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

July 13, 2006
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

Build models **empirically** from data; language learning and processing are inference.

**Symbolic** formalisms for elegance, efficiency, and intelligibility.

Machine Learning / Statistics

Linguistics / Cognitive Science

July 13, 2006
An Example: Parsing

Sentence

Dynamic Programming Algorithm

Model

Discrete Search

Parse Tree

Mr. Smith, 39, retains the title of chief financial officer.

Their problem is one of inefficiency of an industrial economy.

But such a contribution also presents great risks.
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?

Mr. Smith, 39, retains the title of chief financial officer.

Their problem is one of inefficiency of an industrial economy.

But such a contribution also presents great risks.
Deeper Problem

- How far can we get with unsupervised estimation?
Outline of the Talk

Learning To Parse
Chapters 1, 2

Learning = Optimizing a Function
Chapter 3

Improving the Function

Improving the Optimizer

Multilingual Experiments

Chapter 7
- 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)

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*
Mr. Smith, 39, retains the title of chief financial officer.
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 $D$. 

July 13, 2006
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.

```
VBZ retains the of chief financial officer
```

Klein & Manning, 2004
Stochastic and Log-linear CFGs

\[ p_{\theta}(x, y) = \prod e^{\theta_r} \]

\[ = \prod e^{f_r(x, y) \theta_r} \]

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

\[ \bar{p}_{\theta}(x, y) = \exp \left( \vec{f}(x, y) \cdot \vec{\theta} \right) \]

Context-Free Grammar (production rules)

Rule weights

Model

Set of all sentences and their trees

Sentence, tree

July 13, 2006
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.

$O(n^5)$ naïve; $O(n^3)$
(Eisner & Satta, 1999)
Evaluation

Treebank tree (gold standard)

Mr. Smith, 39, retains the title of chief financial officer.

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

Treebank tree (gold standard)

Mr. Smith, 39, retains the title of chief financial officer.

Hypothesis tree

Mr. Smith, 39, retains the title of chief financial officer.

Undirected Accuracy = $\frac{5}{(5 + 4)} = 55.5\%$
Fixed Grammar, Learned Weights

All dependency trees on all tag sequences can be derived.

How do we learn the weights?
Maximum Likelihood Estimation

\[
\max_{\hat{\theta}} p_{\hat{\theta}}(\text{observed data})
\]

Supervised training:
“observed data” are sentences with trees

\[
\max_{\hat{\theta}} \left[ \prod_{i=1}^{n} p_{\hat{\theta}}(x_i, y_i) \right]
\]

Independence among examples

For PCFGs, closed form solution

Unsupervised training:
“observed data” are sentences

\[
\max_{\hat{\theta}} \left[ \prod_{i=1}^{n} \sum_{y} p_{\hat{\theta}}(x_i, y) \right]
\]

Requires numerical optimization

Marginalize over trees
Expectation-Maximization

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

\[ p_{\theta}(\vec{x}) \]
EM for Stochastic Grammars

• E step
  
  Compute expected rule counts for each sentence:

  \[ c_r \leftarrow \mathbb{E}_{p_{\hat{\theta}^{(i)}}} \left[ f_r(x_j, Y) \right] \]

• M step

  Renormalize counts into multinomial distributions.

  \[ \theta_r^{(i+1)} = \log(c_r) - Z \]
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

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>Undirected Accuracy (%)</th>
<th>Iterations</th>
<th>Cross-Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach-Left</td>
<td>22.6</td>
<td>62.1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Attach-Right</td>
<td>39.5</td>
<td>62.1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Zero</td>
<td>22.7</td>
<td>58.8</td>
<td>49</td>
<td>26.07</td>
</tr>
<tr>
<td>MLE/EM K&amp;M</td>
<td>41.7</td>
<td>62.1</td>
<td>62</td>
<td>25.16</td>
</tr>
<tr>
<td>Local</td>
<td>22.8</td>
<td>58.9</td>
<td>49</td>
<td>26.07</td>
</tr>
</tbody>
</table>
Dirichlet Priors for PCFG Multinomials

- Simplest conceivable smoothing: add-\(\lambda\)
- Slight change to M step:

\[
\theta_r^{(i+1)} = \log(c_r + \lambda) - Z
\]

As if we saw each event an additional \(\lambda\) times.

