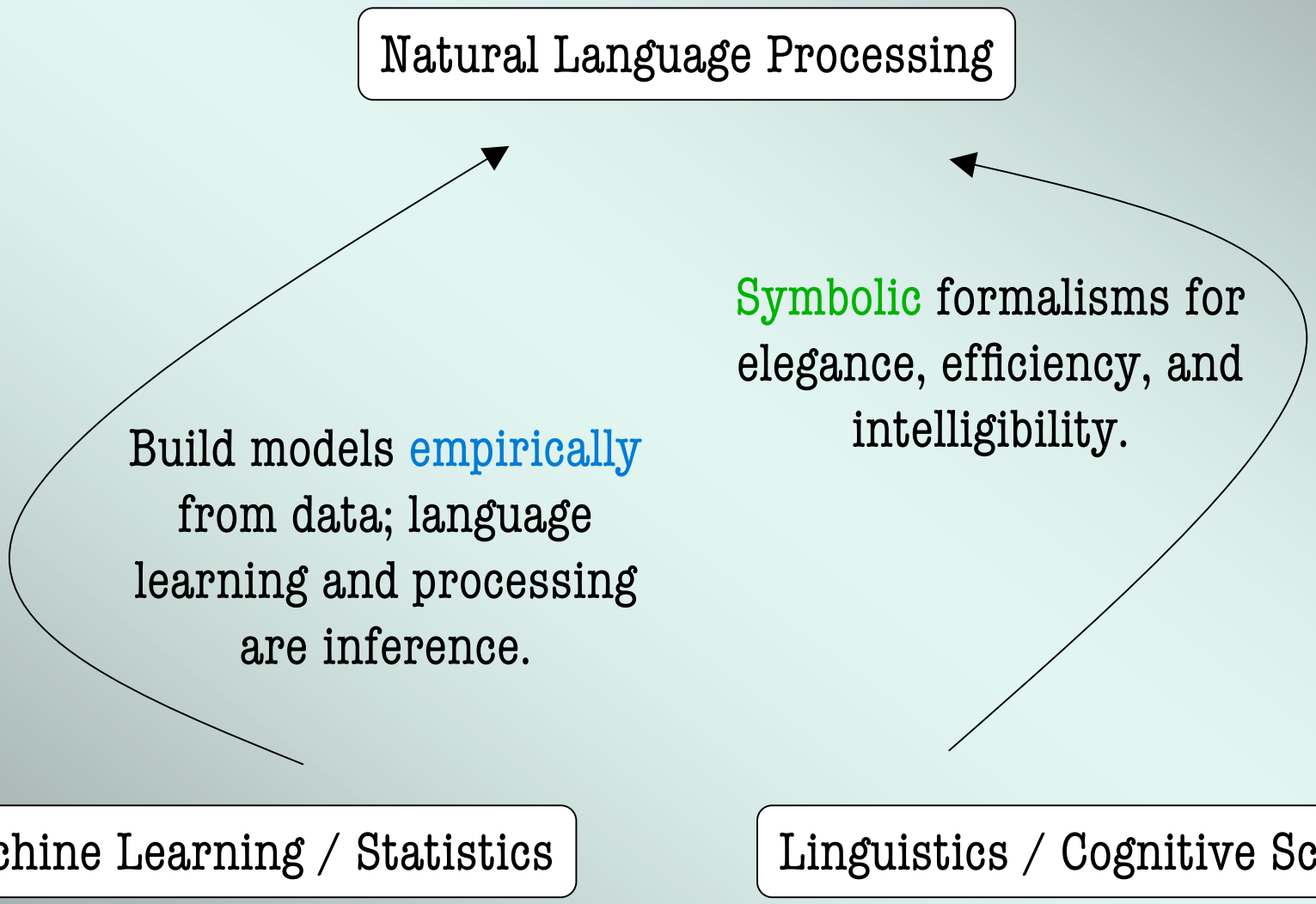
Assistant Professor

Language Technologies Institute /
Machine Learning Department
School of Computer Science
Carnegie Mellon University

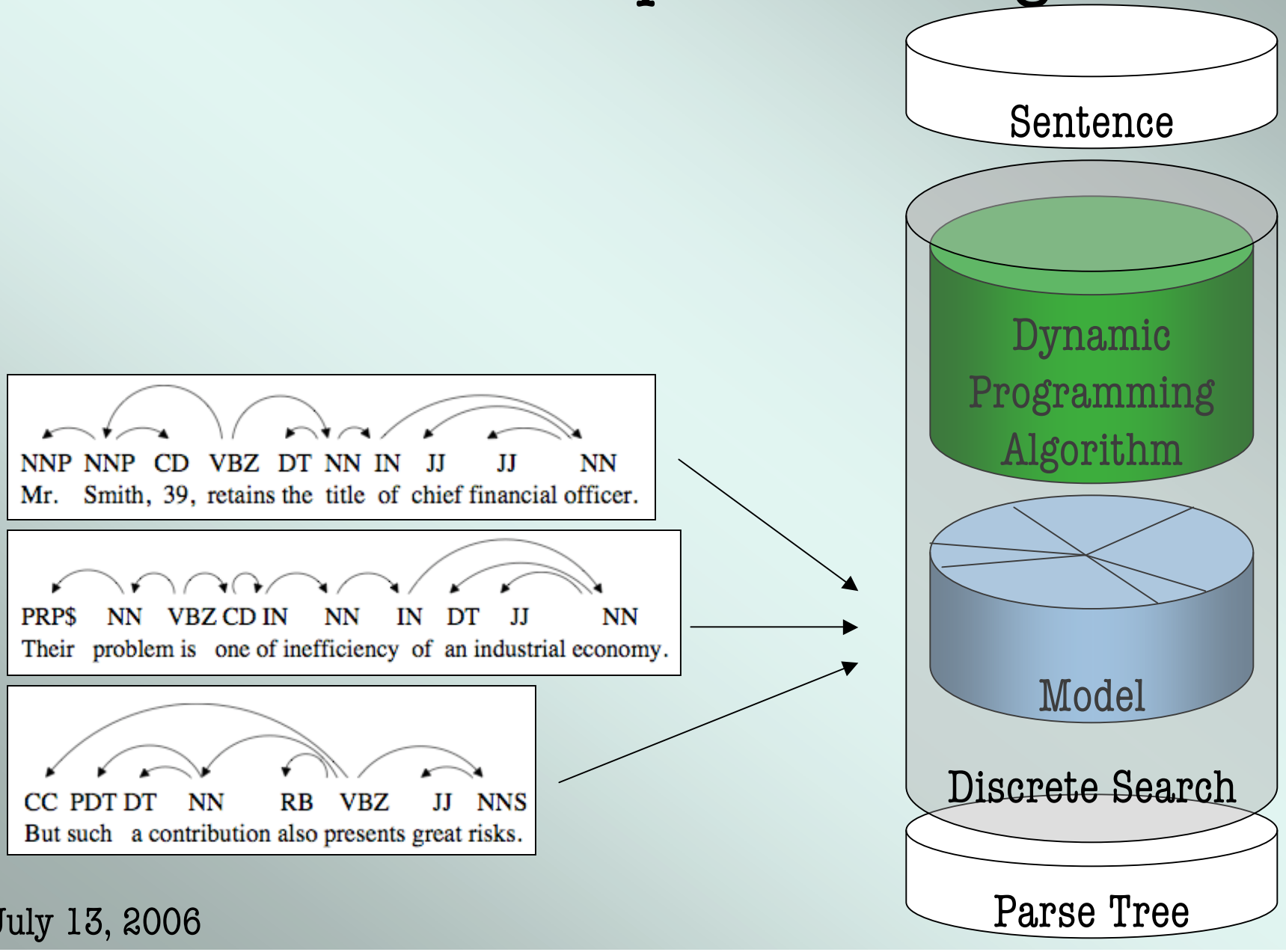
Situating the Thesis

- Too much information in the world!
- Most information is represented linguistically.
  - Most of us can understand one language or more.
- How can computers help?
- Can NLP systems “build themselves”?

Modern NLP


Natural Language Processing
Symbolic formalisms for elegance, efficiency, and intelligibility.
Build models empirically from data; language learning and processing are inference.
Machine Learning / Statistics
Linguistics / Cognitive Science

An Example: Parsing



Is Parsing Useful?

- Speech recognition (Chelba & Jelinek, 1998)
- Text correction (Shieber & Tao, 2003)
- Machine translation (Chiang, 2005)
- Information extraction (Viola and Narasimhan, 2005)
- NL interfaces to databases (Zettlemoyer & Collins, 2005)

Different parsers for different problems, and

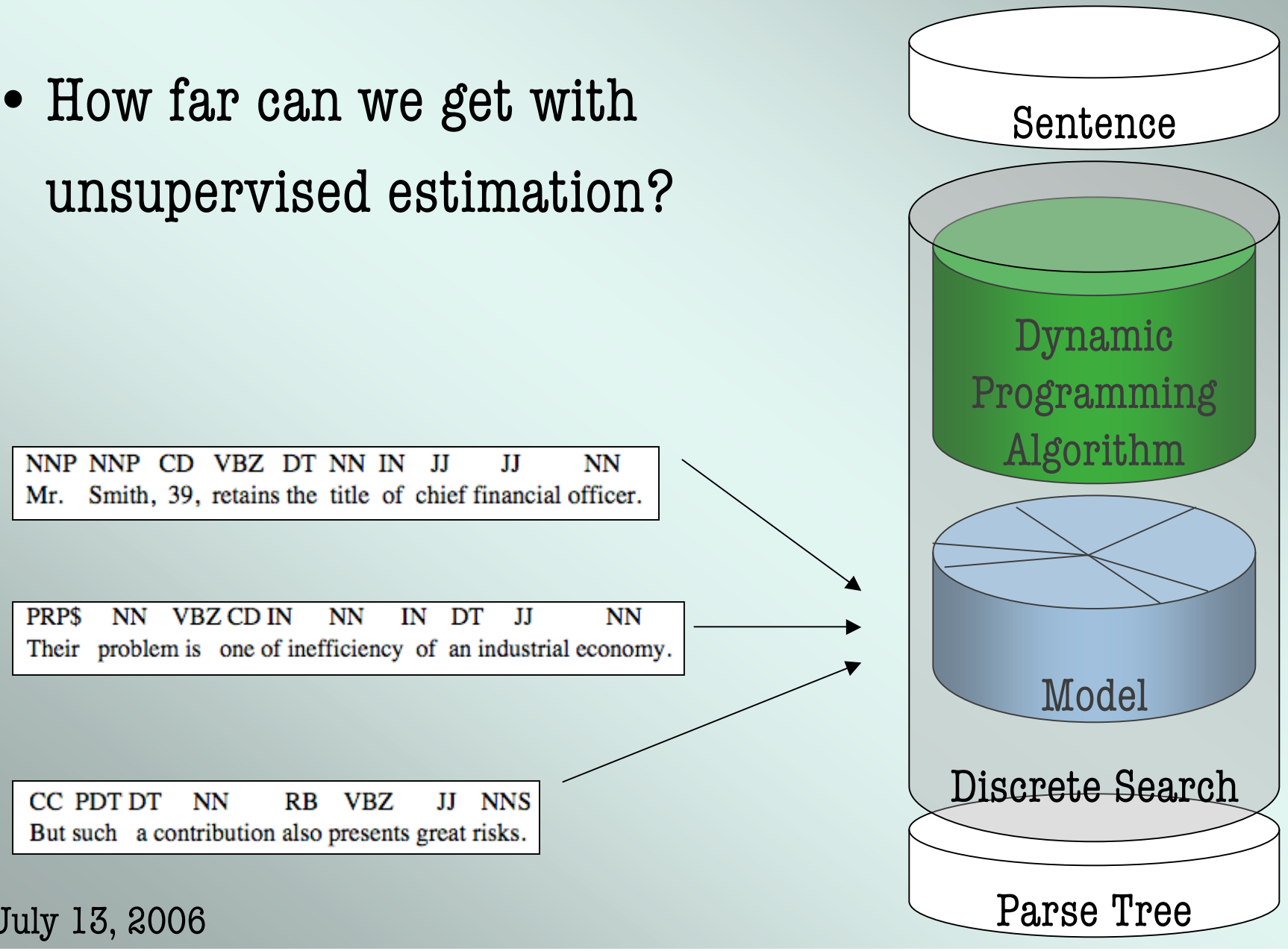
learning depends on the task.

The Current Bottleneck

- Empirical methods are great when you have **enough** of the **right** data.
- Reliable **unsupervised** learning would let us more cheaply:
  - Build models for new domains
  - Train systems for new languages
  - Explore new representations (hidden structures)
  - Focus more on applications

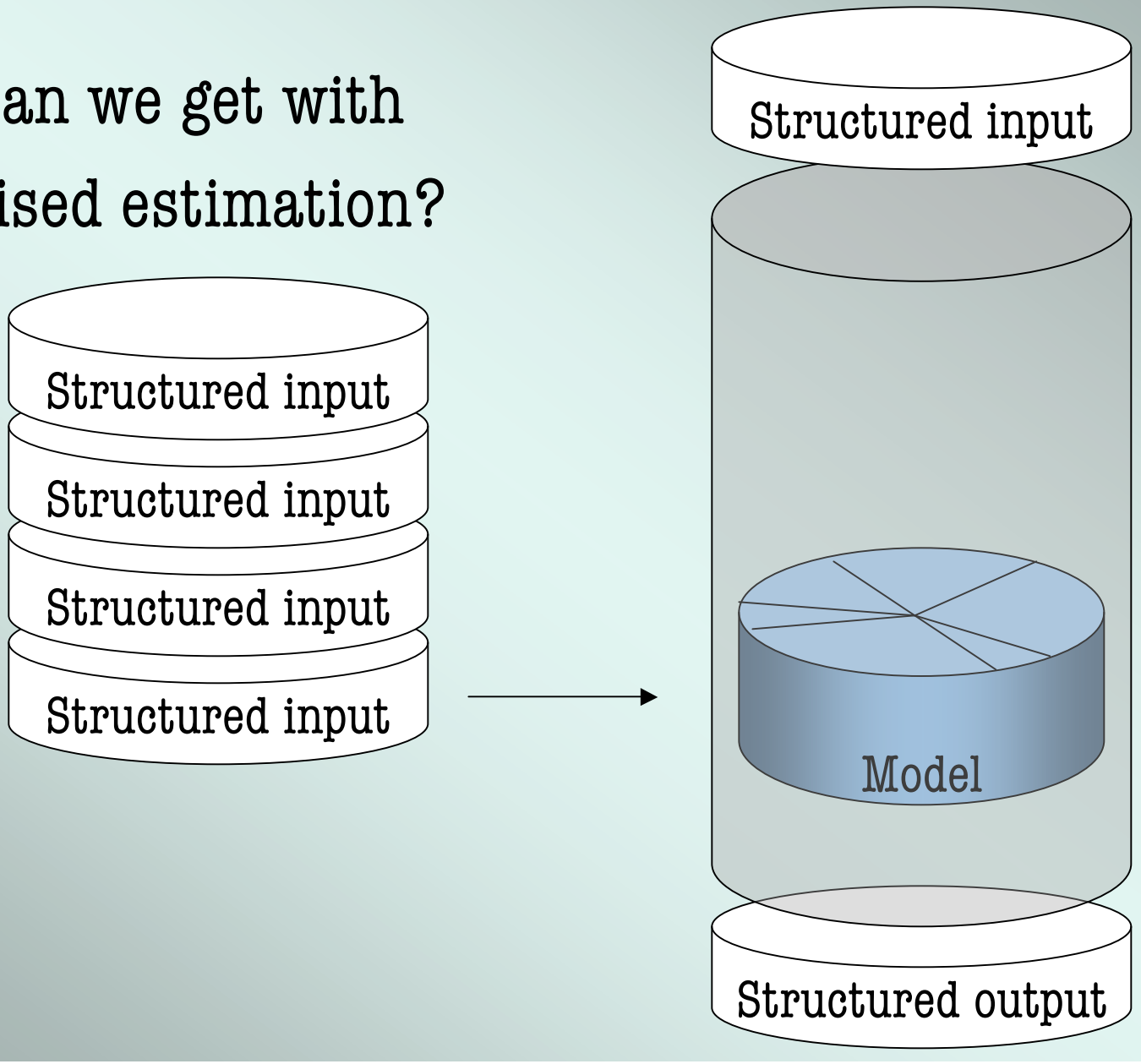
Central **Practical** Problem of the Thesis

- How far can we get with unsupervised estimation?



Deeper Problem

- How far can we get with unsupervised estimation?



Outline of the Talk

Learning To Parse	Learning = Optimizing a Function	Improving the Function	Multilingual Experiments
Chapters 1, 2	Chapter 3	Improving the Optimizer	Chapter 7
		Improving the Function and the Optimizer	
		Chapters 4, 5, 6	

- German
- English
- Bulgarian
- Mandarin
- Turkish
- Portuguese

Maximum Likelihood by EM

Contrastive Estimation

Deterministic Annealing

Structural Annealing

Dependency Parsing

- Underlies *many* linguistic theories
- Simple model & algorithms (Eisner, 1996)
- Projectivity constraint → context-free

  (cf. McDonald et al., 2005)
- Unsupervised learning:
  - Carroll & Charniak (1992)
  - Yuret (1998)
  - Paskin (2002)
  - Klein & Manning (2004)

Applications:

- Relation extraction

  Culotta & Sorenson (2004)
- Machine translation

  Ding & Palmer (2005)
- Language modeling

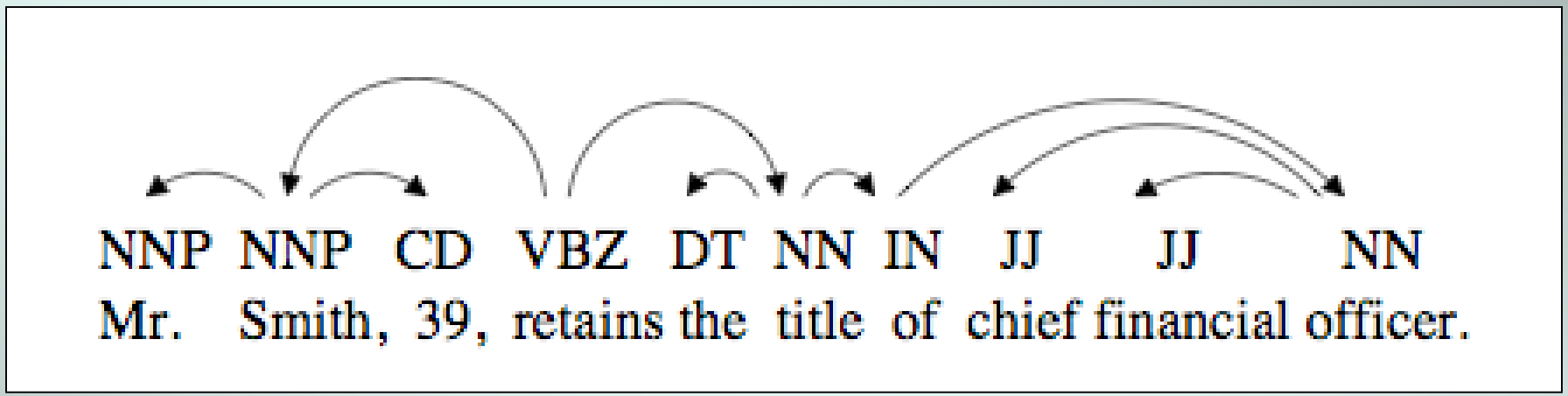
  Chelba & Jelinek (1998)
- All kinds of lexical learning

  Lin & Pantel (2001), *inter alia*
- Semantic role labeling

  Carerras & Marquez (2004)
- Textual entailment

  Raina et al. (2005), *inter alia*

A Dependency Tree



Our Model A ("DMV")

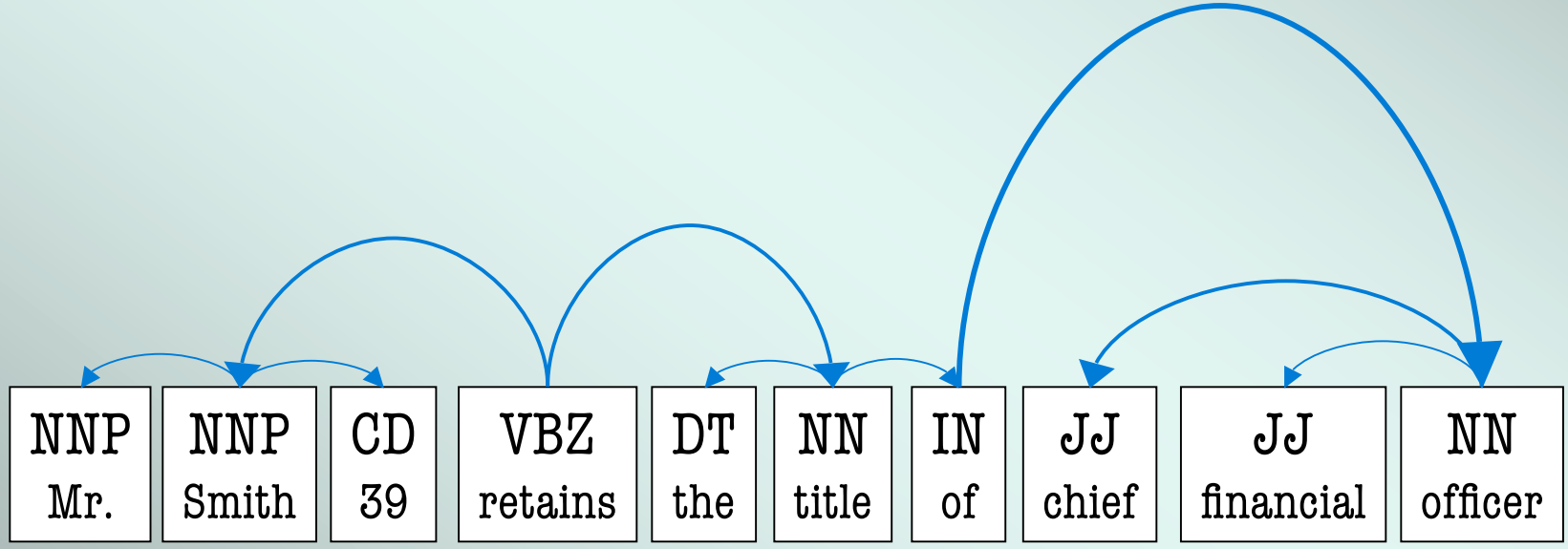
- Expressible as a SCFG
- Can be viewed as a log-linear model with these features:
  - Root tag is U.
  - Tag U has a child tag V in direction D.
  - Tag U has no children in direction D.
  - Tag U has at least one child in direction D.
  - Tag U has only one child in direction D.
  - Tag U has a non-first child in direction D.

