Approximate Inference in Natural Language Processing

Noah Smith
Language Technologies Institute and Machine Learning Department
School of Computer Science
Carnegie Mellon University

Joint work with Kevin Gimpel, André F. T. Martins, and Eric P. Xing
Requisite Joke

• The organizers confused me with someone else.
Requisite Joke

• The organizers confused me with someone else.

• *David* Smith (U. Mass.) has been doing great work on extending loopy belief propagation to solve hard NLP problems.

  • I won’t talk about any of that.

  • Instead I’ll talk about work from my group.

• If you think the organizers should have invited him instead, blame it on faulty (human) NLP.
Talk Outline

1. Natural language processing and parsing in a perfect world

2. Generalizing dynamic programming with semirings and beyond

3. Parsing as integer linear programming and relaxations
Natural Language Parsing

• Linguistic analysis of sentences

• “Bare bones” dependency trees: one of many formalisms
  • Idealized representation of the syntax of NL.

• Uses: information extraction, question answering, summarization, building lexical databases, **translation** (we’ll come back to this)

$ The professor chuckled with unabashed glee$
Context-Free Dependency Parsing

- $S \rightarrow$ $\$ \text{CHUCKLED}$
- $\text{CHUCKLED} \rightarrow \text{CHUCKLED GLEE}$
- $\text{CHUCKLED} \rightarrow \text{PROFESSOR CHUCKLED}$
- $\text{GLEE} \rightarrow \text{WITH GLEE}$
- $\text{GLEE} \rightarrow \text{UNABASHED GLEE}$
- $\text{PROFESSOR} \rightarrow \text{THE PROFESSOR}$

- $\text{THE} \rightarrow \text{the}$
- $\text{PROFESSOR} \rightarrow \text{professor}$
- $\text{CHUCKLED} \rightarrow \text{chuckled}$
- $\text{WITH} \rightarrow \text{with}$
- $\text{UNABASHED} \rightarrow \text{unabashed}$
- $\text{GLEE} \rightarrow \text{glee}$

binary rules capture bilexical dependencies

unary rules model geometric valence distributions
Parsing with CFGs

• Probabilistic CKY equations:

\[
Y_{N,i-1,i} = \theta_{N \rightarrow x_i} \\
Y_{N,i,k} = \max_{N',N'' \in \mathcal{N}, j \in \{i+1, \ldots, k-1\}} Y_{N',i,j} \times Y_{N'',j,k} \times \theta_{N \rightarrow N'N''}
\]

\[
p_\theta \left( X = x, Y = \arg \max_y p_\theta(y \mid x) \right) = Y_{S,0,|x|}
\]

• n-length sentence parses in $O(n^3)$ time, $O(n^2)$ space, using bottom-up dynamic programming.

• Specialized version for dependency CFGs, cubic with strong assumptions (Eisner, 1996; Eisner and Satta, 1999).
Parsing and Inference

- “Semiring” CKY equations (Goodman, 1999):

\[
Y_{N,i-1,i} = \theta_{N \rightarrow x_i}
\]

\[
Y_{N,i,k} = \bigoplus_{N', N'' \in \mathcal{N}, j \in \{i+1, \ldots, k-1\}} Y_{N',i,j} \otimes Y_{N'',j,k} \otimes \theta_{N \rightarrow N' N''}
\]

- Useful for mode of \( p_\theta(Y \mid X = x) \) (“max”), marginal \( p_\theta(X = x) \) (“sum”), entropy \( H(Y \mid X = x) \), loss-augmented max, ...

- Dynamic programming still applies.

  - Connection to sum-product, max-product belief propagation (Sato, 2007).
Parsing and Inference

• “Semiring” CKY equations with **features**:

\[
Y_{N,i-1,i} = \psi_{N,i,x} \\
Y_{N,i,k} = \bigoplus_{N',N'' \in \mathcal{N}, j \in \{i+1, \ldots, k-1\}} Y_{N',i,j} \otimes Y_{N'',j,k} \otimes \psi_{N,N',N'',i,j,k,x} \\
\psi_{\ldots} = \exp w^\top f(\ldots)
\]

• Log-linear, “max ent,” exponential, global linear, undirected ...