This is Maximum a Posteriori estimation, or “MLE with a prior.”

How to pick \(\lambda\)?
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 $\lambda$ values to train with.
## Experimental Results: MAP/EM

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Undirected Accuracy (%)</th>
<th>Iterations</th>
<th>Cross-Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach-Right</td>
<td>39.5</td>
<td>62.1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Zero</td>
<td>22.7</td>
<td>58.8</td>
<td>49</td>
<td>26.07</td>
</tr>
<tr>
<td>MLE/EM K&amp;M</td>
<td>41.7</td>
<td>62.1</td>
<td>62</td>
<td>25.16</td>
</tr>
<tr>
<td>Local</td>
<td>22.8</td>
<td>58.9</td>
<td>49</td>
<td>26.07</td>
</tr>
<tr>
<td>MAP/EM (sel. $\lambda$, initializer)</td>
<td>41.6</td>
<td>62.2</td>
<td>49</td>
<td>25.54</td>
</tr>
</tbody>
</table>
“Typical” Trees

Treebank

Mr. Smith, 39, retains the title of chief financial officer.

Their problem is one of inefficiency of an industrial economy.

But such a contribution also presents great risks.

learned model

July 13, 2006
Good and Bad News About Likelihood

![Graph showing directed accuracy vs. log-likelihood with various model types labeled: K&M, K&M Zero, Local, Zonal.](image)
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 = Optimizing a Function

Learning To Parse

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

$\sum^*$

observed sentences
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)

\[ \Sigma^* \]

observed sentences

implicitly negative sentences
Maximum Likelihood Estimation vs. Contrastive Estimation

**MLE/MAP:**
- observed data are **Sentences**
- neighborhood is $\Sigma^*$

\[
\max_{\theta} \left[ \prod_{i=1}^{n} \sum_y p_{\theta}(x_i, y) \right]
\]

**CE:**
- observed data are **sentences**
- neighborhood is ...?

Require numerical optimization

\[
\max_{\theta} \left[ \prod_{i=1}^{n} \frac{\sum_{y} p_{\theta}(x_i, y)}{\sum_{x} \sum_{y} p_{\theta}(x, y)} \right]
\]

\[
= \max_{\theta} \left[ \prod_{i=1}^{n} p_{\theta}(X = x_i \mid X \in \mathcal{N}(x_i)) \right]
\]

July 13, 2006
Partition Neighborhood = Conditional EM

\[ \Sigma^* \]

- Observed sentences
- Implicitly negative sentences

July 13, 2006
Riezler’s (1999) Approximation

\[ \Sigma^* \]

observed sentences
Analogy to Conditional Estimation
(Supervised)
CE for Syntax

Σ*

observed sentences

Same content, syntactically ill-formed
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

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

\[
\frac{\partial F}{\partial \theta_r} = \left[ \sum_{i=1}^{n} E_{p_{\vec{\theta}}} [f_r(x_i, Y)] - E_{p_{\vec{\theta}}} [f_r(X, Y) \mid X \in \mathcal{N}(x_i)] \right]
\]

- 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 \( p(\text{tree, sentence}) \).

• So do some WCFGs - if you can normalize.

\( \bar{Z}_{\theta}(\mathcal{W}) \)

(Requires a finite sum over all derivation scores.)

PCFGs and WCFGs represent the same family.

• PCFGs represent \( p(\text{tree} \mid \text{sentence}) \).

• So do some WCFGs - if you can normalize.

\( \bar{Z}_{\bar{\theta}}(\mathbf{x}) \)

(Requires a finite sum over all sentence derivations.)

PCFGs and WCFGs represent the same conditional family.