Example Derivation of the Model

Root tag is VBZ. | VBZ has a right child. | VBZ has NN as right child.

VBZ has only 1 right child. | VBZ has a left child. | VBZ has NNP as left child.

VBZ has only 1 left child. | NNP has a right child. | NNP has CD as right child.



Klein & Manning, 2004

Stochastic and Log-linear CFGs

sentence, tree

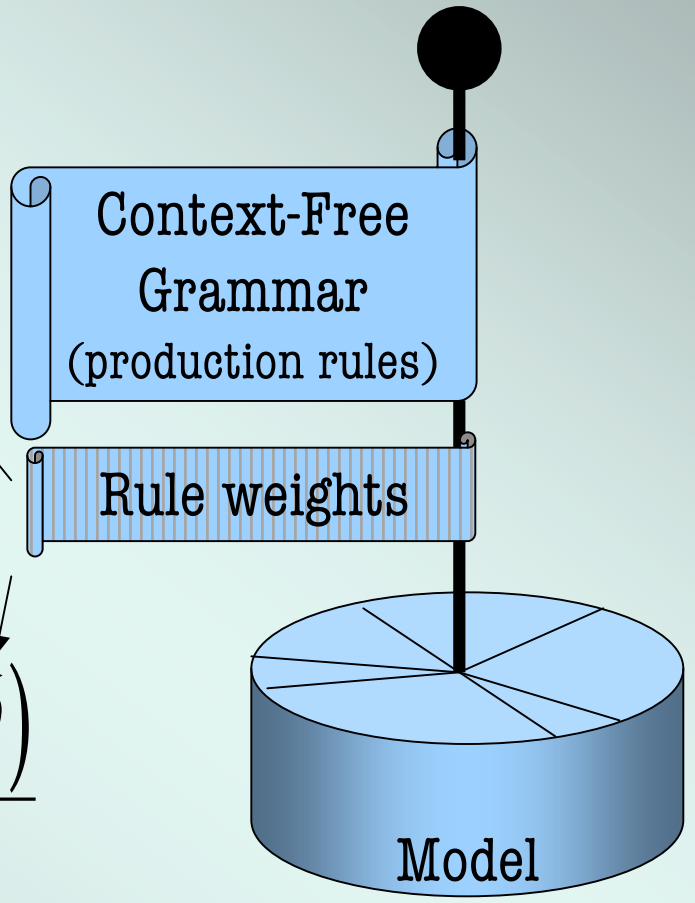
$$p_{\vec{\theta}}(\mathbf{x},\mathbf{y}) = \prod_{\text{derivation feature tokens } \mathbf{r}} e^{\theta_{\mathbf{r}}}$$

$$= \prod_{\text{rules } \mathbf{r}} e^{f_{\mathbf{r}}(\mathbf{x},\mathbf{y})\theta_{\mathbf{r}}}$$

$$= \exp\left(\vec{f}(\mathbf{x},\mathbf{y}) \cdot \vec{\theta}\right)$$

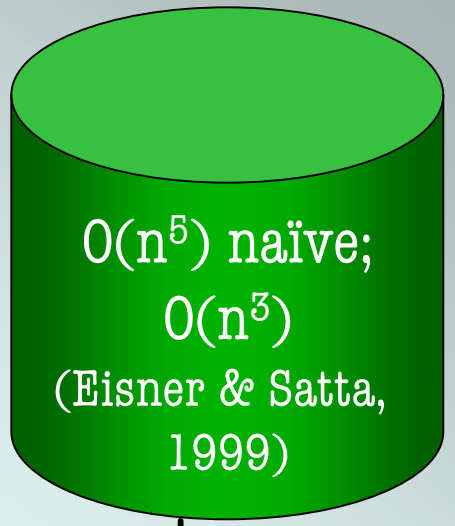
$$\ddot{p}_{\vec{\theta}}(\mathbf{x},\mathbf{y}) \stackrel{\text{def}}{=} \frac{\exp\left(\vec{f}(\mathbf{x},\mathbf{y}) \cdot \vec{\theta}\right)}{\ddot{Z}_{\vec{\theta}}(\mathcal{W})}$$

Set of all sentences and their trees



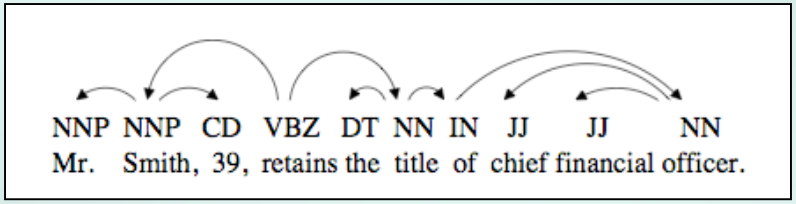
Model A is Very Simple!

- Connected, directed trees over tags.
  - Tag-tag relationships
  - Affine valency model

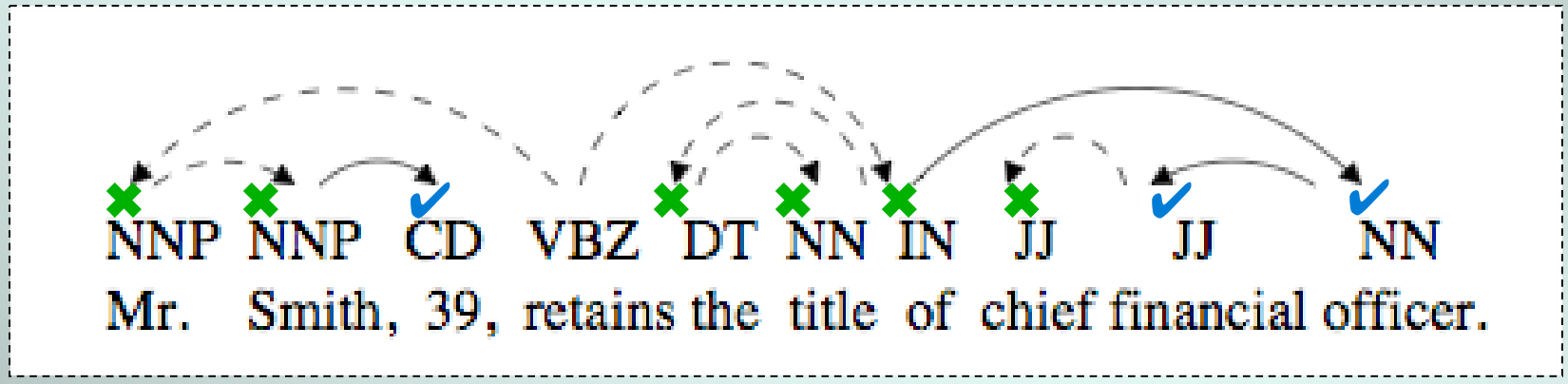


- No sister effects, even on same side of parent.
- No grandparent effects.
- No lexical selection, subcategorization, anything.
- No distance effects.

Evaluation



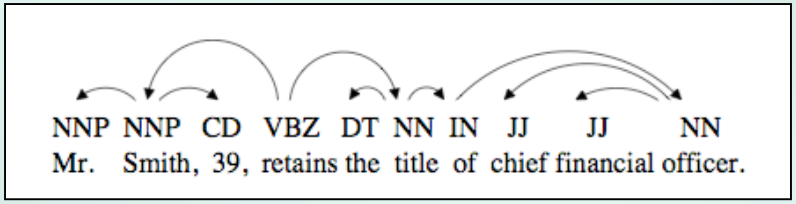
Treebank tree (gold standard)



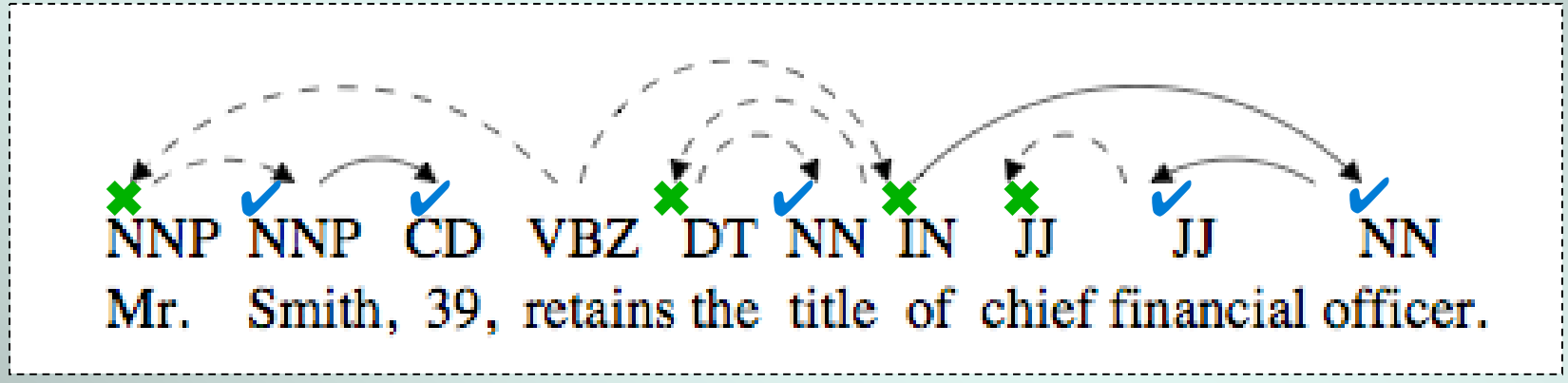
hypothesis tree

Accuracy = 3 / (3 + 6) = 33.3%

Evaluation



Treebank tree (gold standard)



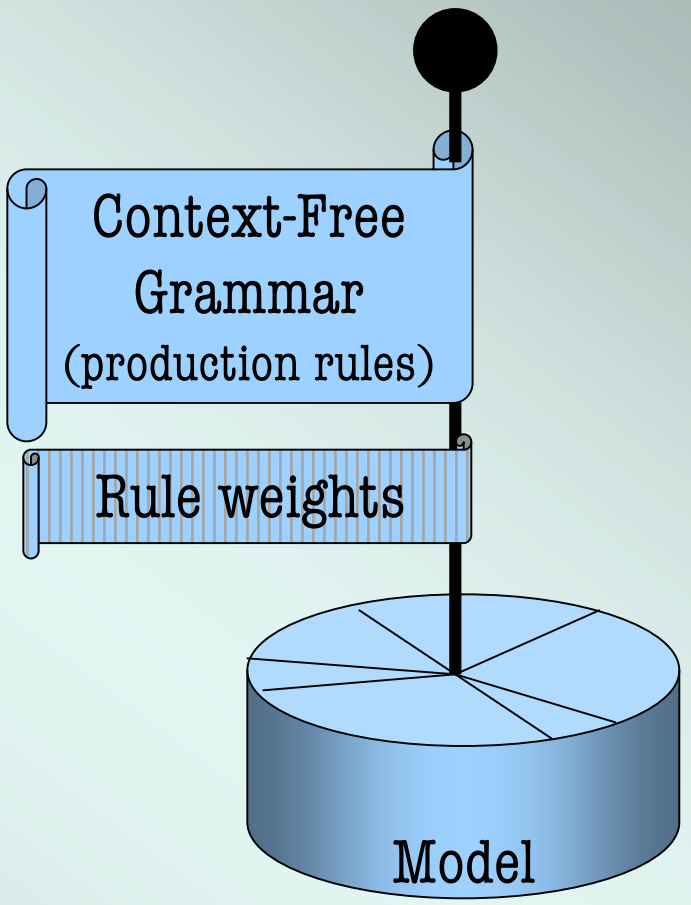
hypothesis tree

Undirected Accuracy = 5 / (5 + 4) = 55.5%

Fixed Grammar, Learned Weights

All dependency trees on all tag sequences can be derived.

How do we learn the weights?



$$\vec{\theta}$$

Maximum Likelihood Estimation

$$\max_{\vec{\theta}} p_{\vec{\theta}}(\text{observed data})$$

Supervised training: "observed data" are sentences with trees

$$\max_{\vec{\theta}} \left[\prod_{i=1}^{n} p_{\vec{\theta}}(\mathbf{x}_i, \mathbf{y}_i) \right]$$

Independence among examples

For PCFGs, **closed form** solution

Requires numerical optimization

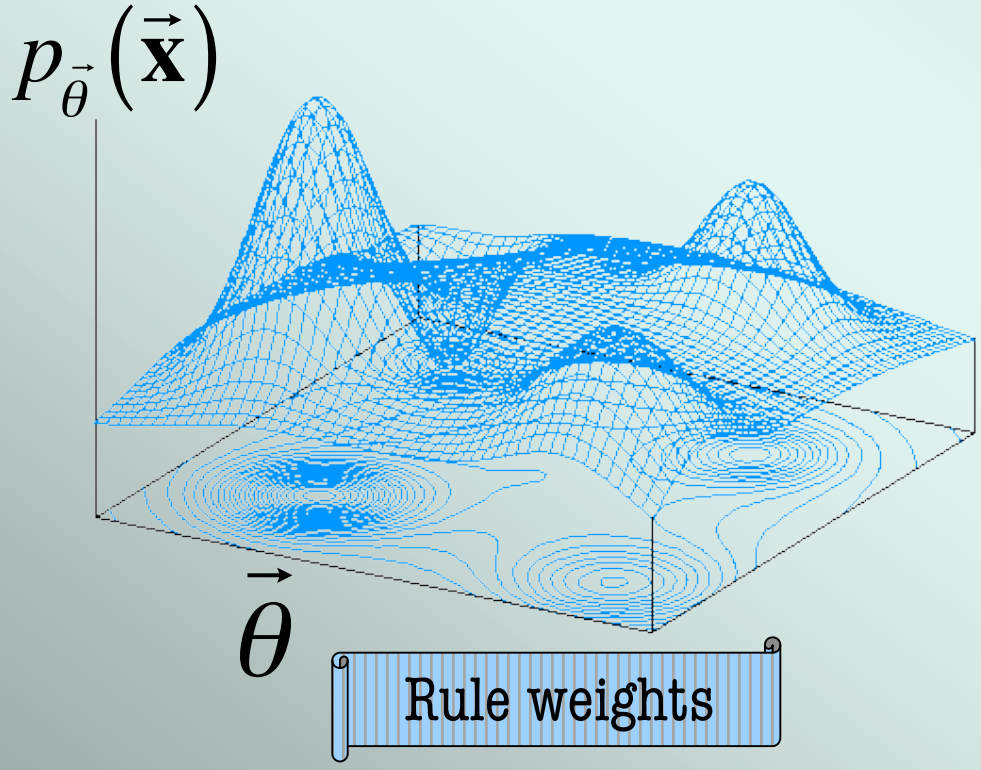
Unsupervised training: "observed data" are sentences

$$\max_{\vec{\theta}} \left[\prod_{i=1}^{n} \sum_{\mathbf{y}} p_{\vec{\theta}}(\mathbf{x}_i, \mathbf{y}) \right]$$

Marginalize over trees

Expectation-Maximization

- Hillclimber for the likelihood function.
- Quality of the estimate depends on the starting point.

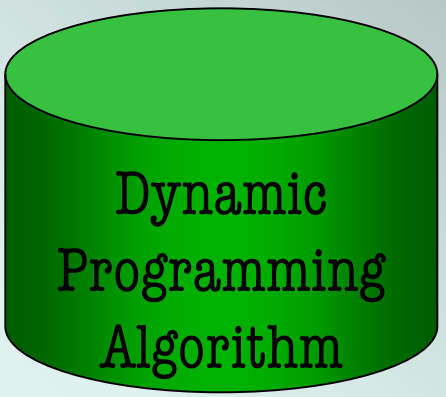


EM for Stochastic Grammars

- **E step**

  Compute expected rule counts for each sentence:

$$c_{\mathbf{r}} \overset{+}{\leftarrow} \mathbf{E}_{p_{\vec{\theta}^{(i)}}}\left[f_{\mathbf{r}}\left(\mathbf{x}_j, \mathbf{Y}\right)\right]$$



- **M step**

  Renormalize counts into multinomial distributions.

$$\theta_{\mathbf{r}}^{(i+1)} = \log\left(c_{\mathbf{r}}\right) - Z$$

Experiment

- WSJ10: 5300 part-of-speech sequences of length ≤10
- Words ignored, punctuation stripped
- **Three** initializers:
  - Zero: all weights set to zero
  - K&M: Klein and Manning (2004), roughly
  - Local: Slight variation on K&M, more smoothed
- 530 test sentences

Experimental Results: MLE/EM

		Accuracy (%)	Undirected Accuracy (%)	Iterations	Cross-Entropy
Attach-Left		22.6	62.1	0	-
Attach-Right		39.5	62.1	0	-
Zero		22.7	58.8	49	26.07
MLE/EM	K&M	41.7	62.1	62	25.16
	Local	22.8	58.9	49	26.07

Dirichlet Priors for PCFG Multinomials

- Simplest conceivable smoothing: add-λ
- Slight change to **M step**:

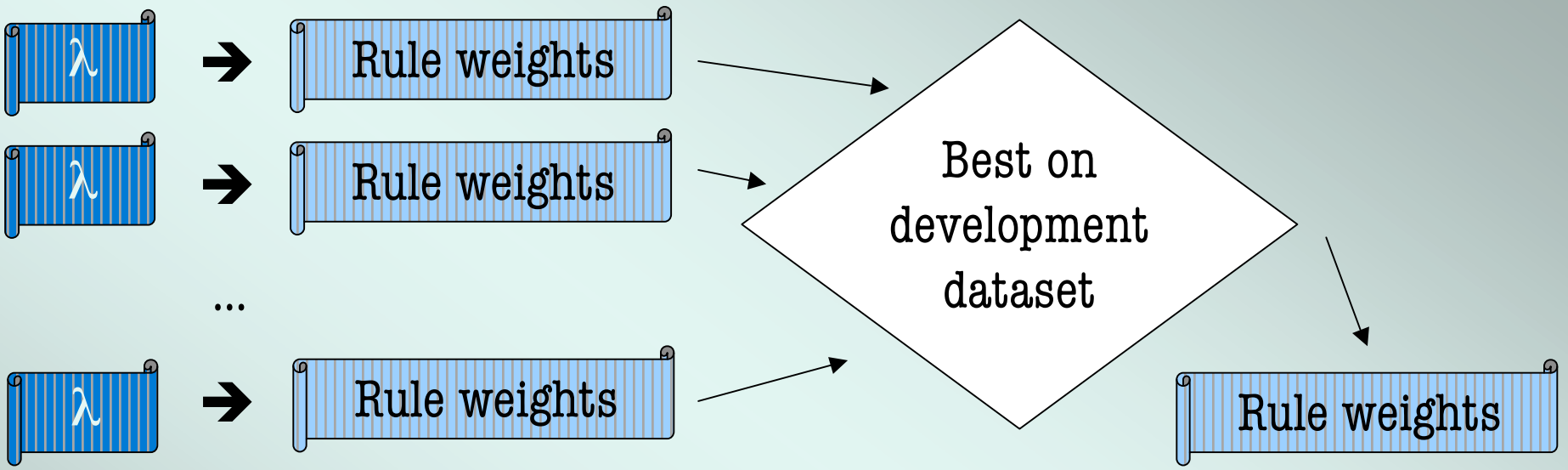
As if we saw each event an additional λ times.

$$\theta_{\mathbf{r}}^{(i+1)} = \log\left(c_{\mathbf{r}} + \lambda\right) - Z$$

This is **Maximum a Posteriori** estimation, or "MLE with a prior."

How to pick λ?

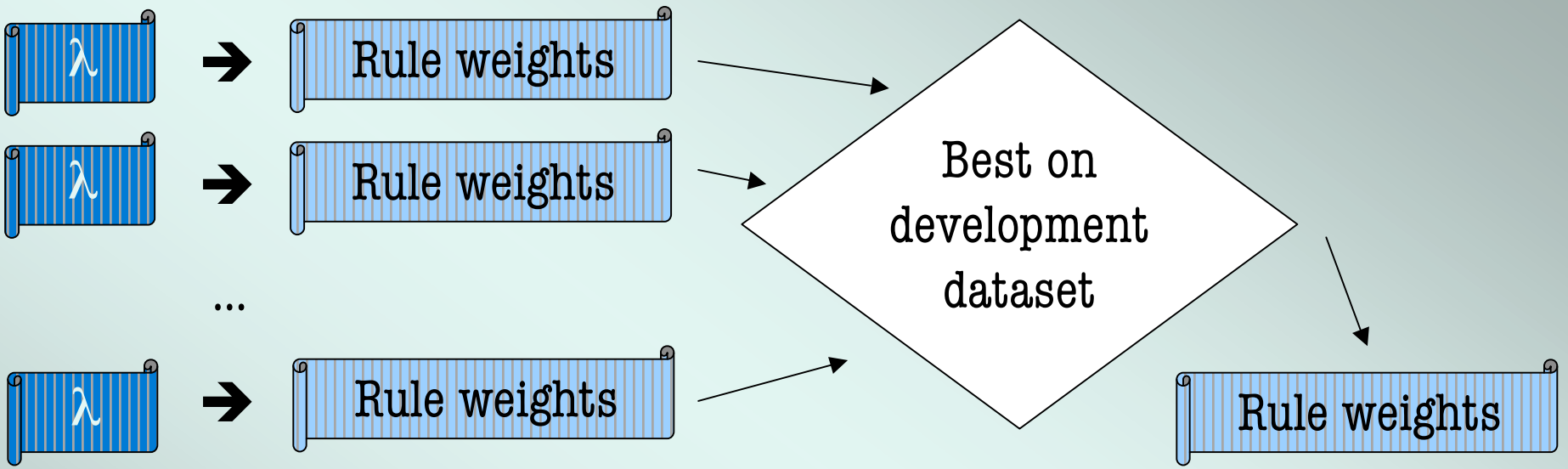
Model Selection



Supervised selection: best **accuracy** on **annotated** development data (presented in talk)

Unsupervised selection: best **likelihood** on **unannotated** development data (given in thesis)

Model Selection



Advantages:

- Can re-select later for different applications/datasets.

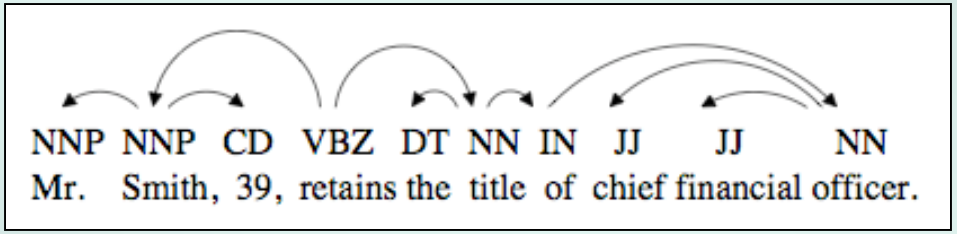
Disadvantages:

- Lots of models to train!
- Still have to decide which λ values to train with.

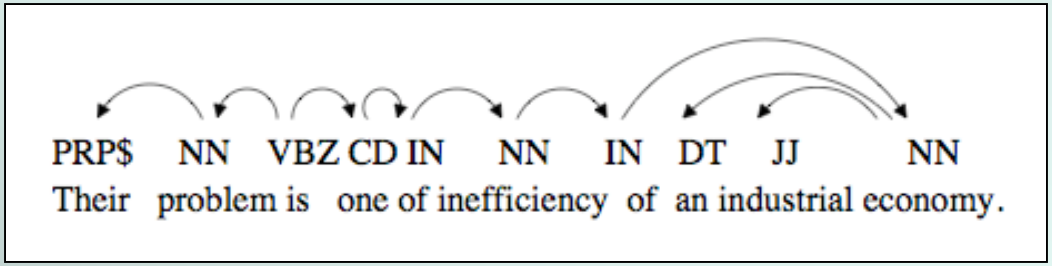
Experimental Results: MAP/EM

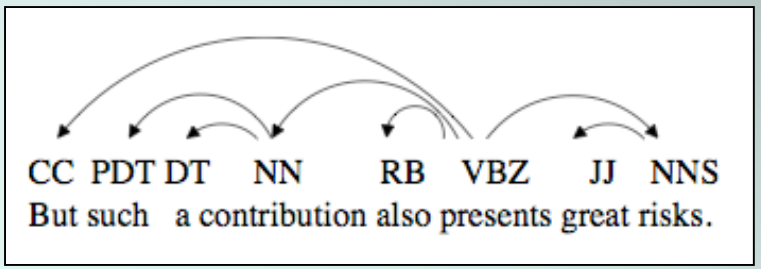
	Accuracy (%)	Undirected Accuracy (%)	Iterations	Cross-Entropy
Attach-Right	39.5	62.1	0	-
Zero	22.7	58.8	49	26.07
MLE/EM K&M	41.7	62.1	62	25.16
Local	22.8	58.9	49	26.07
MAP/EM (sel. λ, initializer)	41.6	62.2	49	25.54

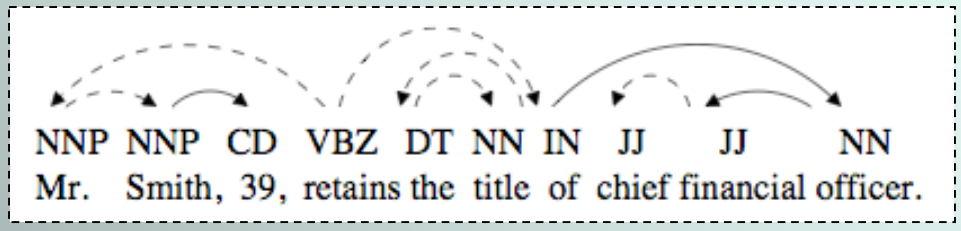
"Typical" Trees



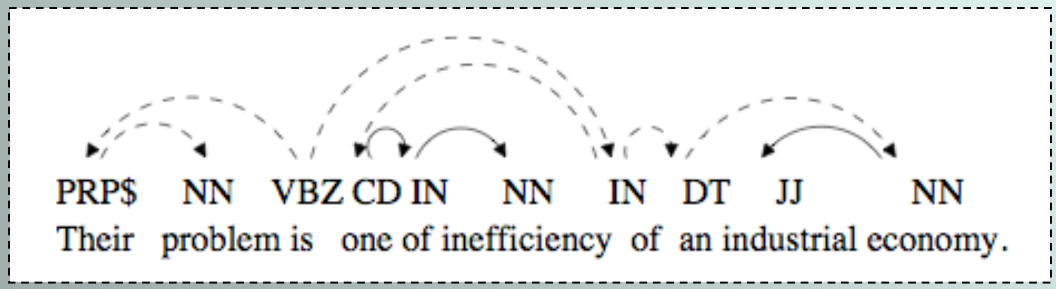
Treebank

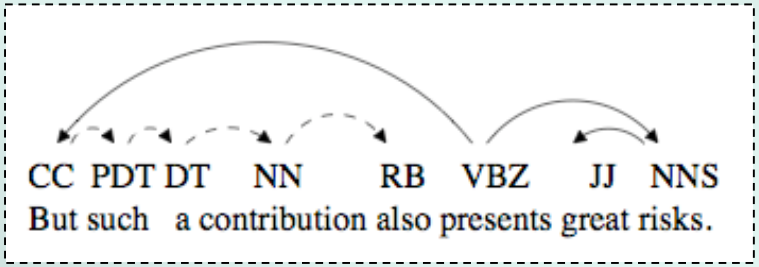




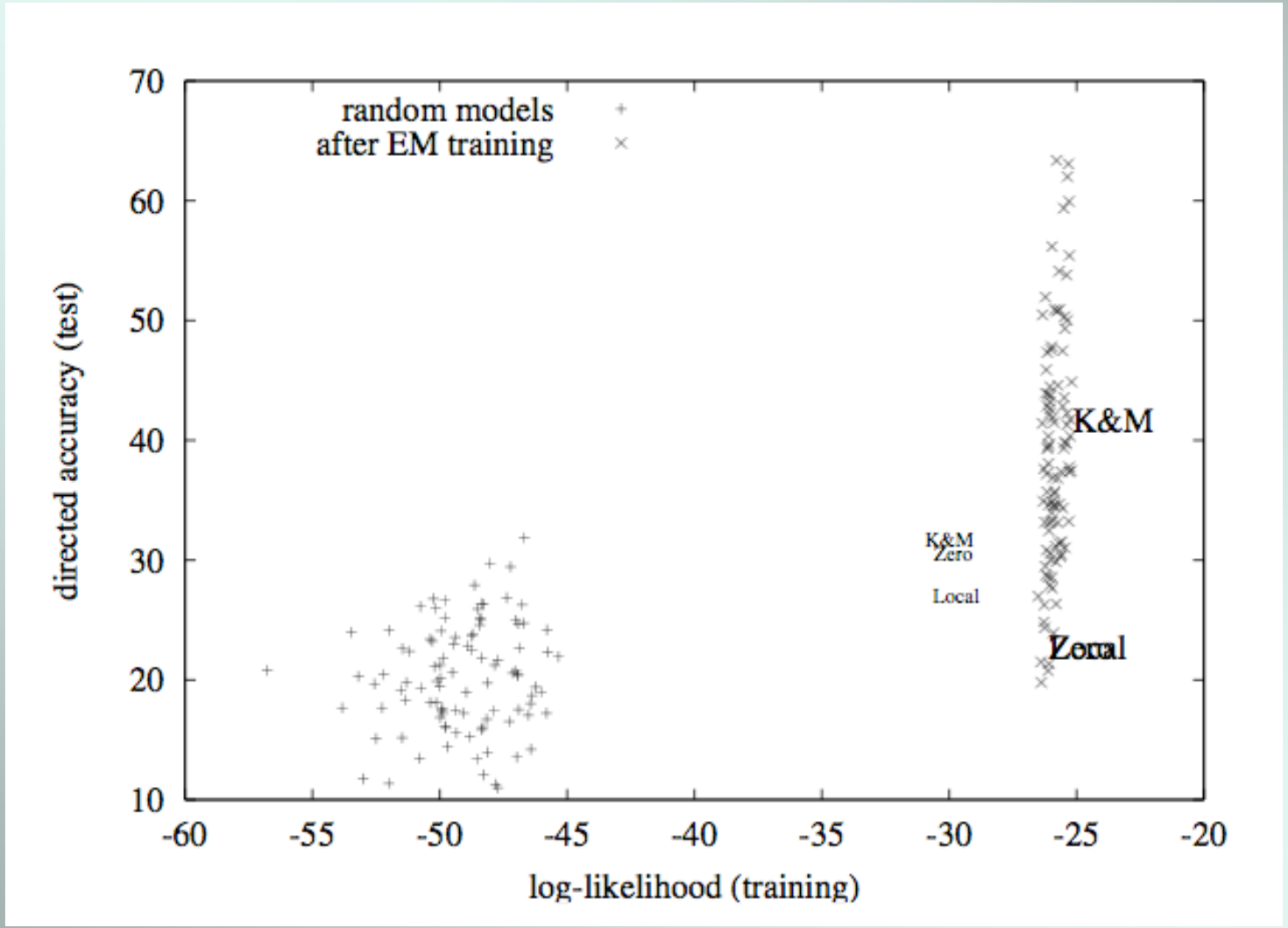


learned model

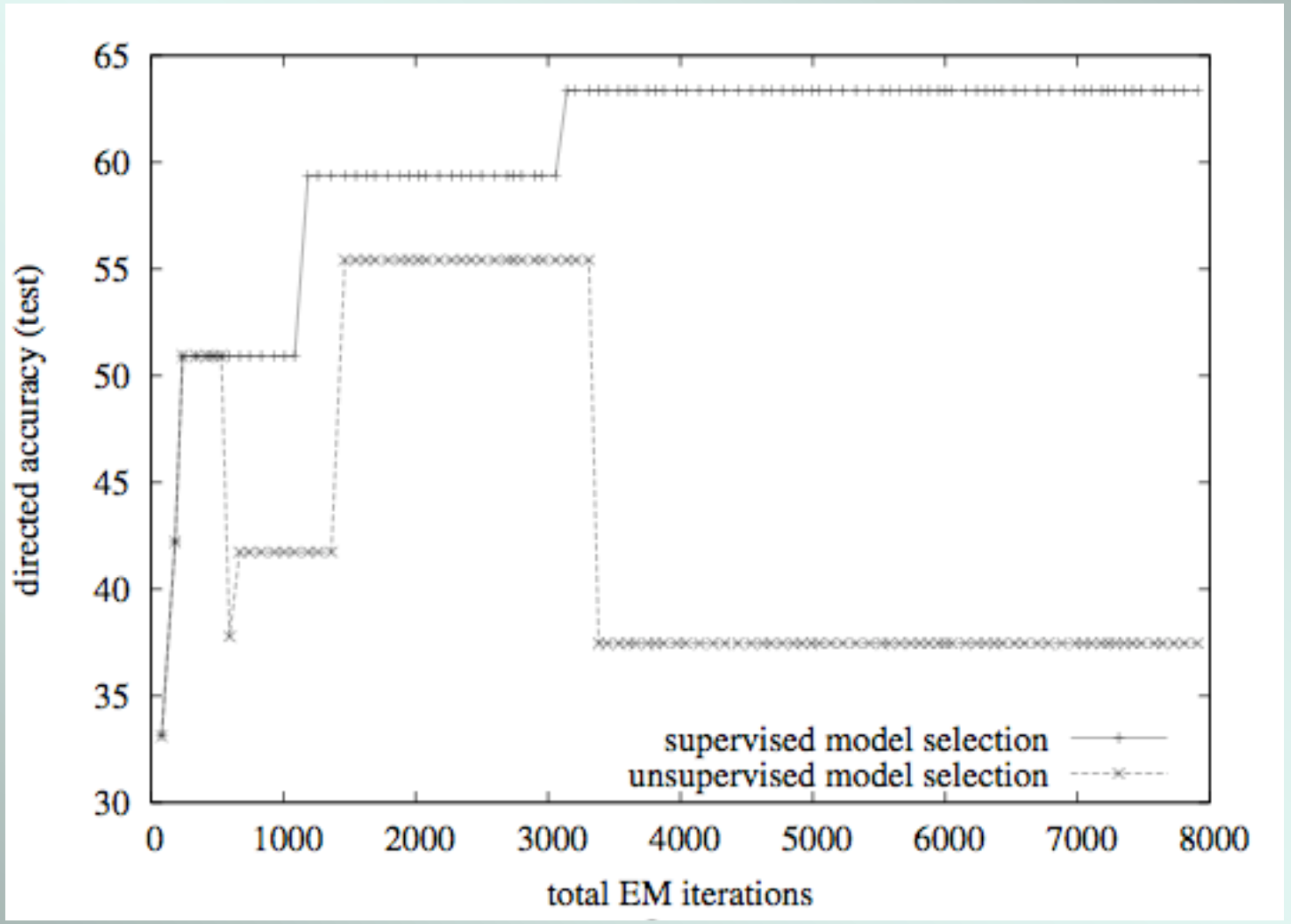




Good and Bad News About Likelihood



Selection over Random Initializers



On Aesthetics

✓ Hyperparameters should be interpretable.

✗ Reasonable initializers should perform reasonably.

- These are a form of domain knowledge that should help, not hurt performance.
- If all else fails, “Zero” (maxent) initializer should perform well.

Can we have both?

Where are we?

Learning To Parse	Learning = Optimizing a Function	Improving the Function	☐
☑	☑		

Likelihood as Teacher

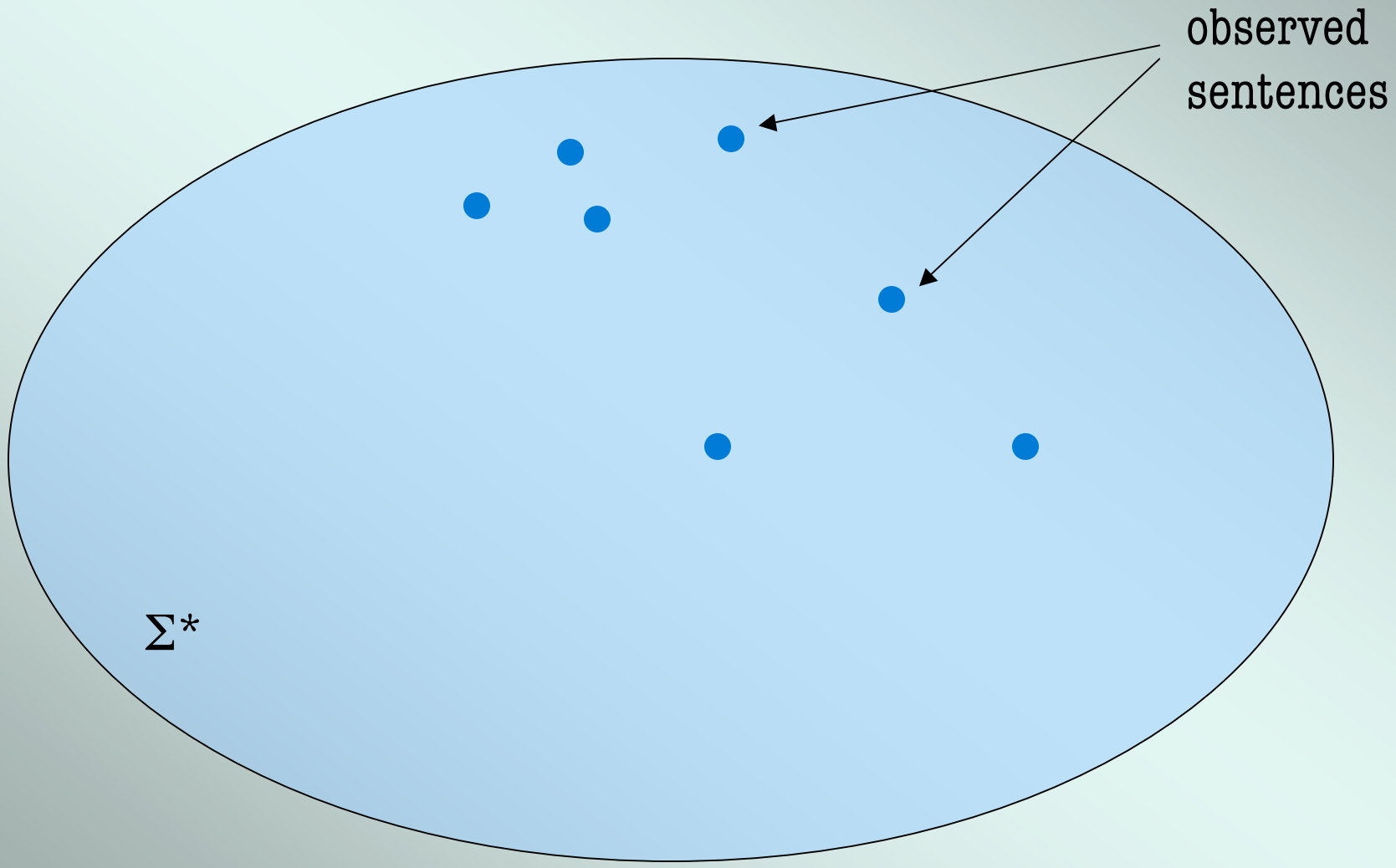
Red leaves don't hide blue jays.

Mommy doesn't love you.

Dishwashers are a dime a dozen.

Dancing granola doesn't hide blue jays.

Probability Allocation



What We'd Like

- Focus on the model on the properties of the data that will lead to an explanation of syntax.

Red leaves don't hide blue jays.