Smith and Johnson (2005)
Optimizing Contrastive Likelihood

\[ F(\tilde{\theta}) = \left[ \sum_{i=1}^{n} \log p_{\tilde{\theta}}(X = x_i) - \log p_{\tilde{\theta}}(X \in \mathcal{N}(x_i)) \right] \]

\[ \frac{\partial F}{\partial \theta_r} = \left[ \sum_{i=1}^{n} \mathbb{E}_{p_{\tilde{\theta}}} [f_r(x_i, Y)] - \mathbb{E}_{p_{\tilde{\theta}}} [f_r(X, Y) \mid X \in \mathcal{N}(x_i)] \right] \]

- Expected count
  - Of rule \( r \) in sentence \( i \)
  - Of rule \( r \) in neighborhood \( i \)

☑️ What about the simplex constraints?
☐ How to make the second term efficient?
Summing over $\mathcal{N}(x)$

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

\[ N(x) = \text{all permutations of } x \]

- Up to \(|x|!\) reorderings and requires lattice with \(O(2^{|x|})\) 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

The Thesis Zone

Prof. Smith?

Knock, knock.

Huh. I didn't know Prof. Smith took such an interest in helping us graduate...

Hmm, what's this?

Wait a minute, let me wipe off this chalk dust...

How to graduate students

How to exploit graduate students

www.phdcomics.com
CE Computation

Dynamic Programming Algorithm

July 13, 2006
# Experimental Results: CE

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Undirected Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach-Right</td>
<td>39.5</td>
<td>62.1</td>
</tr>
<tr>
<td>MAP/EM (sel. $\lambda$, initializer)</td>
<td>41.6</td>
<td>62.2</td>
</tr>
<tr>
<td>Del10rTrans1 (sel. $\sigma^2$, init.)</td>
<td>57.6</td>
<td>69.0</td>
</tr>
<tr>
<td>Del1 (sel. $\sigma^2$, init.)</td>
<td>39.7</td>
<td>53.5</td>
</tr>
<tr>
<td>Trans1 (sel. $\sigma^2$, init.)</td>
<td>41.2</td>
<td>62.5</td>
</tr>
<tr>
<td>Dynasearch (sel. $\sigma^2$, init.)</td>
<td>47.6</td>
<td>65.3</td>
</tr>
<tr>
<td>Length (sel. $\sigma^2$, init.)</td>
<td>45.5</td>
<td>64.9</td>
</tr>
</tbody>
</table>
## Experimental Results: Del1OrTransl

<table>
<thead>
<tr>
<th></th>
<th>Zero</th>
<th>K&amp;M</th>
<th>Local</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dir. (%)</td>
<td>Undir. (%)</td>
<td>Dir. (%)</td>
</tr>
<tr>
<td>Attach-Right</td>
<td>39.5</td>
<td>62.1</td>
<td>39.5</td>
</tr>
<tr>
<td>MLE/EM</td>
<td>22.7</td>
<td>58.8</td>
<td>41.7</td>
</tr>
<tr>
<td>MAP/EM (sel. $\lambda$)</td>
<td>23.8</td>
<td>58.9</td>
<td>41.6</td>
</tr>
<tr>
<td>Del1OrTrans1 (unreg.)</td>
<td>35.8</td>
<td>62.2</td>
<td>48.6</td>
</tr>
<tr>
<td>Del1OrTrans1 (sel. $\sigma^2$)</td>
<td>36.4</td>
<td>61.8</td>
<td>48.4</td>
</tr>
</tbody>
</table>
“Typical” Trees

Mr. Smith, 39, retains the title of chief financial officer.

Their problem is one of inefficiency of an industrial economy.

But such a contribution also presents great risks.
Cause for Concern?

![Graph showing CE's improvement on directed accuracy (test) against MAP/EM's improvement on directed accuracy (test). The graph includes points labeled 'Local', 'K&M', and 'Zero', with a line indicating random models.]

July 13, 2006
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.