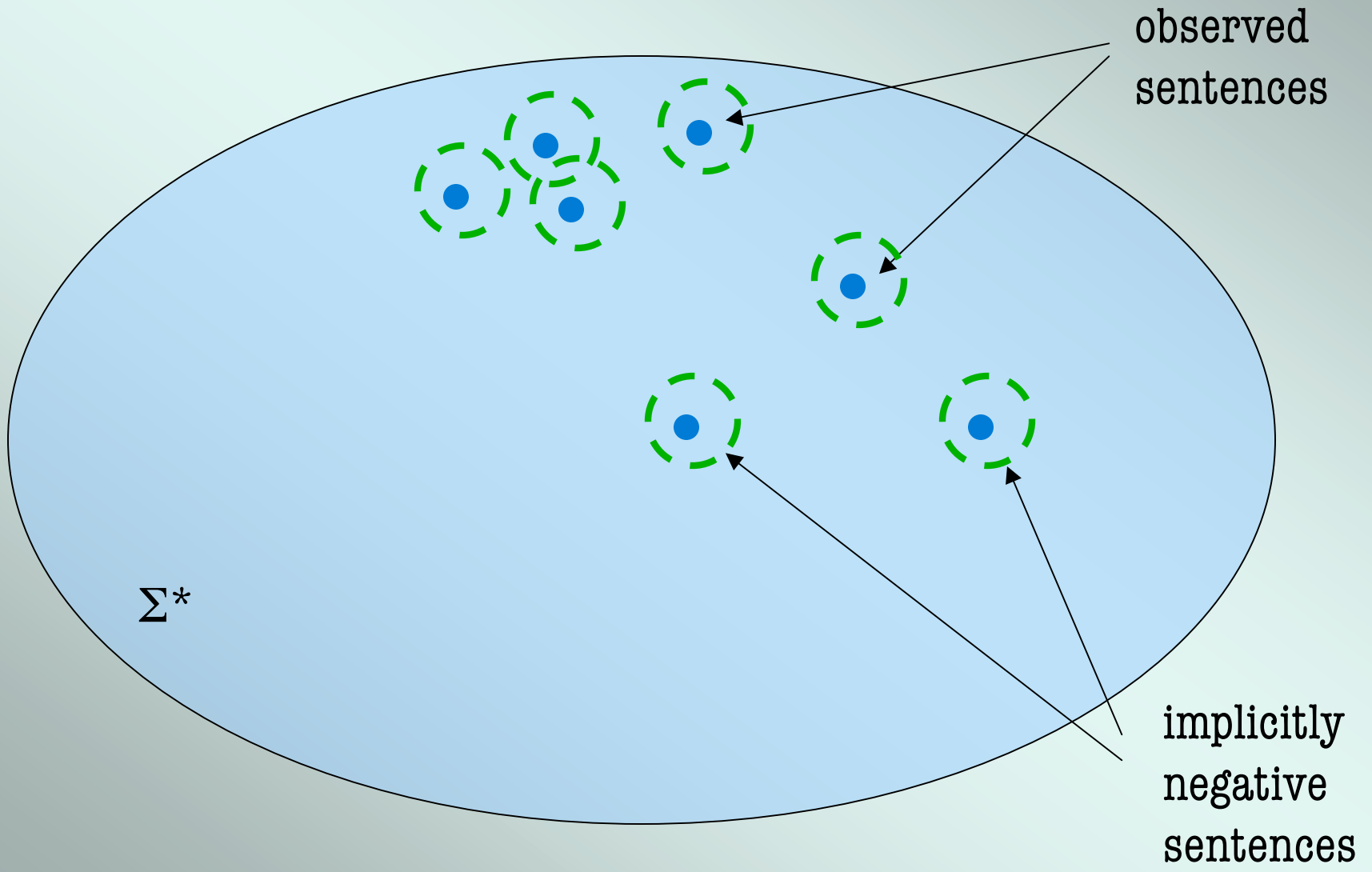
*Jays blue hide don't leaves red.

*Blue don't hide jays leaves red.

*Hide don't blue jays red leaves.

- Idea: train model to explain **order** but not content.

Contrastive Estimation
(Smith & Eisner, 2005)



Maximum Likelihood Estimation vs. Contrastive Estimation

MLE/MAP: observed data are Sentences, neighborhood is Σ^*

Require numerical optimization

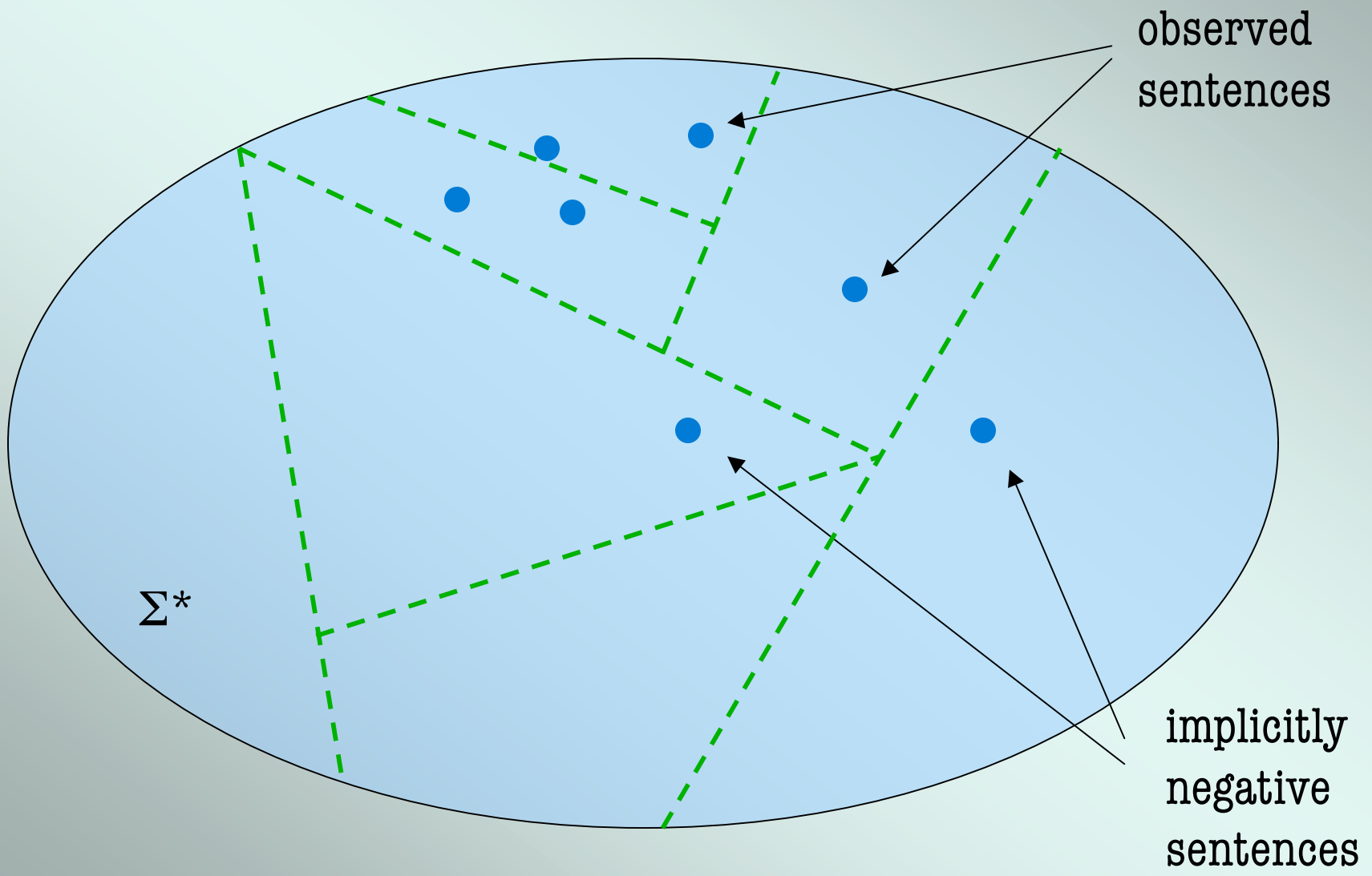
CE: observed data are sentences, neighborhood is ...?

$$\max_{\vec{\theta}} \left[\prod_{i=1}^{n} \sum_{\mathbf{y}} p_{\vec{\theta}}(\mathbf{x}_i, \mathbf{y}) \right]$$

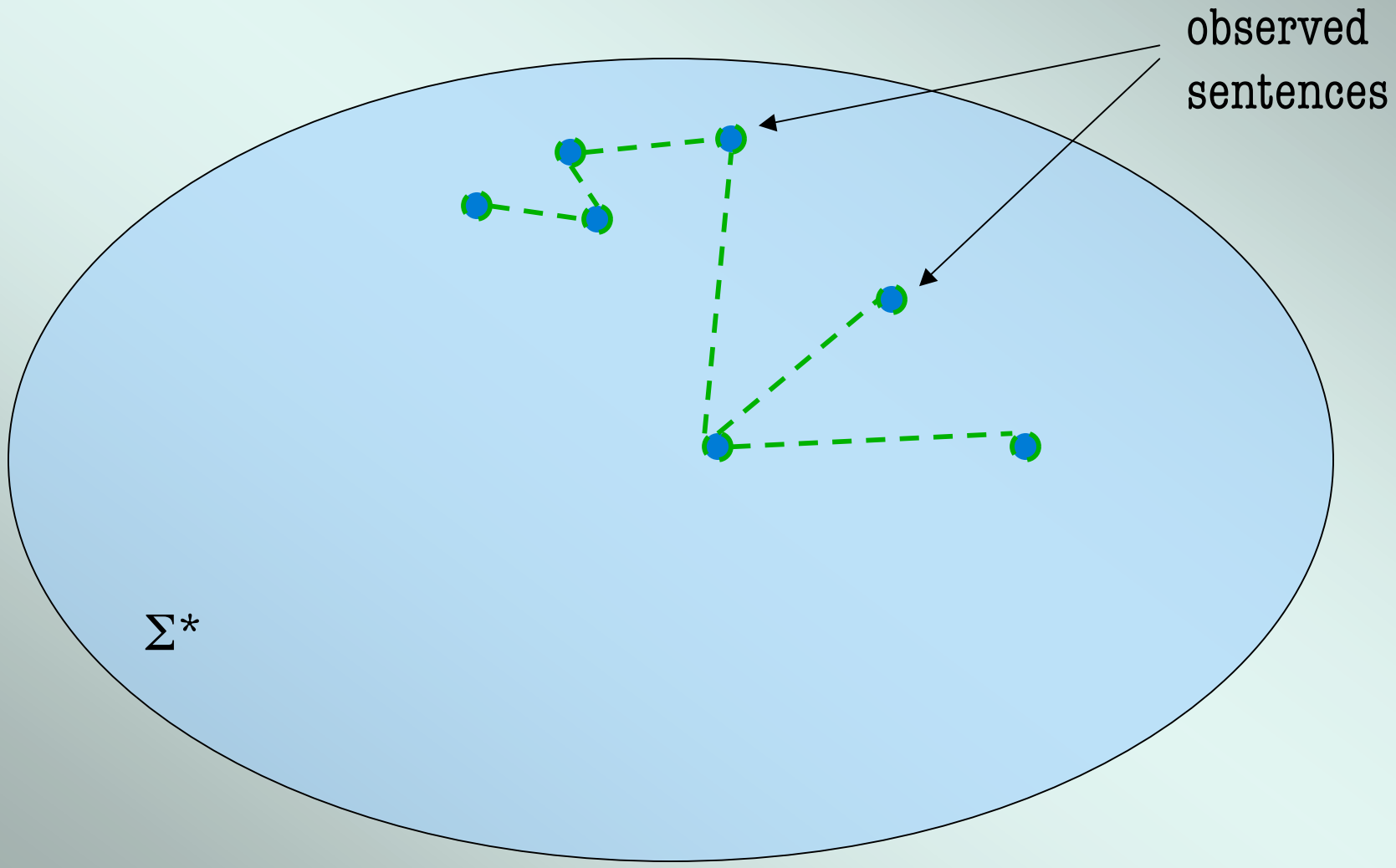
$$\max_{\vec{\theta}} \left[\prod_{i=1}^{n} \frac{\sum_{\mathbf{y}} p_{\vec{\theta}}(\mathbf{x}_i, \mathbf{y})}{\sum_{\mathbf{x} \in \mathcal{N}(\mathbf{x}_i)} \sum_{\mathbf{y}} p_{\vec{\theta}}(\mathbf{x}, \mathbf{y})} \right]$$

$$= \max_{\vec{\theta}} \left[\prod_{i=1}^{n} p_{\vec{\theta}}\left(\mathbf{X} = \mathbf{x}_i \mid \mathbf{X} \in \mathcal{N}(\mathbf{x}_i)\right) \right]$$

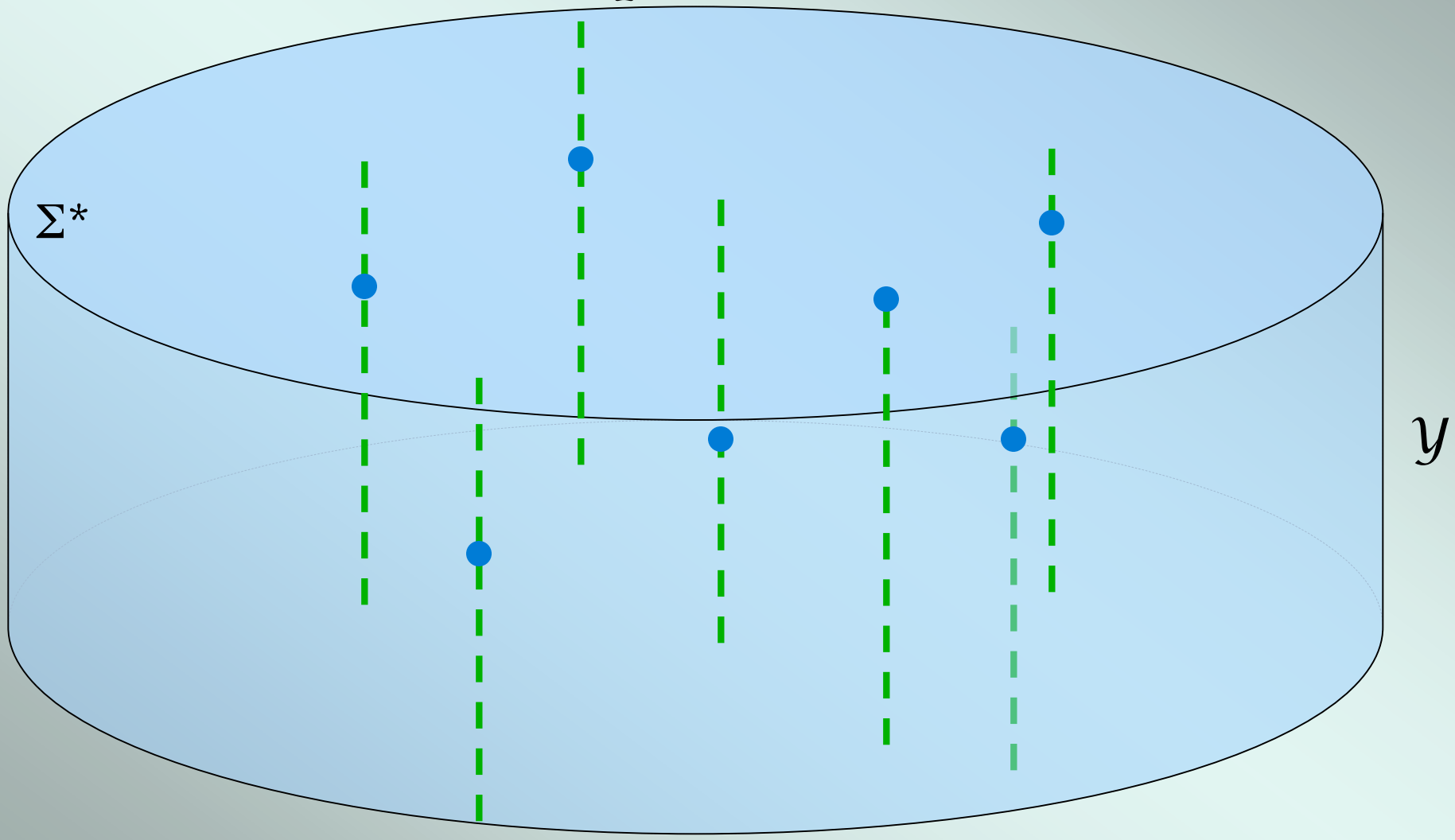
Partition Neighborhood = Conditional EM



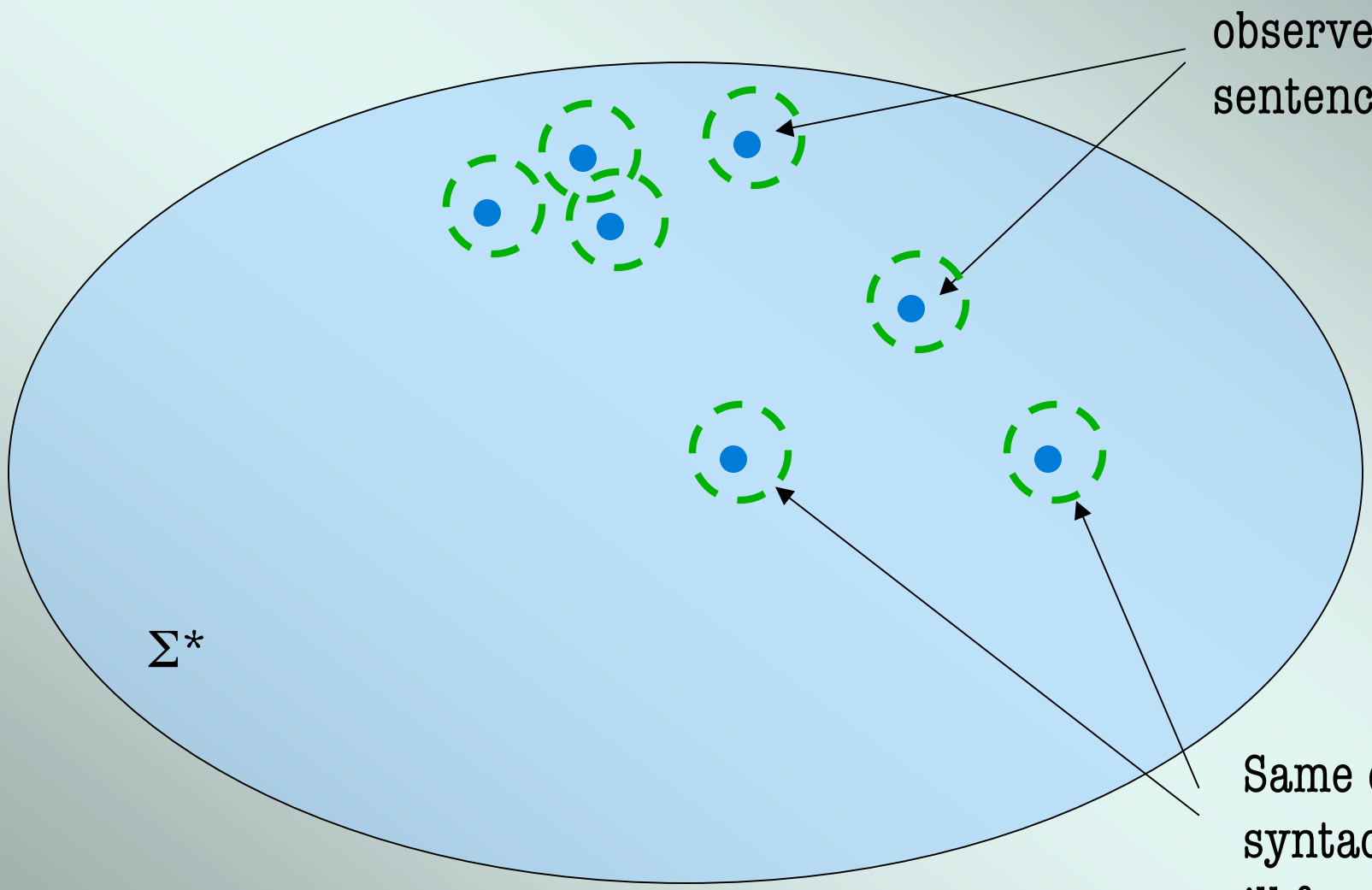
Riezler's (1999) Approximation



Analogy to Conditional Estimation (Supervised)



CE for Syntax



CE as Teacher

Red leaves don't hide blue jays.

Leaves red don't hide blue jays.

Red don't leaves hide blue jays.

Red leaves hide don't blue jays.

Optimizing Contrastive Likelihood

$$F\left(\vec{\theta}\right) = \left[\sum_{i=1}^{n} \log p_{\vec{\theta}}\left(\mathbf{X} = \mathbf{x}_i\right) - \log p_{\vec{\theta}}\left(\mathbf{X} \in \mathcal{N}\left(\mathbf{x}_i\right)\right)\right]$$

$$\frac{\partial F}{\partial \theta_{\mathbf{r}}} = \left[\sum_{i=1}^{n} \mathbf{E}_{p_{\vec{\theta}}}\left[f_{\mathbf{r}}\left(\mathbf{x}_i, \mathbf{Y}\right)\right] - \mathbf{E}_{p_{\vec{\theta}}}\left[f_{\mathbf{r}}\left(\mathbf{X}, \mathbf{Y}\right) \mid \mathbf{X} \in \mathcal{N}\left(\mathbf{x}_i\right)\right]\right]$$

Expected count
Of rule **r** in sentence *i*

Expected count
Of rule **r** in neighborhood *i*

Gradient ascent,
Conjugate gradient,
LMVM/LBFGS

- What about the simplex constraints?
- How to make the second term efficient?

Getting Rid of Simplex Constraints

- PCFGs represent distributions p(tree, sentence).
- So do some WCFGs - if you can normalize.

$$\ddot{Z}_{\vec{\theta}}(\mathcal{W})$$

(Requires a finite sum over **all** derivation scores.)

PCFGs and WCFGs represent the **same family**.

Chi (1999)

Abney et al. (1999)

- PCFGs represent p(tree | sentence).
- So do some WCFGs - if you can normalize.

$$\ddot{Z}_{\vec{\theta}}(\mathbf{x})$$

(Requires a finite sum over all **sentence** derivations.)

PCFGs and WCFGs represent the **same conditional family**.

Smith and Johnson (2005)

Optimizing Contrastive Likelihood

$$F\left(\vec{\theta}\right) = \left[\sum_{i=1}^{n} \log p_{\vec{\theta}}\left(\mathbf{X} = \mathbf{x}_i\right) - \log p_{\vec{\theta}}\left(\mathbf{X} \in \mathcal{N}\left(\mathbf{x}_i\right)\right)\right]$$

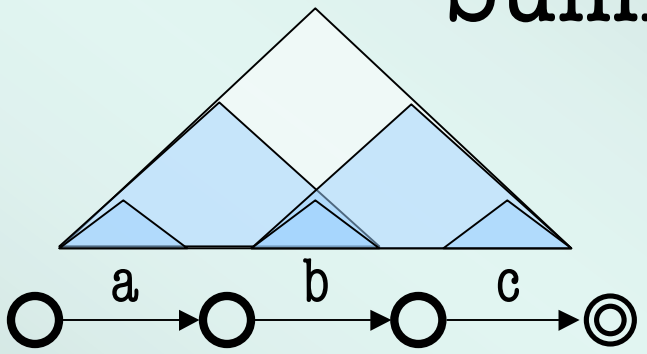
$$\frac{\partial F}{\partial \theta_{\mathbf{r}}} = \left[\sum_{i=1}^{n} \mathbf{E}_{p_{\vec{\theta}}}\left[f_{\mathbf{r}}\left(\mathbf{x}_i, \mathbf{Y}\right)\right] - \mathbf{E}_{p_{\vec{\theta}}}\left[f_{\mathbf{r}}\left(\mathbf{X}, \mathbf{Y}\right) \mid \mathbf{X} \in \mathcal{N}\left(\mathbf{x}_i\right)\right]\right]$$

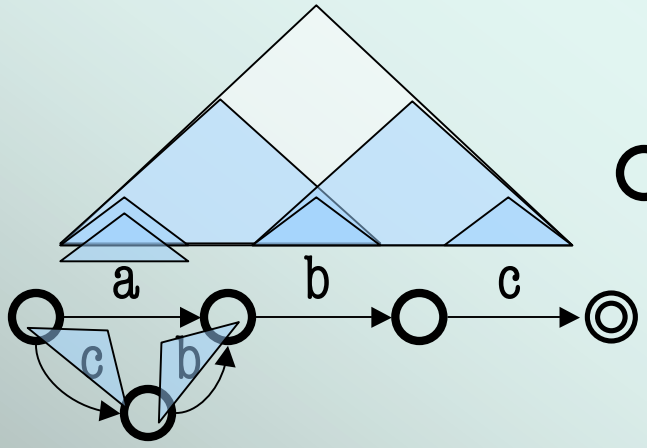
Expected count
Of rule **r** in sentence *i*

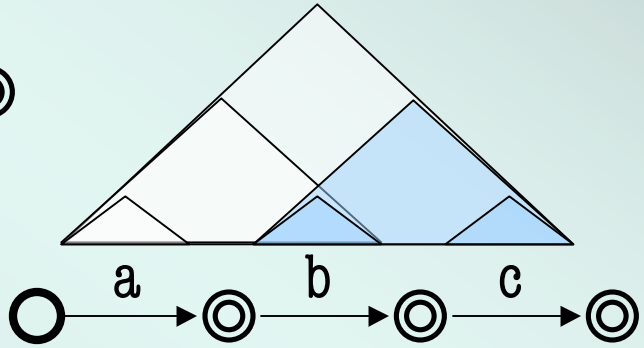
Expected count
Of rule **r** in neighborhood *i*

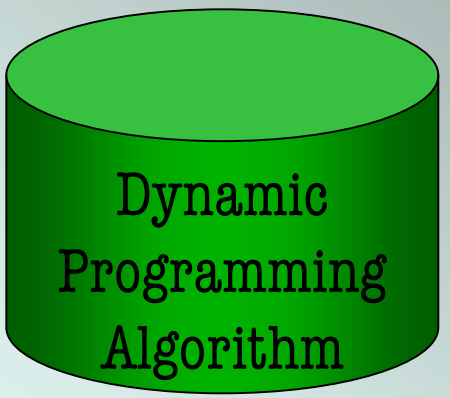
- ☑ What about the simplex constraints?
- ☐ How to make the second term efficient?

Summing over $\mathcal{N}(\mathbf{x})$









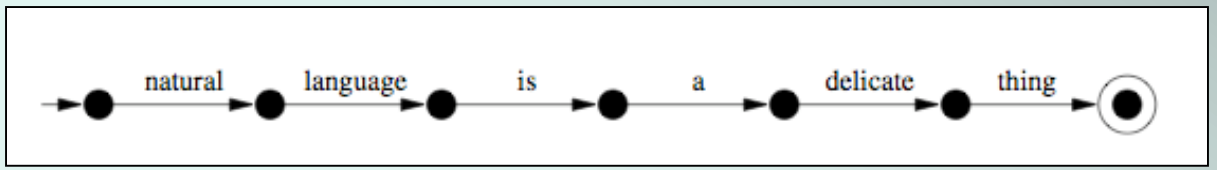
- Dynamic programming saves the day again!
- If the set $\mathcal{N}(\mathbf{x})$ is represented as a lattice, we can apply the usual Inside-Outside algorithm with a slight change.

Original Idea: Word Order

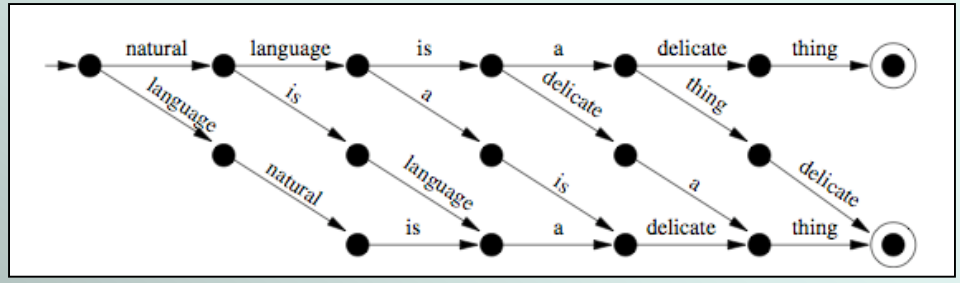
$\mathcal{N}(\mathbf{x})$ = all permutations of $\mathbf{x}$

- Up to $|\mathbf{x}|!$ reorderings and requires lattice with $O(2^{|\mathbf{x}|})$ arcs
- Tradeoff: we want
  - A small lattice
  - A neighborhood that includes as many conceivable negative examples as possible
  - A neighborhood that has few false negative examples

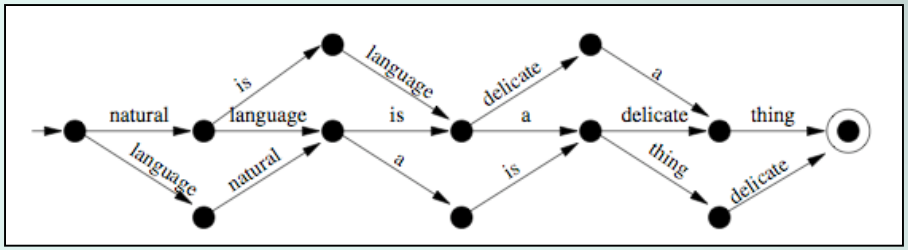
Crude Lattice Neighborhoods



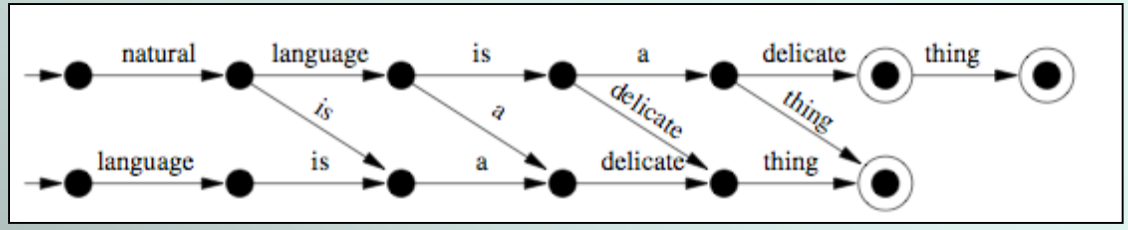
- Mangle the syntax of the sentence by locally reordering and/or deleting some tags.



Transpose 1



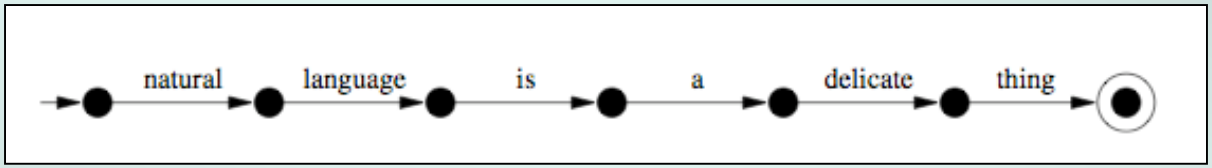
Dynasearch

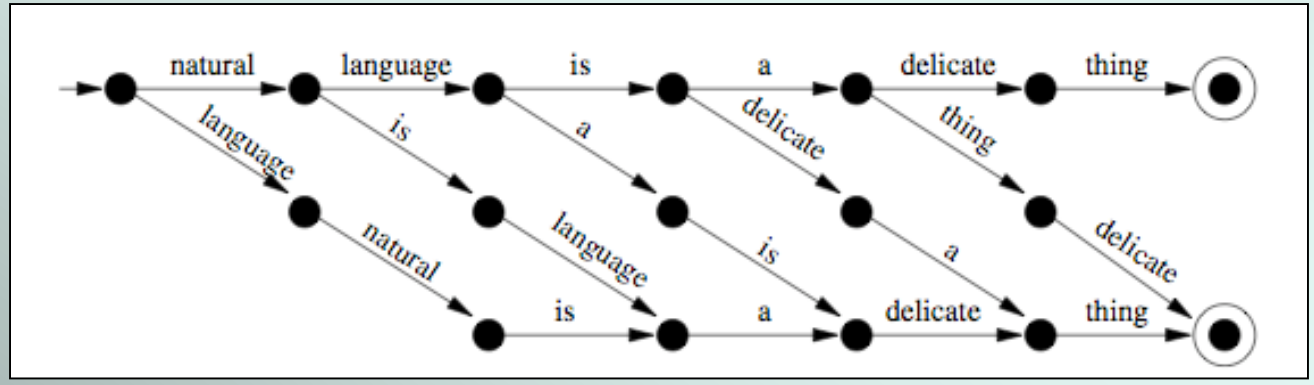


Delete 1

Midpoint Joke

CE Computation





Dynamic Programming Algorithm

Dynamic Programming Algorithm

Experimental Results: CE

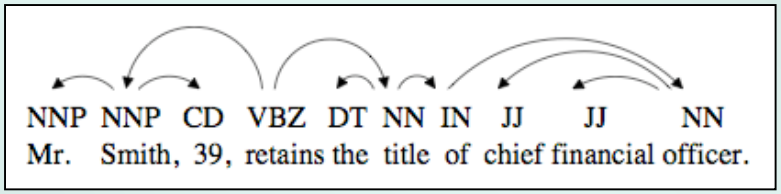
	Accuracy (%)	Undirected Accuracy (%)
Attach-Right	39.5	62.1
MAP/EM (sel. λ, initializer)	41.6	62.2
Del1OrTrans1 (sel. σ^2, init.)	57.6	69.0
Del1 (sel. σ^2, init.)	39.7	53.5
Trans1 (sel. σ^2, init.)	41.2	62.5
Dynasearch (sel. σ^2, init.)	47.6	65.3
Length (sel. σ^2, init.)	45.5	64.9

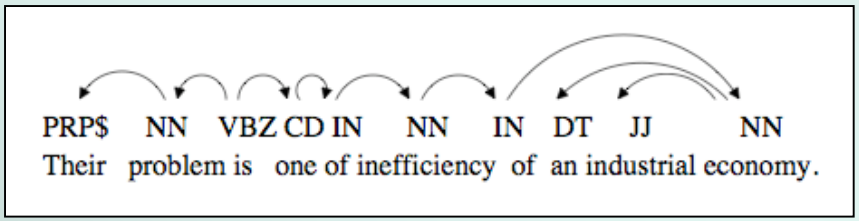
Experimental Results: Dell0rTrans1

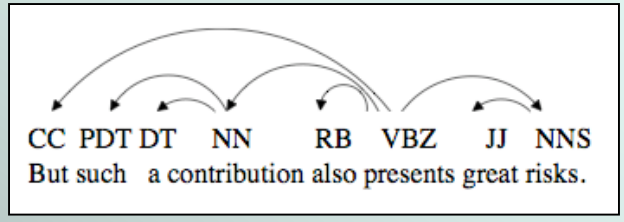
	Zero		K&M		Local	
	Dir. (%)	Undir. (%)	Dir. (%)	Undir. (%)	Dir. (%)	Undir. (%)
Attach-Right	39.5	62.1	39.5	62.1	39.5	62.1
MLE/EM	22.7	58.8	41.7	62.2	22.8	58.9
MAP/EM (sel. λ)	23.8	58.9	41.6	62.2	24.4	59.4
Dell0rTrans1 (unreg.)	35.8	62.2	48.6	64.9	57.6	69.0
Dell0rTrans1 (sel. σ^2)	36.4	61.8	48.4	65.4	57.6	69.0

"Typical" Trees

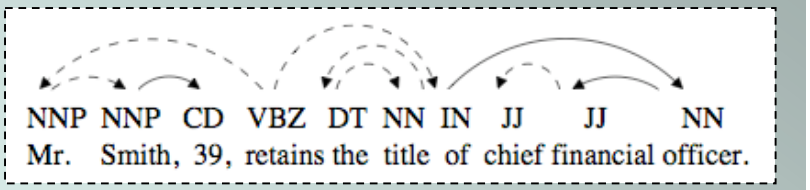
Treebank

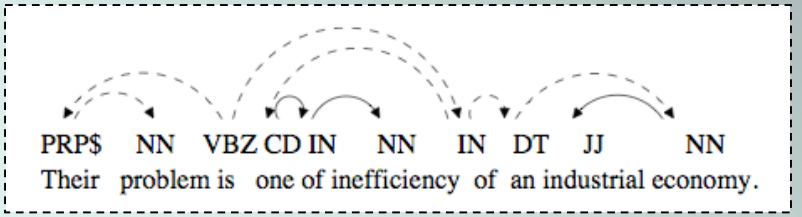


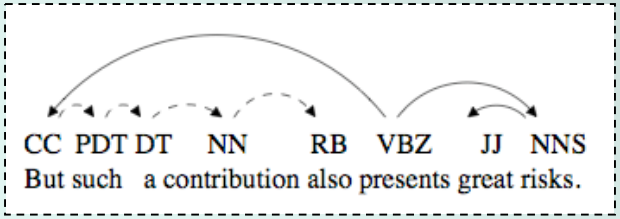




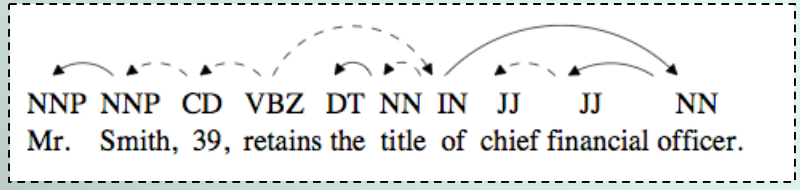
MAP/EM

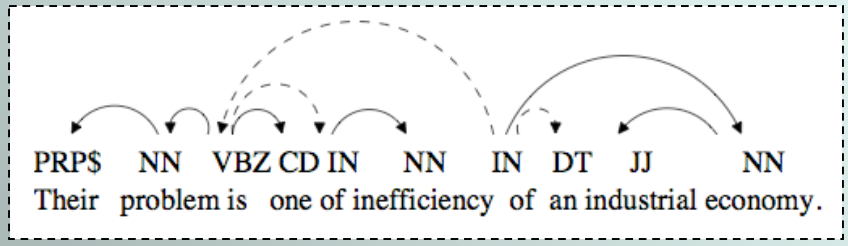


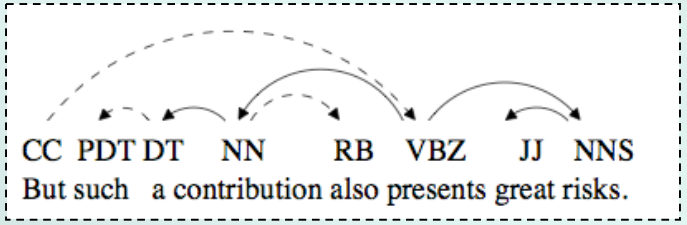




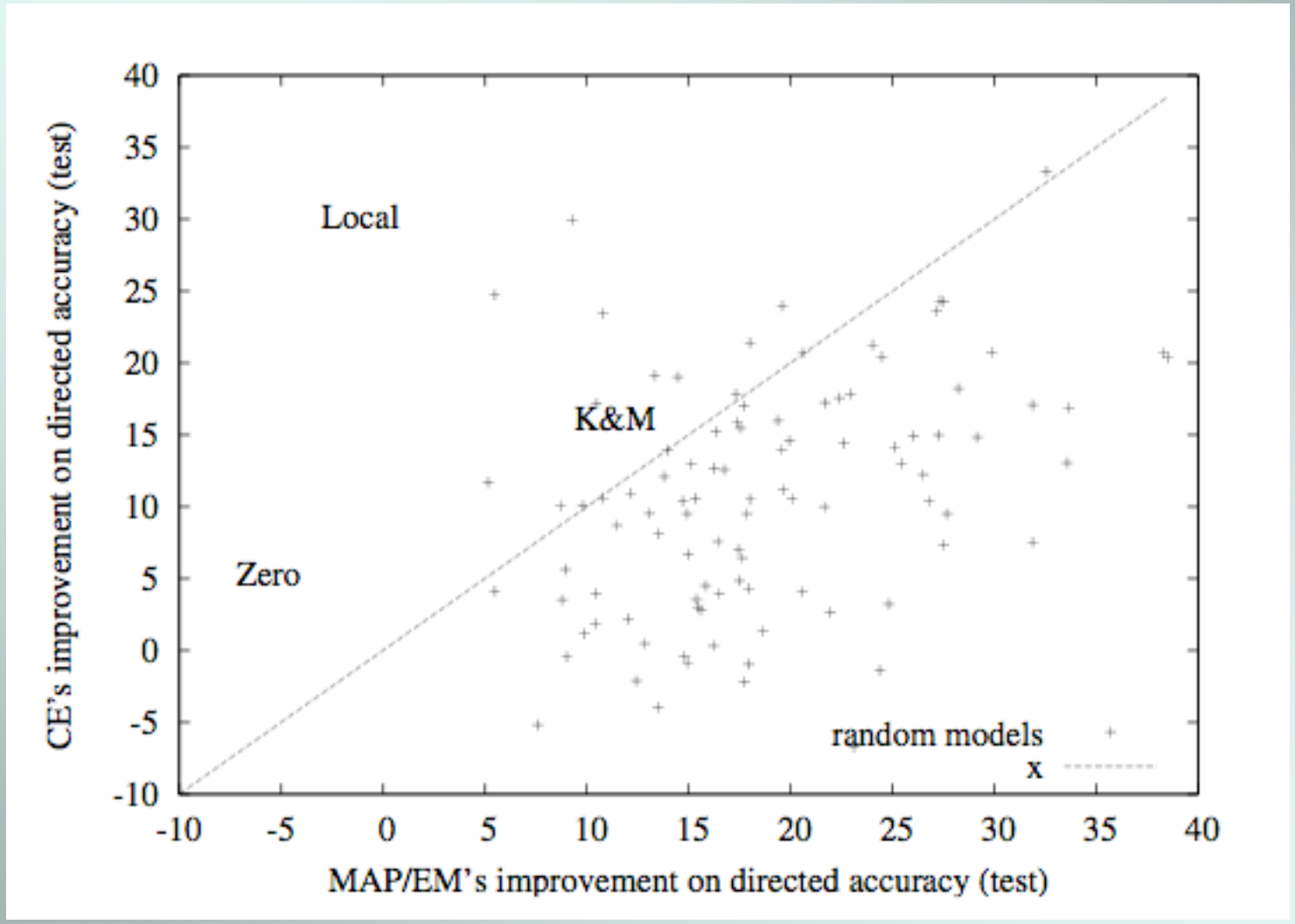
CE







Cause for Concern?



Bonus!