$P_{\theta}(\vec{x})$

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

\[ f_{EM}(\tilde{\theta}, q) \overset{\text{def}}{=} \]

\[
\underbrace{\mathbb{E}_{\tilde{p}(X)} [H(q(Y \mid X))] + \mathbb{E}_{\tilde{p}(X) \cdot q(Y \mid X)} [\log p_{\theta}(Y \mid X)]}_{\text{improved on E step}} + \underbrace{\mathbb{E}_{\tilde{p}(X)} [\log p_{\theta}(X)]}_{\text{improved on M step}}
\]

Neal and Hinton (1998)
Deterministic Annealing

\[
f_{DA}(\bar{\theta}, q, \beta) = \frac{1}{\beta} E_{\tilde{p}(x)} \left[ H(q(Y | X)) \right] + E_{\tilde{p}(x) \cdot q(Y | X)} \left[ \log p_{\tilde{\theta}}(Y | X) \right] + E_{\tilde{p}(x)} \left[ \log p_{\tilde{\theta}}(X) \right]
\]

High entropy required

No entropy constraint

\[\beta \approx 0\]

\[\beta = 1\]

July 13, 2006
Skewed Deterministic Annealing (Smith and Eisner, 2004)

Clever initializer
Skewed Deterministic Annealing

\[
f_{\text{SDA}}(\tilde{\theta}, q, \beta, \hat{p}) = \frac{1}{\beta} \mathbb{E}_{\tilde{p}(x)} [H(q(Y | X))] + \mathbb{E}_{\tilde{p}(x) \cdot q(y|x)} [\log p_{\tilde{\theta}}(Y | X)] \\
+ \mathbb{E}_{\tilde{p}(x)} [\log p_{\tilde{\theta}}(X)] + \frac{1 - \beta}{\beta} \mathbb{E}_{\tilde{p}(x) \cdot q(y|x)} [\log \hat{p}(Y | X)](5.13)
\]

Low divergence from initializer  No divergence constraint

\[\beta \approx 0\]  \[\beta = 1\]  \text{time}
### Optimizers of Likelihood

<table>
<thead>
<tr>
<th></th>
<th>Can exploit good initializer</th>
<th>Tries to avoid local optima</th>
<th>Accuracy (s-sel.; %)</th>
<th>Cross-entropy (training)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EM</strong></td>
<td>✔</td>
<td>✗</td>
<td>41.6</td>
<td>26.07</td>
</tr>
<tr>
<td><strong>DA</strong></td>
<td>✗</td>
<td>✔</td>
<td>34.8</td>
<td>22.12</td>
</tr>
<tr>
<td><strong>Skewed DA</strong></td>
<td>✔</td>
<td>✔</td>
<td>46.7</td>
<td>27.92</td>
</tr>
</tbody>
</table>

Supervised selection applied across initializers, $\lambda$ (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?

<table>
<thead>
<tr>
<th>Learning</th>
<th>Learning = Optimizing a Function</th>
<th>Improving the Function</th>
<th>Improving the Optimizer</th>
<th>Improving the Function and the Optimizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

Mr. Smith, 39, retains the title of chief financial officer.

Their problem is one of inefficiency of an industrial economy.

But such a contribution also presents great risks.
Dependency Length Distribution

July 13, 2006
A Locality Feature (Model L)

Global sum-of-lengths feature, factors \textbf{locally}

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

Accuracy

\[ \delta \]
Structural Annealing

• **Early:** Big penalty for long attachments
  \[ \delta \ll 0 \]
  ... gradually increase \( \delta \) ...

• **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

<table>
<thead>
<tr>
<th>Attach-Right</th>
<th>Accuracy (%)</th>
<th>Undirected Accuracy (%)</th>
<th>Hyper-parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP/EM (sel. ( \lambda ), initializer)</td>
<td>39.5</td>
<td>62.1</td>
<td>-</td>
</tr>
<tr>
<td>CE/Del10rTrans1 (sel. ( \sigma^2 ), init.)</td>
<td>41.6</td>
<td>62.2</td>
<td>( 10^{-2/3} ), K&amp;cM</td>
</tr>
<tr>
<td>Locality Bias (sel. ( \lambda ), ( \delta ), init.)</td>
<td>57.6</td>
<td>69.0</td>
<td>( \infty ), Local</td>
</tr>
<tr>
<td>Structural Annealing</td>
<td>61.8</td>
<td>69.4</td>
<td>10, -0.6, Zero</td>
</tr>
<tr>
<td>(sel. ( \lambda ), ( \delta_0 ), ( \Delta \delta ), ( \delta_f ), init.)</td>
<td>66.7</td>
<td>73.1</td>
<td>10, -0.6, 0.1, 0.1, Zero</td>
</tr>
</tbody>
</table>
Structural Annealing Performance

Zero initializer, $\lambda = 10$
Treebank

Mr. Smith, 39, retains the title of chief financial officer.