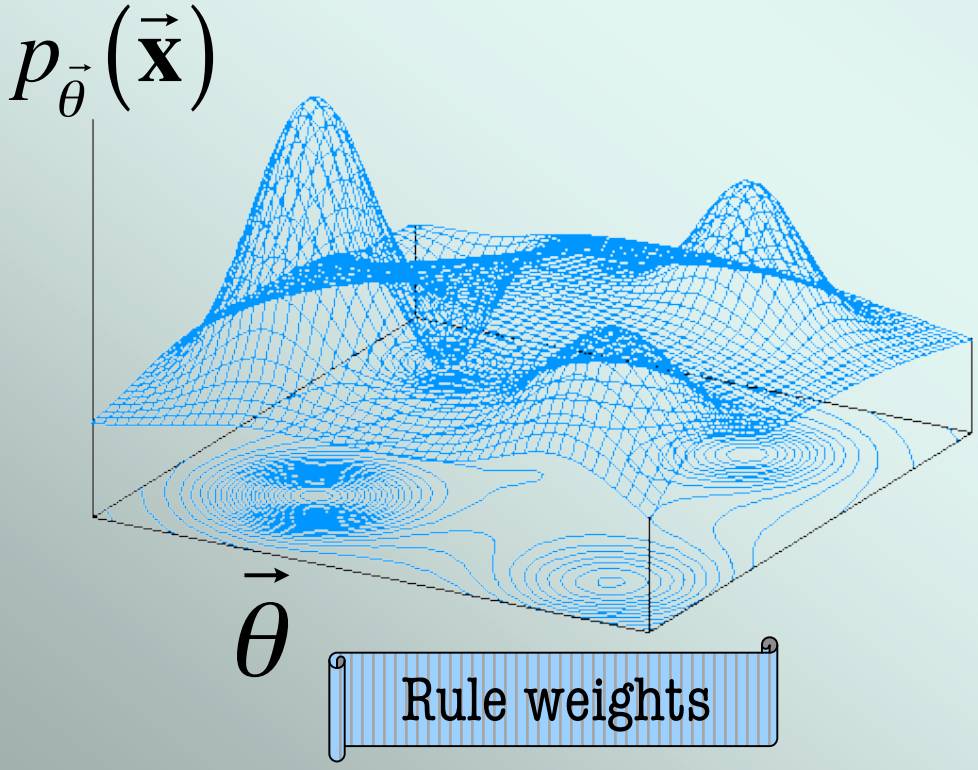
- Log-linear grammars can model more features.
- Smith & Eisner (2005): in HMM estimation from unlabeled data, spelling features can make up for worse dictionaries.
- In thesis: Model U
  - Not representable as a stochastic model (only log-linear)
  - Improvement with spelling features (poor man's lexicalization)

Where are we?

Learning To Parse	Learning = Optimizing a Function	Improving the Function	☑
☑	☑	Improving the Optimizer	☐

Expectation-Maximization

- Hillclimber for the likelihood function.
- Quality of the estimate depends on the starting point.

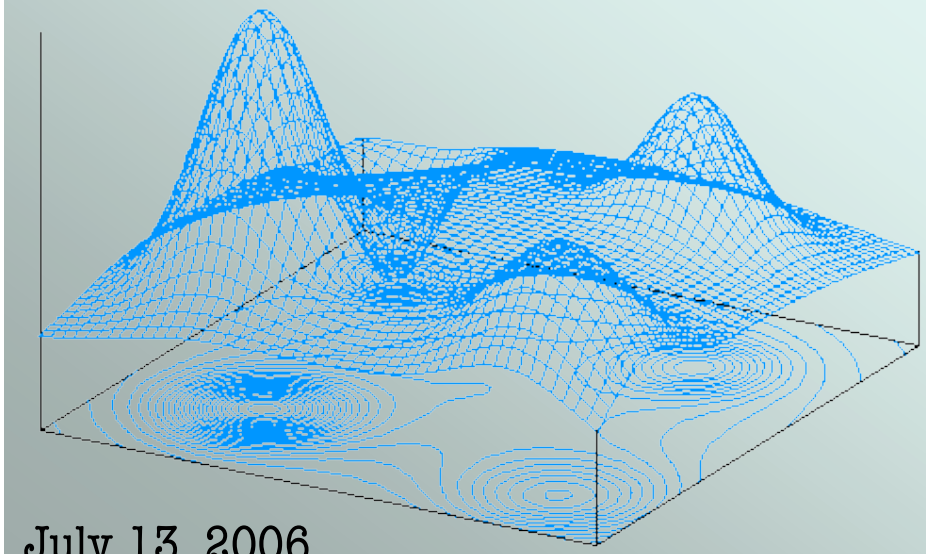


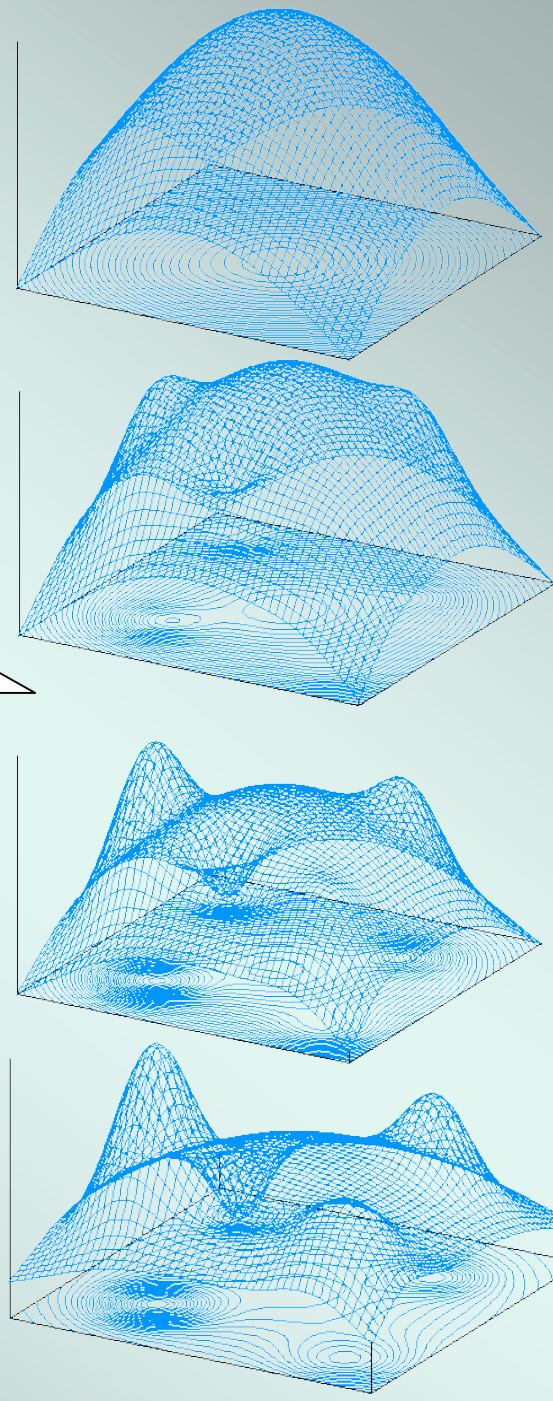
Can we improve the search procedure to avoid getting stuck on local optima?

Deterministic Annealing

Rose et al. (1990)

Ueda and Nakano (1998)





EM as Coordinate Ascent

$$f_{\mathrm{EM}}(\vec{\theta}, q) \stackrel{\mathrm{def}}{=} \tag{5.10}$$

$$\overbrace{\mathbf{E}_{\tilde{p}(\mathbf{X})}\left[\mathbf{H}\left(q(\mathbf{Y} \mid \mathbf{X})\right)\right]}^{\text{improved on E step}} + \overbrace{\mathbf{E}_{\tilde{p}(\mathbf{X}) \cdot q(\mathbf{Y}|\mathbf{X})}\left[\log p_{\vec{\theta}}(\mathbf{Y} \mid \mathbf{X})\right] + \mathbf{E}_{\tilde{p}(\mathbf{X})}\left[\log p_{\vec{\theta}}(\mathbf{X})\right]}^{\text{improved on M step}}$$

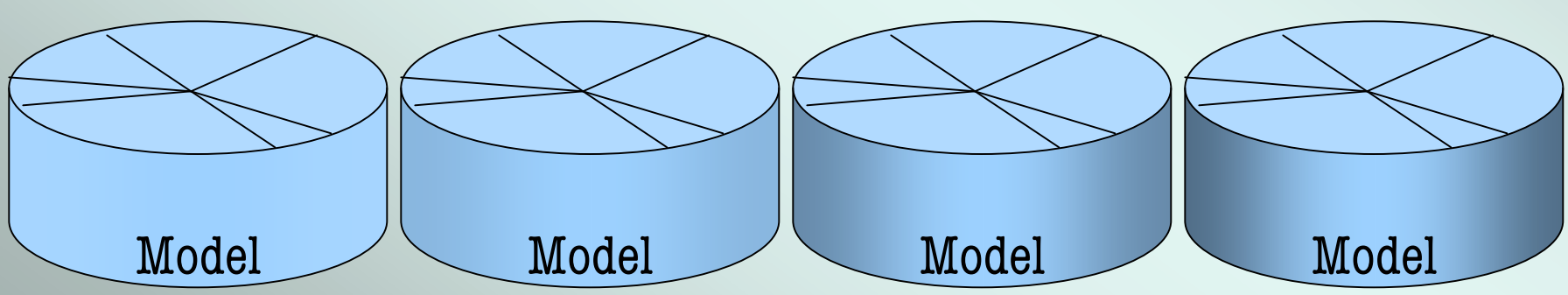
Neal and Hinton (1998)

Deterministic Annealing

$$f_{\mathrm{DA}}(\vec{\theta}, q, \beta) = \frac{1}{\beta}\mathbf{E}_{\bar{p}(\mathbf{X})}\left[\mathbf{H}\left(q(\mathbf{Y} \mid \mathbf{X})\right)\right] + \mathbf{E}_{\bar{p}(\mathbf{X})\cdot q(\mathbf{Y}|\mathbf{X})}\left[\log p_{\vec{\theta}}(\mathbf{Y} \mid \mathbf{X})\right] + \mathbf{E}_{\bar{p}(\mathbf{X})}\left[\log p_{\vec{\theta}}(\mathbf{X})\right] \tag{5.11}$$

High entropy required

No entropy constraint



$\beta \approx 0$

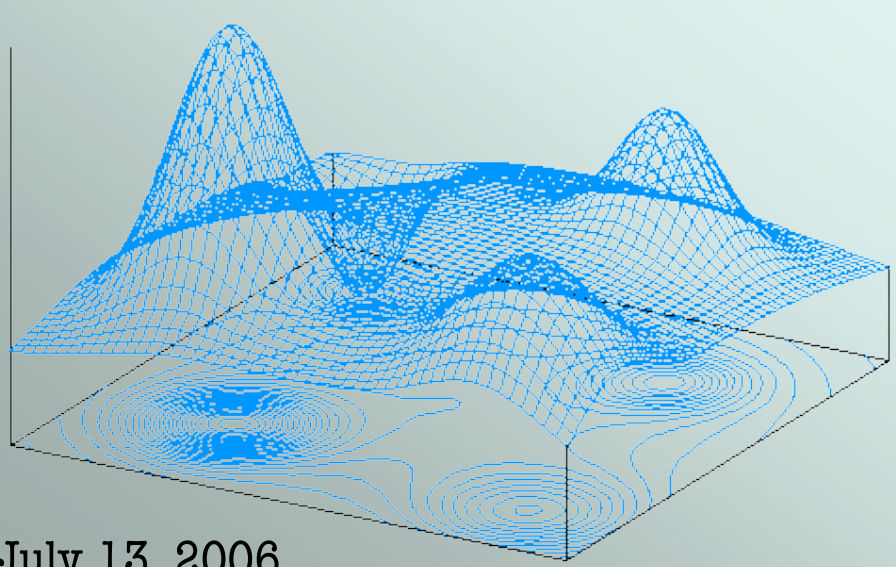
$\beta = 1$

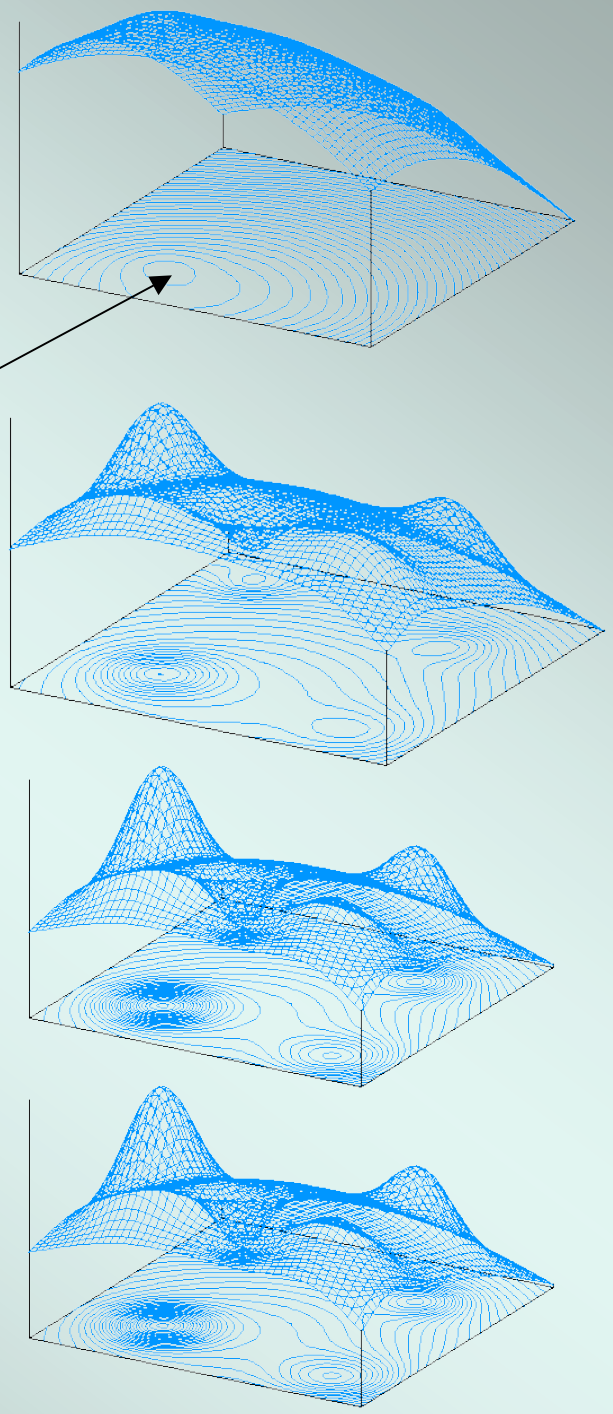
time

Skewed

Deterministic Annealing (Smith and Eisner, 2004)

Clever initializer



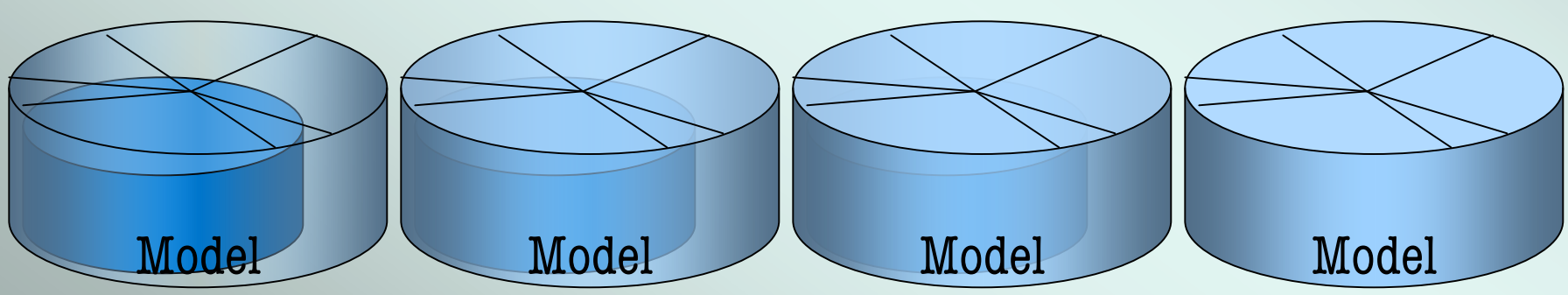


Skewed Deterministic Annealing

$$f_{\mathrm{SDA}}(\vec{\theta}, q, \beta, \acute{p}) = \frac{1}{\beta}\mathbf{E}_{\tilde{p}(\mathbf{X})}\left[\mathbf{H}\left(q(\mathbf{Y} \mid \mathbf{X})\right)\right] + \mathbf{E}_{\tilde{p}(\mathbf{X})\cdot q(\mathbf{Y}|\mathbf{X})}\left[\log p_{\vec{\theta}}(\mathbf{Y} \mid \mathbf{X})\right]$$
$$+\mathbf{E}_{\tilde{p}(\mathbf{X})}\left[\log p_{\vec{\theta}}(\mathbf{X})\right] + \frac{1-\beta}{\beta}\mathbf{E}_{\tilde{p}(\mathbf{X})\cdot q(\mathbf{Y}|\mathbf{X})}\left[\log \acute{p}(\mathbf{Y} \mid \mathbf{X})\right] \quad (5.13)$$

Low divergence from initializer

No divergence constraint



$\beta \approx 0$

time

$\beta = 1$

Optimizers of Likelihood

	Can exploit good initializer	Tries to avoid local optima	Accuracy (s-sel.; %)	Cross-entropy (training)
EM	✔	✖	41.6	26.07
DA	✖	✔	34.8	22.12
Skewed DA	✔	✔	46.7	27.92

Supervised selection applied across initializers, λ (for EM), and schedule (for DA, SDA).

Summary So Far

- EM just barely outperforms Attach-Right
- CE training does better with good initializers
  - ✓ Bonus: log-linear models, so new features can be added
  - ✗ Concern: performance gain not consistent on random models
- DA does its job (better likelihood) but doesn't help accuracy!
- SDA can outperform EM, but not because it avoided a local optimum. (Either luck, or effect of search *trajectory*.)

Objective matters. Search matters.

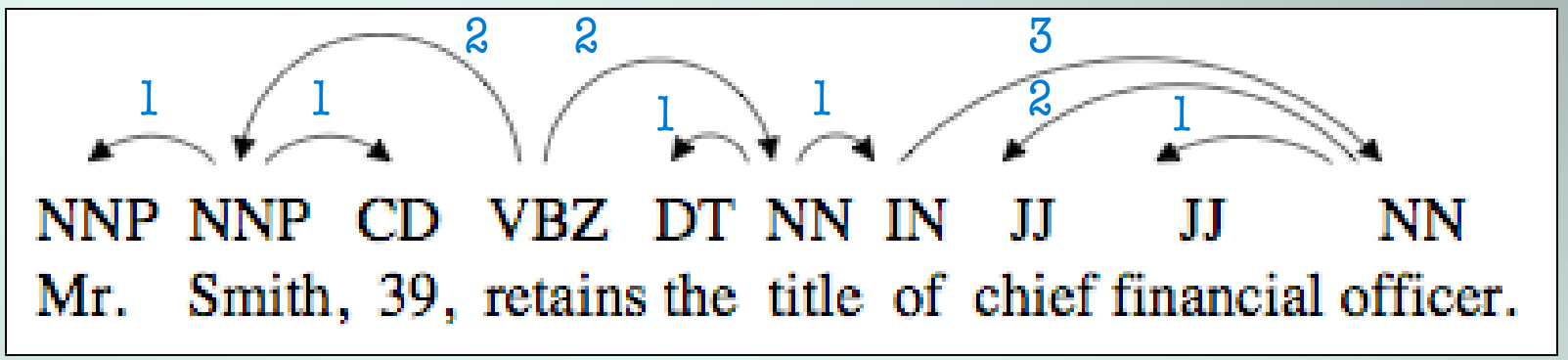
Where are we?

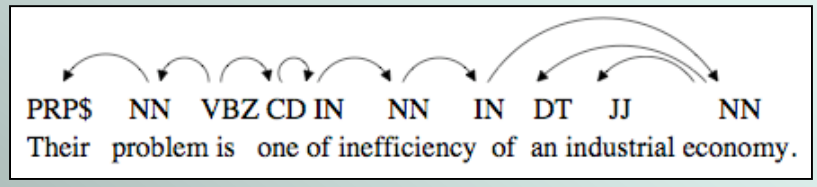
Learning To Parse	Learning = Optimizing a Function	Improving the Function	Improving the Optimizer	Improving the Function and the Optimizer
☑	☑	☑	☑	☐

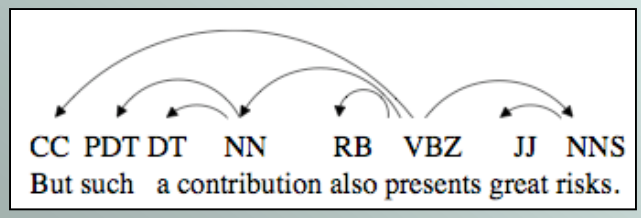
A Different Approach

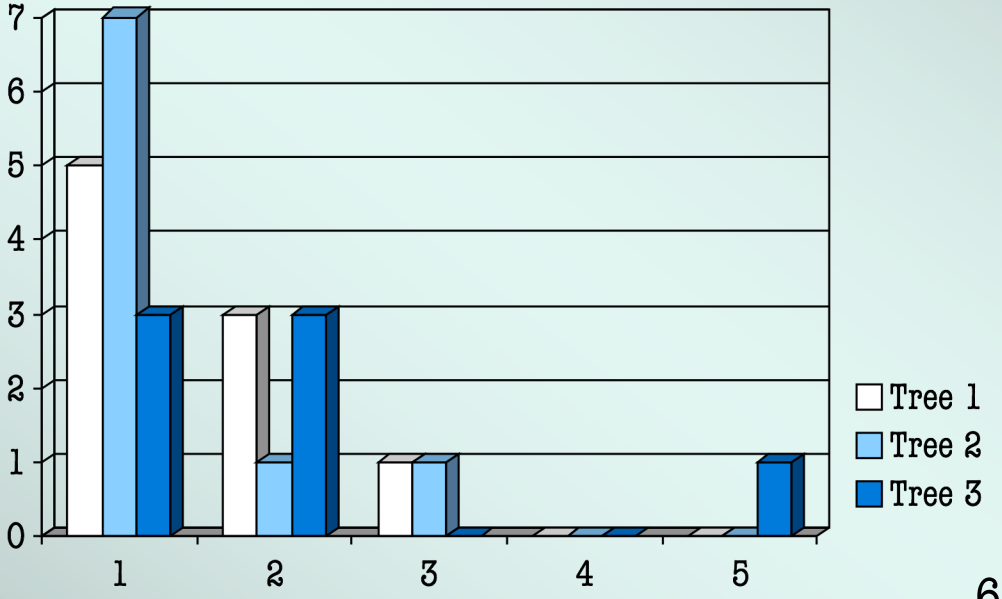
- CE: Domain knowledge defines neighborhood
  - Define what structure is supposed to "explain"
- DA/SDA: "Managed" difficulty improves search
  - Easy function → difficult function
- Structural Annealing:
  - Domain knowledge informs our ideas about search difficulty
  - Easy structures → difficult structures

Short Dependency Preference

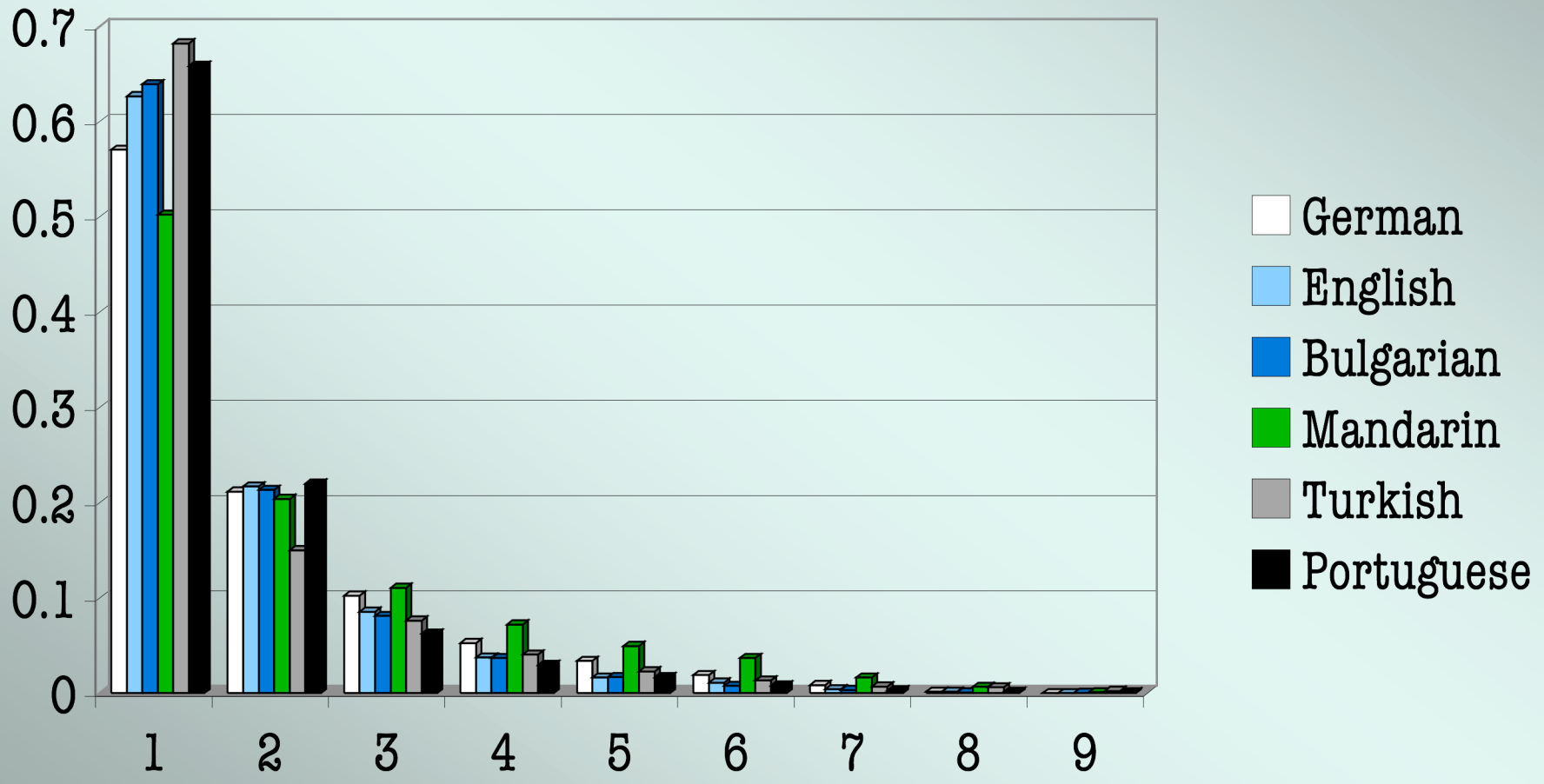








Dependency Length Distribution



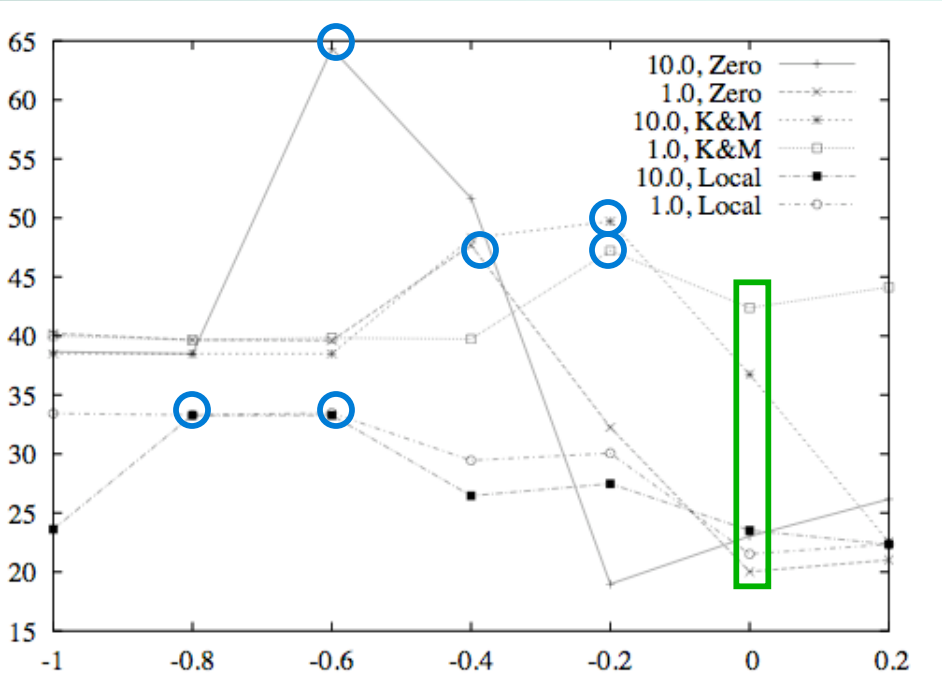
A Locality Feature (Model L)

Global sum-of-lengths feature, factors **locally**

$$p_{\vec{\theta},\delta}(\mathbf{x},\mathbf{y}) \propto p_{\vec{\theta}}(\mathbf{x},\mathbf{y}) \cdot \exp\left(\delta \sum_{i=1}^{|\mathbf{x}|} \sum_{j \in \mathbf{y}(i)} |i-j|\right)$$

Locality bias

accuracy



δ

Structural Annealing

- **Early:** Big penalty for long attachments

($\delta << 0$)

... gradually increase δ ...

- **Later:** No penalty

($\delta = 0$)

(Keep going, using development data to decide when to stop.)

Two Views of SA

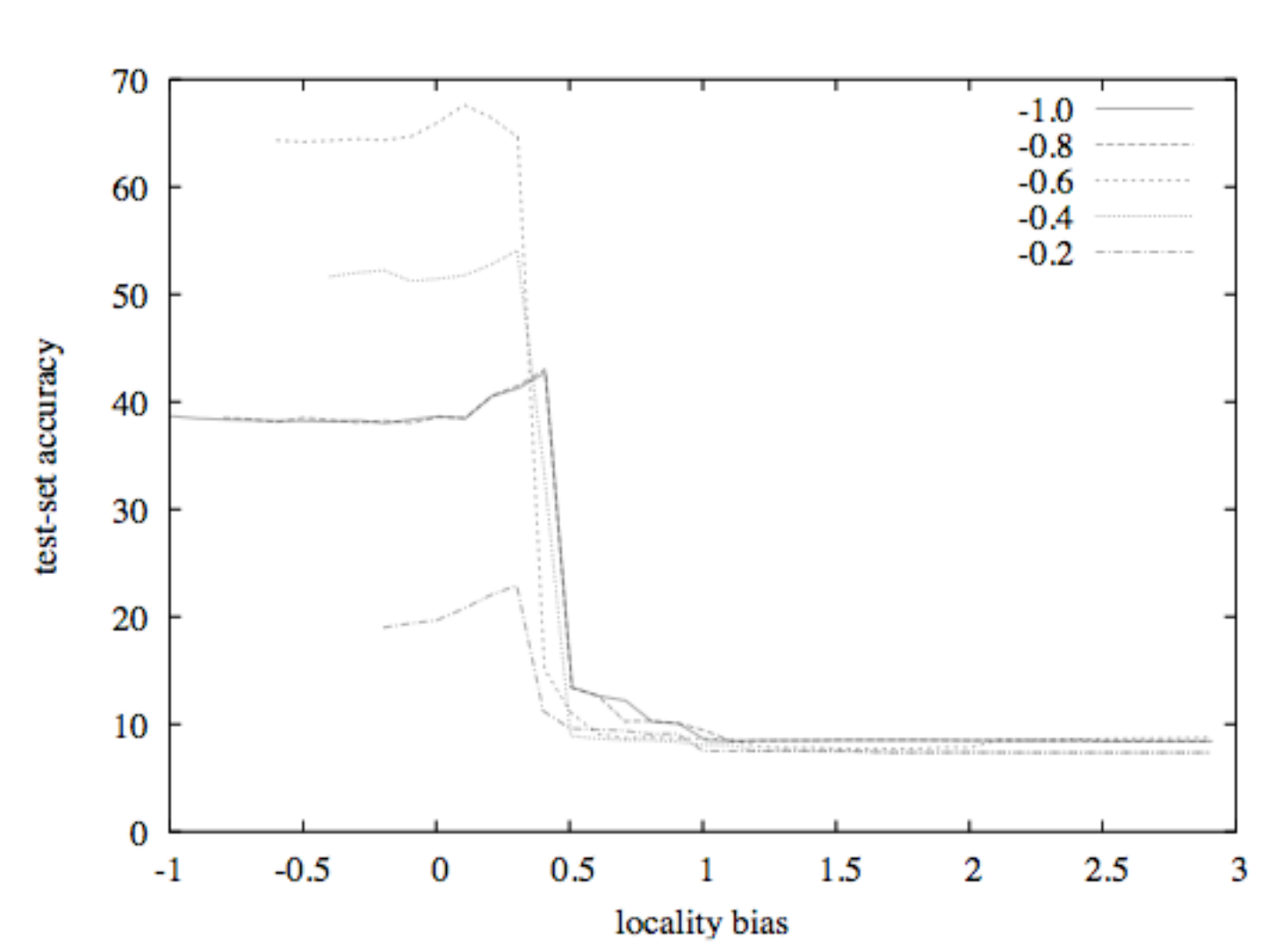
- **Search View:** We start with an easier objective and move to a harder one.

- **Objective Function View:**
  - We added a feature to the model, during training.
  - Its weight is trained in a different way, because we know roughly what it should be.
  - Adding a feature changes the objective.

Experimental Results: SA

	Accuracy (%)	Undirected Accuracy (%)	Hyper-parameters
Attach-Right	39.5	62.1	-
MAP/EM (sel. λ, initializer)	41.6	62.2	$10^{-2/3}$, K&M
CE/Del1OrTrans1 (sel. σ^2, init.)	57.6	69.0	∞, Local
Locality Bias (sel. λ, δ, init.)	61.8	69.4	10, -0.6, Zero
Structural Annealing (sel. λ, δ_0, $\Delta\delta$, δ_f, init.)	66.7	73.1	10, -0.6, 0.1, 0.1, Zero

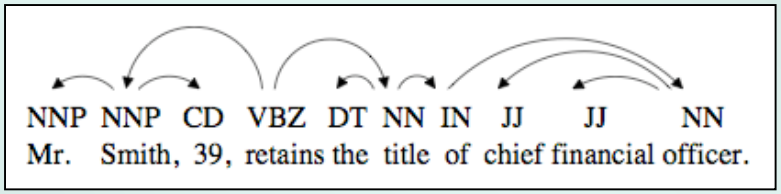
Structural Annealing Performance

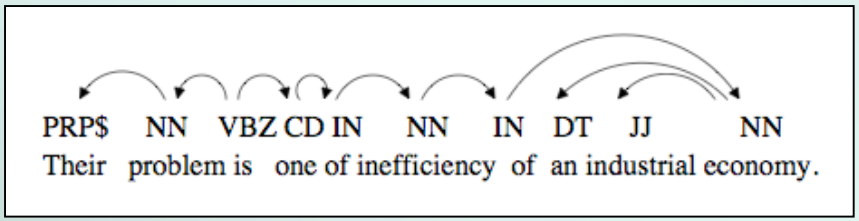


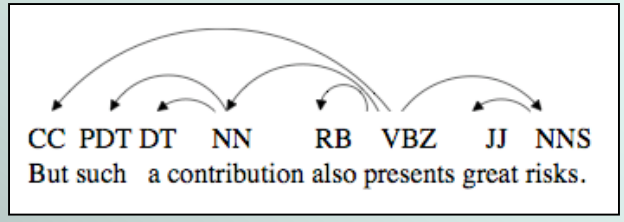
Zero initializer, $\lambda = 10$

"Typical" Trees

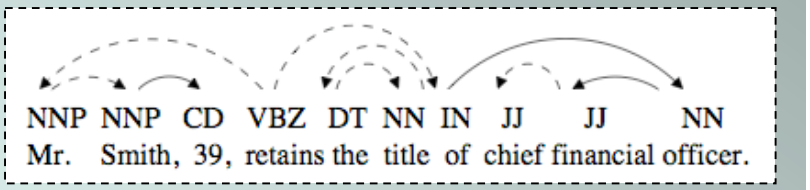
Treebank

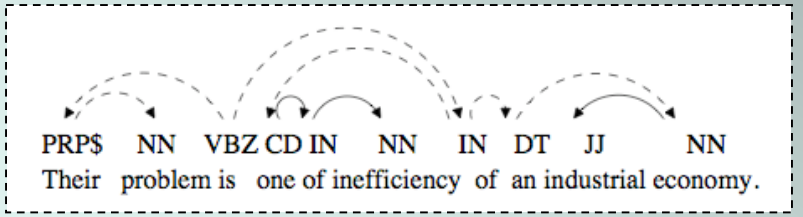


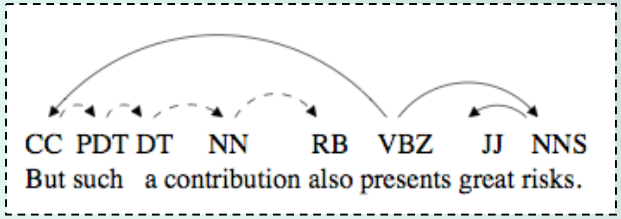




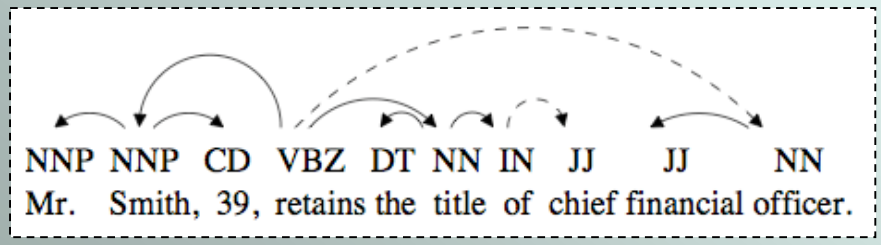
MAP/EM

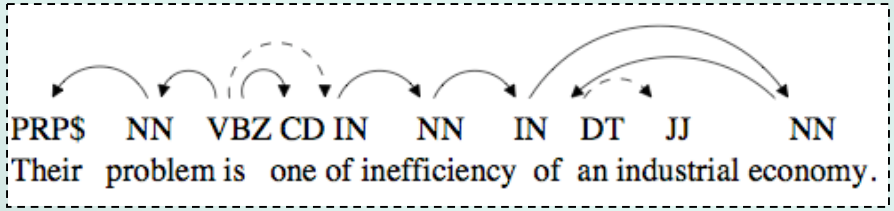


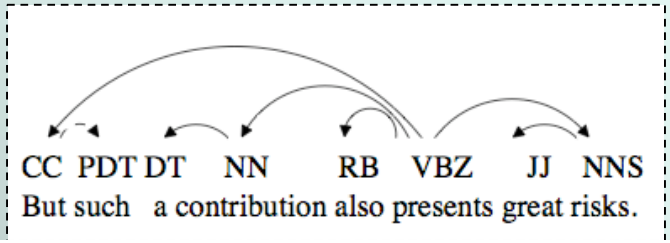




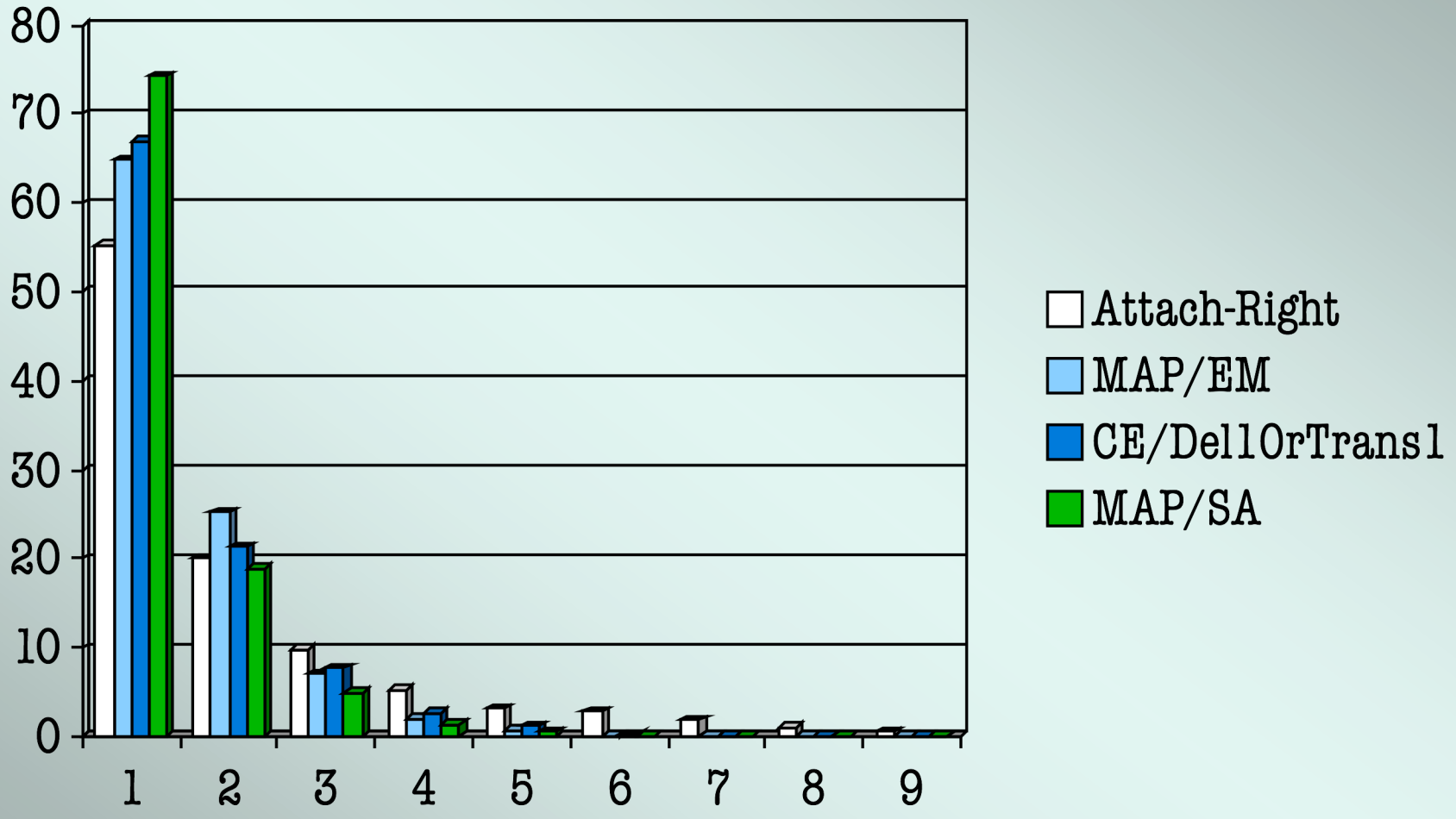
MAP/SA





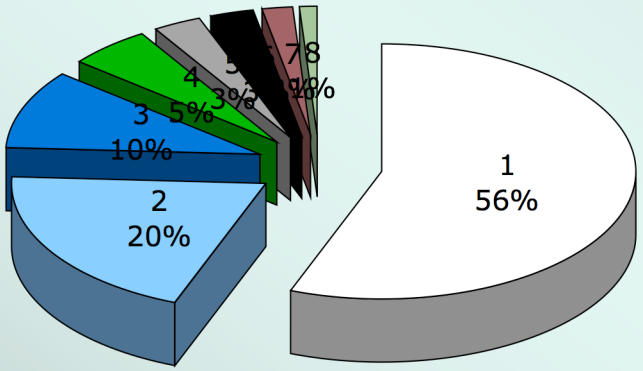


Path Analysis

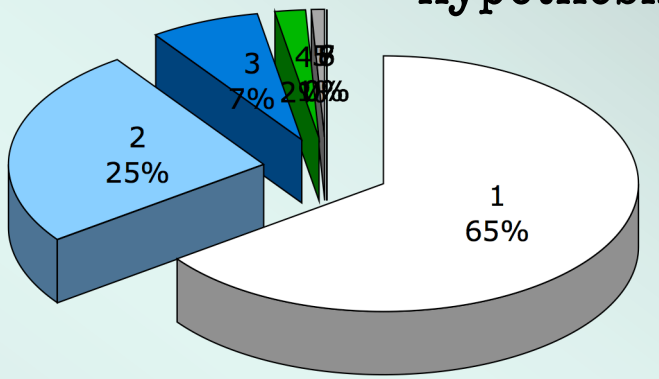


Path Analysis

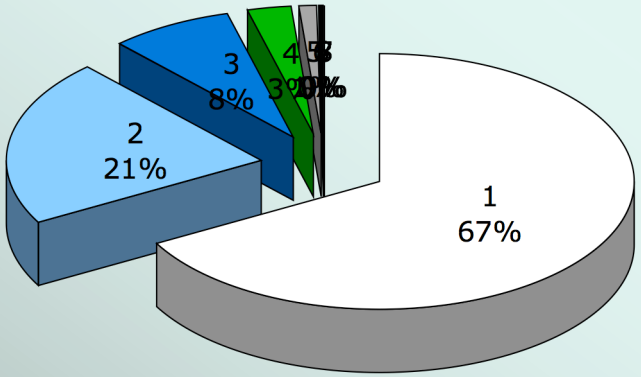
Distribution over distance from a tag to its true parent, in the hypothesized (undirected) tree.