Their problem is one of inefficiency of an industrial economy.

But such a contribution also presents great risks.

“Typical” Trees

Mr. Smith, 39, retains the title of chief financial officer.

Their problem is one of inefficiency of an industrial economy.

But such a contribution also presents great risks.

MAP/EM

MAP/SA

July 13, 2006
Path Analysis

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

Attach-Right

MAP/EM

CE/Del1OrTrans1

MAP/SA

July 13, 2006
## CE and SA

<table>
<thead>
<tr>
<th>objective search</th>
<th>CE (Del1OrTrans1)</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>(No bias)</td>
<td>41.6 / 62.2</td>
<td>57.6 / 69.0</td>
</tr>
<tr>
<td>Fixed bias</td>
<td>61.8 / 69.4</td>
<td>63.5 / 71.5</td>
</tr>
<tr>
<td>Annealed bias</td>
<td>66.7 / 73.1</td>
<td>65.5 / 72.3</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th></th>
<th>Directed (%)</th>
<th>Undirected (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model A (MAP/EM)</td>
<td>41.6</td>
<td>62.2</td>
</tr>
<tr>
<td>Model S (fix $\beta$)</td>
<td>55.6</td>
<td>67.0</td>
</tr>
<tr>
<td>(anneal $\beta$)</td>
<td>58.4</td>
<td>68.8</td>
</tr>
</tbody>
</table>
Decoding under Model S
On Supervision

Use SA if you have <50 trees
Where are we?

- Learning To Parse
- Learning = Optimizing a Function
- Improving the Function
- Improving the Optimizer
- Improving the Function and the Optimizer
- Multilingual Experiments
Experimental Setup

• Similar to English:
  – Part-of-speech tags only, sequences of $\leq 10$ tags after stripping punctuation
  – $\approx$500 development, $\approx$500 test sentences

• Training:
  – 8K German (Tiger)
  – 5K English (WSJ) & Bulgarian (BulgarianTreeBank)
  – 3K Mandarin (Penn Chinese) & Turkish (METU-Sabanci)
  – 2K Portuguese (Bosque)

• Supervised model selection
## Multilingual Experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>German</th>
<th>English</th>
<th>Bulgarian</th>
<th>Mandarin</th>
<th>Turkish</th>
<th>Portuguese</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach-Left</td>
<td>8.2</td>
<td>22.6</td>
<td>37.2</td>
<td>13.1</td>
<td>6.6</td>
<td>36.2</td>
</tr>
<tr>
<td>Attach-Right</td>
<td>47.0</td>
<td>39.5</td>
<td>23.8</td>
<td>42.9</td>
<td>61.8</td>
<td>29.5</td>
</tr>
<tr>
<td>MAP/EM</td>
<td>54.4</td>
<td>41.6</td>
<td>45.6</td>
<td>50.0</td>
<td>48.0</td>
<td>42.3</td>
</tr>
<tr>
<td>MAP/δ</td>
<td>61.3</td>
<td>61.8</td>
<td>49.2</td>
<td>51.1</td>
<td>62.3</td>
<td>50.4</td>
</tr>
<tr>
<td>MAP/SA</td>
<td>71.8</td>
<td>66.7</td>
<td>58.7</td>
<td>58.0</td>
<td>62.3</td>
<td>50.5</td>
</tr>
</tbody>
</table>

| Supervised   | 83.7   | 82.5    | 79.2      | 72.3     | 72.5    | 86.5       |

July 13, 2006
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


😊 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 for weighted dynamic programming (Eisner, Goldlust, & Smith, 2004, 2005)
- **STRAND** bilingual text mining system (Resnik, 1999; Resnik and Smith, 2003)
Model A, Supervised

- MLE: 82.5% accuracy, 84.8% undirected
- MAP (oracle $\lambda$): 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

map/em (directed accuracy)

directed accuracy

- CE/Dell10rTrans1
- MAP/SA

July 13, 2006