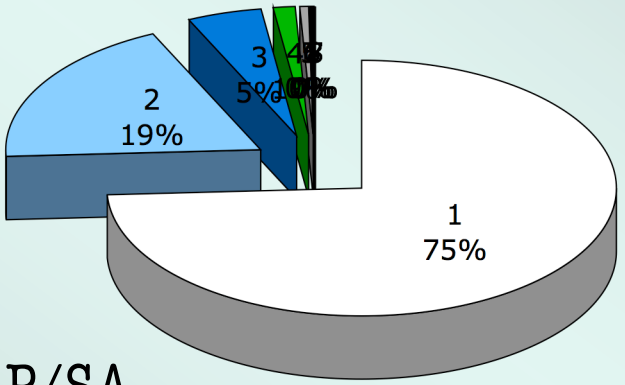
Attach-Right



MAP/EM



CE/Dell0rTransl



MAP/SA

CE and SA

objective

search

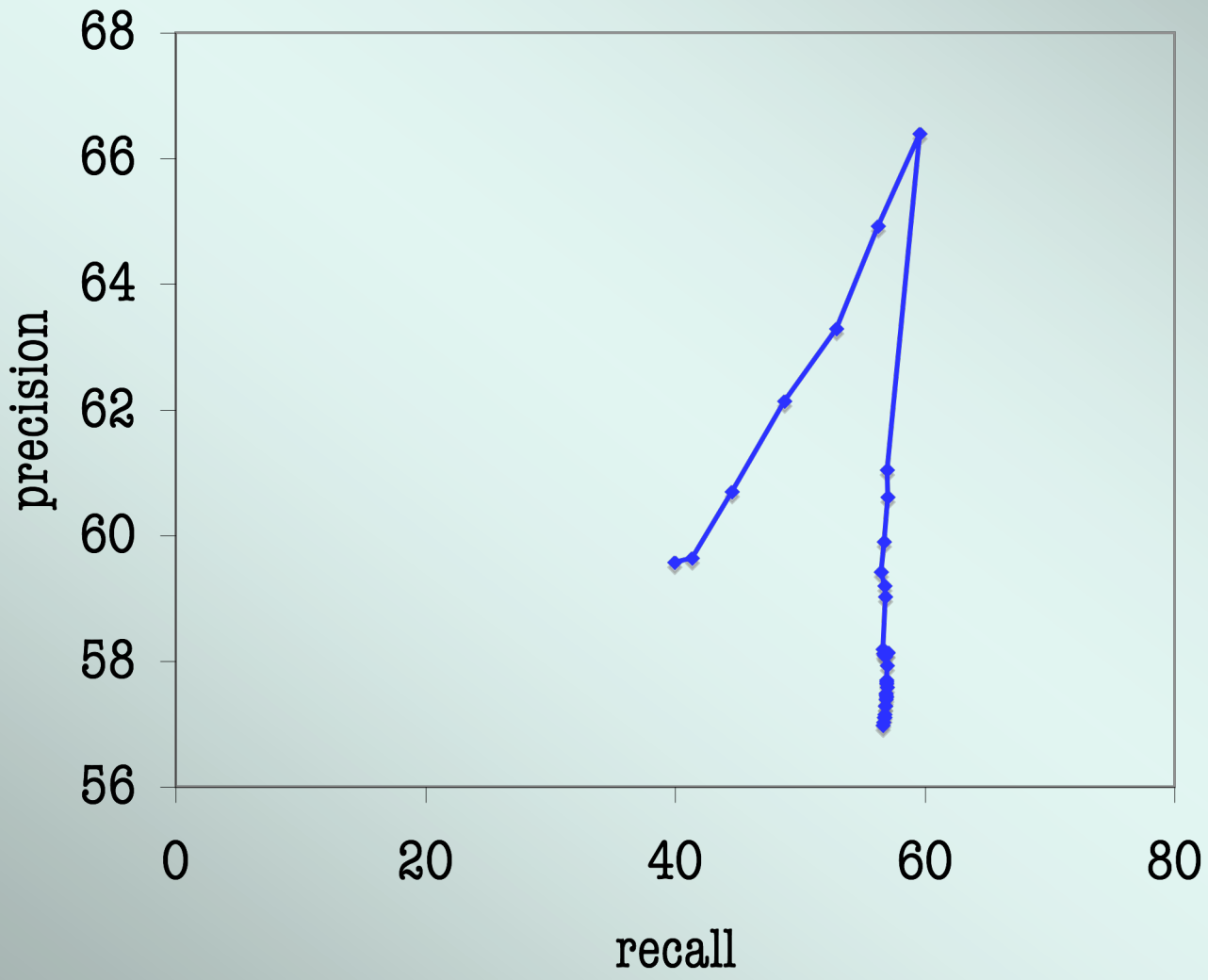
	MAP	CE (Dell0rTrans1)
(No bias	41.6 / 62.2	57.6 / 69.0)
Fixed bias	61.8 / 69.4	63.5 / 71.5
Annealed bias	66.7 / 73.1	65.5 / 72.3

Another Structural Feature

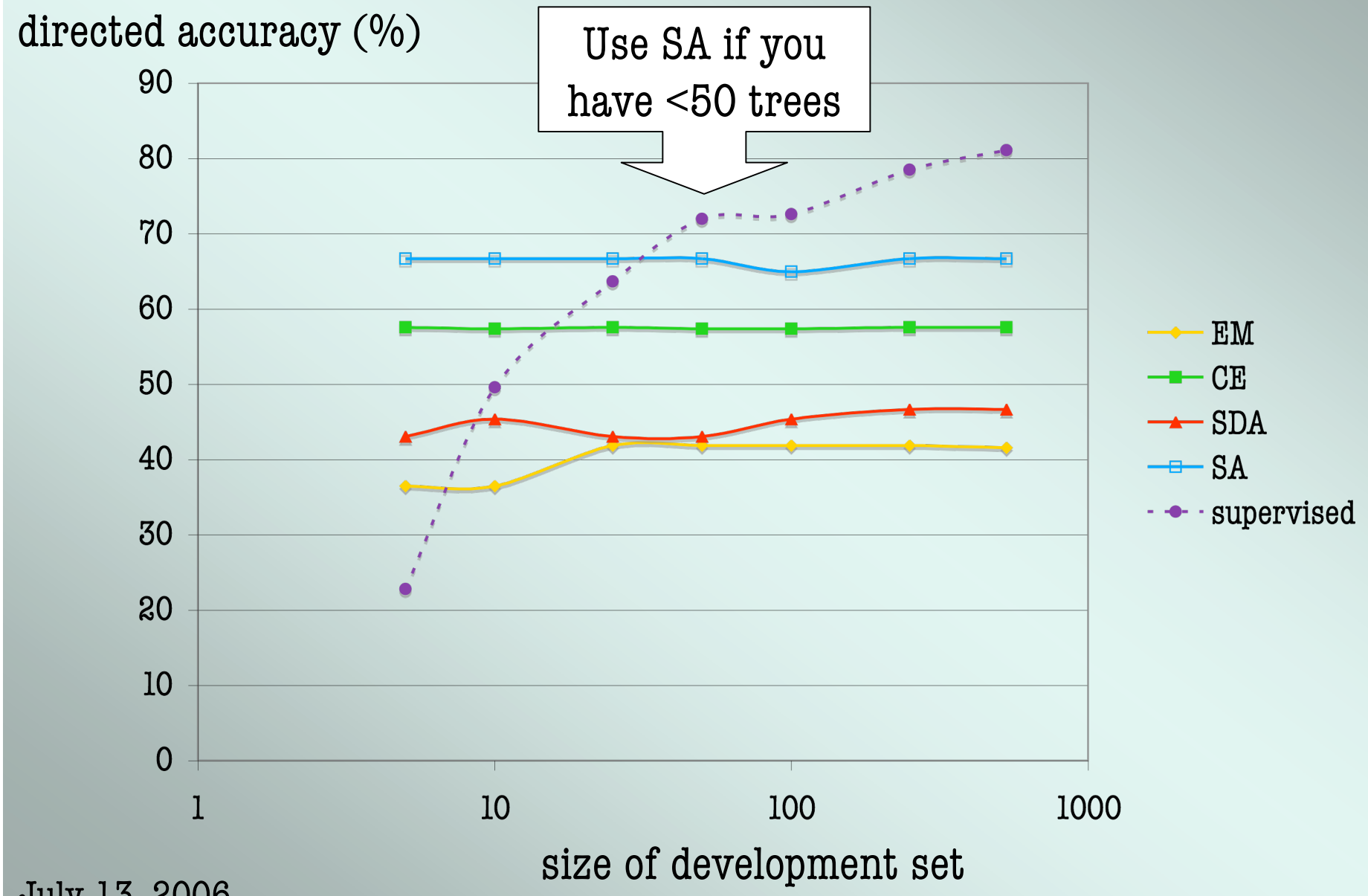
- "Model S" - just like Model A, but allows broken trees (roots modeled by unigram distribution).
- Gradually increase bias toward connectedness.
- Decode with Model A.

	Directed (%)	Undirected (%)
Model A (MAP/EM)	41.6	62.2
Model S (fix β)	55.6	67.0
(anneal β)	58.4	68.8

Decoding under Model S



On Supervision



Where are we?

Learning To Parse	Learning = Optimizing a Function	Improving the Function	Multilingual Experiments
☑	☑		☐

Improving the Optimizer

Improving the Function and the Optimizer
☑

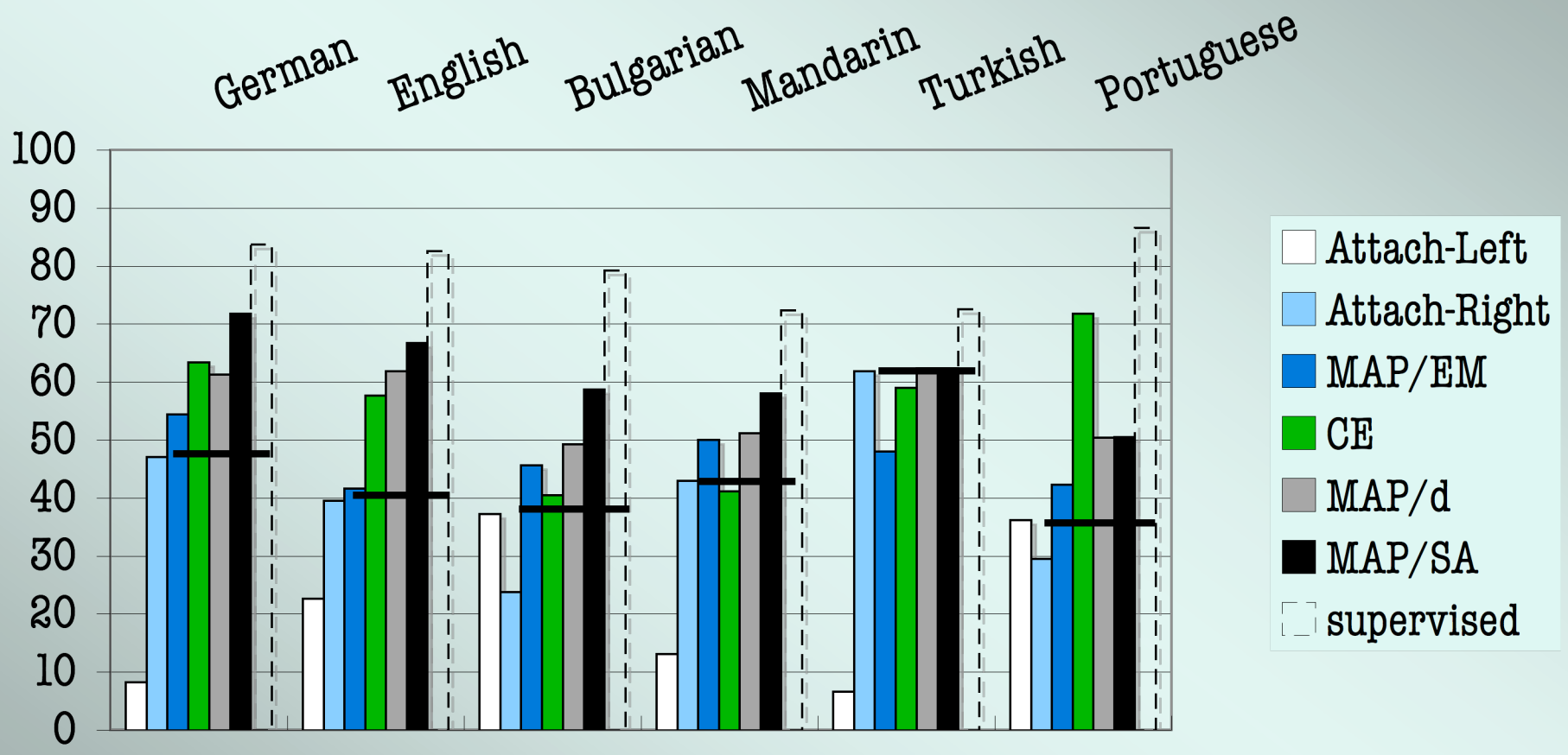
Experimental Setup

- Similar to English:
  - Part-of-speech tags only, sequences of ≤10 tags after stripping punctuation
  - ≈500 development, ≈500 test sentences
- Training:
  - 8K German (Tiger)
  - 5K English (WSJ) & Bulgarian (BulTreeBank)
  - 3K Mandarin (Penn Chinese) & Turkish (METU-Sabanci)
  - 2K Portuguese (Bosque)
- Supervised model selection

Multilingual Experiments

	German	English	Bulgarian	Mandarin	Turkish	Portuguese
Attach-Left	8.2	22.6	37.2	13.1	6.6	36.2
Attach-Right	47.0	39.5	23.8	42.9	61.8	29.5
MAP/EM	54.4	41.6	45.6	50.0	48.0	42.3
CE	63.4	57.6	40.5	41.1	59.0	71.8
MAP/δ	61.3	61.8	49.2	51.1	62.3	50.4
MAP/SA	71.8	66.7	58.7	58.0	62.3	50.5
supervised	*83.7*	*82.5*	*79.2*	*72.3*	*72.5*	*86.5*

Multilingual Experiments



Future Work

- Hyperparameter selection should be part of optimization.
  - More Bayesian (and expensive) approach: optimize hyperparameters, **integrating out** the parameters!
- Better models that can capture **lexical** effects.
  - "Anneal" from Model A into such models?
- Learning & testing on longer sentences.
  - Structural annealing might be even more helpful!
- Better or more task-focused CE neighborhoods?
- Other kinds of structure
  - Cross-lingual structure (word alignments, trees, etc.)
  - Morphology, semantics, discourse, tertiary protein structure ...

Conclusion

- Explored two key dimensions of unsupervised structure learning:
  - What do you optimize? (objective function)
  - How do you optimize it? (search)

  Both are important!
- Five-fold increase in "labeled data threshold."
- State-of-the-art performance on all 6 languages tested.
- Two clean ways to improve unsupervised modeling using domain knowledge: CE, SA

Notes of Appreciation

- ☺ Hertz Foundation (esp. Lowell Wood)
- ☺ Jason Eisner, Dale Schuurmans, Paul Smolensky, David Yarowsky
- ☺ Markus Dreyer, Ben Klemens, David Smith, Roy Tromble
- ☺ Eric Brill, Bill Byrne, Eugene Charniak, Michael Collins, Bob Frank, Joshua Goodman, Keith Hall, Rebecca Hwa, Fred Jelinek, Mark Johnson, Damianos Karakos, Sanjeev Khudanpur, Dan Klein, John Lafferty, Chris Manning, Dan Melamed, Philip Resnik, Dan Roth, Giorgio Satta, Zak Shafran
- ☺ Geetu, John, Silviu, Jia, Sourin, Yonggang, Elliott, Trish, Ahmad, Erin, Hans, Nikesh, Arnab, Eric, Shankar, Gideon, Lambert, Paul, Charles, Yi, Veera, Paola, Chris, Rich, Jun, Peng, Lisa
- ☺ Laura Graham, Eiwe Lingfors, Sue Porterfield, Steve Rifkin, Linda Rorke
- ☺ Kay Dixon, Gene Granger, Lorie Smith, Maria Smith, Wayne Smith
- ☺ Karen Thickman

Key Contributions

- Novel generalization of partial-data MLE to incorporate implicit negative evidence (CE).
  - Bonus: easier training of log-linear models (with arbitrary features)
- Novel generalization of deterministic annealing to exploit good initializers (SDA).
- Novel parameter search technique allowing the use of domain knowledge to start simple and gradually push the model toward difficult structures (SA).
- Significant **accuracy improvements** on weighted grammar induction in six diverse languages.

Other Contributions Not in Thesis

- **WCFG = SCFG** (as conditional distributions)

  (Smith & Johnson, in review)

- **Vine grammar:** regular dependency grammars

  (Eisner & Smith, 2005)

- **Multilingual NLP:**

  Korean/English parsing (Smith & Smith, 2004)

  State-of-the-art morphological disambiguation for Korean, Arabic, and Czech (Smith, Smith, & Tromble, 2005)

  Fast, precise vine parsing for 13 languages (Dreyer, Smith, & Smith, 2006)

Contributor to:

- **Dyna** language ror weighted dynamic programming (Eisner, Goldlust, & Smith, 2004, 2005)
- **STRAND** bilingual text mining system (Resnik, 1999; Resnik and Smith, 2003)
- **Egypt** statistical machine translation toolkit (Al-Onaizan et al., 1999, Smith & Jahr, 2000)

Model A, Supervised

- MLE: 82.5% accuracy, 84.8% undirected
- MAP (oracle λ): 82.8%, 85.1%
- MCLE (unreg.): 83.9%, 86.6%

- MLE (train on Sections 2-21): 70.4% (Section 23)
  - With distance model: 75.6% (Eisner & Smith, 2005)

McDonald et al. (2006): 91.5%

Motivation

- Goal of NLP: build software that does **useful** things with language.
  - Transcribe spoken language.
  - Digitize printed language.
  - Find & present information from text & speech databases.
  - Translate between languages.

- Does this have anything to do with **human** intelligence?

Maybe.

Success will have everything to do with understanding language.

7-fold cross-validation