• Miyao and Tsuji’i, 2002, Clark and Curran, 2004, Finkel et al., 2008, *inter* (many) *alia*
Context-Free Dependency Parsing (Eisner, 1996)

$ The professor chuckled with unabashed glee

goal

$ The professor chuckled with unabashed glee
Context-Free Dependency Parsing

Attach:  

$ The professor chuckled with unabashed glee$
The professor chuckled with unabashed glee
The professor chuckled with unabashed glee.

Complete:

$\text{The professor chuckled with unabashed glee}$
Combining Items to Create Updates

\[
\begin{array}{c|c|c}
\times & 0.2 \\
0.4 & 0.08 \\
\end{array}
\]
$ The professor chuckled with unabashed glee
$ The professor chuckled with unabashed glee

Attach:
The professor chuckled with unabashed glee.
The professor chuckled with unabashed glee

$\text{Attach:}$

$\text{$ The professor chuckled with unabashed glee$}$
The professor chuckled with unabashed glee.
The professor chuckled with unabashed glee.
Context-Free Dependency Parsing

$ The professor chuckled with unabashed glee
Context-Free Dependency Parsing

$ The professor chuckled with unabashed glee.
Context-Free Dependency Parsing

Inference = finding the semiring sum; in general requires exhaustive consideration of “pieces”: $O(n^3)$

$\$ The professor chuckled with unabashed glee
1. Pay linguistic experts $ millions to **annotate** news articles with trees.

2. **Train** statistical models from training examples (Charniak, 1997; Collins, 1997, ...)

3. **Parse** (using your favorite inference algorithm) test examples.

4. **Measure** accuracy.
Manual Parse (1 hour, 1 grad student brain)
Automatic Parse (10 minutes, 4.5 GB)
- Stanford Parser

58/153 attachment errors
What’s the Trajectory?

- So far, we’ve been able to use dynamic programming for \textit{exact} inference.

- Computational linguists are not satisfied with the underlying models!
  - Richer formalisms, richer features, weaker independence assumptions
  - Other linguistic structures (integrating morphology, semantics; moving beyond syntax)
• All of NLP isn’t parsing.
  
  • But more and more of it is looking like parsing.
  
  • After sequence labeling, parsing is the next natural **structure-prediction** problem to tackle.
Talk Outline

✓ Natural language processing and parsing in a perfect world

2. Generalizing dynamic programming with semirings and beyond

3. Parsing as integer linear programming and relaxations

Primarily work by Kevin Gimpel
Local and Non-Local Features

• Features that consider only one edge:
  
  • Parent = chuckled
  • Parent = chuckled ∧ Child = professor
  • Parent = chuckled ∧ Child = professor ∧ Distance = 1
  • Parent = chuckled ∧ Child = professor ∧ ChildRight = with

$\text{The professor chuckled with unabashed glee}$
Local and Non-Local Features

- Features that consider multiple edges that are tree-local (second order):
  - Parent = chuckled ∧ Child = glee ∧ Grandchild = with

$\text{The professor chuckled with unabashed glee}$
Local and Non-Local Features

- Features that consider multiple edges that are tree-local (second order):
  - Parent = castigated $\land$ Child1 = professor $\land$ Child2 = student

$\text{The professor castigated the student}$
Local and Non-Local Features

- More extremely non-local:
  - $\text{Word0} = \$ \land \text{Word1} = \text{the} \land \text{Path} = [\$ \to \text{chuckled} \leftarrow \text{professor} \leftarrow \text{the}]$

$\$ The professor chuckled with unabashed glee$
Non-local Features and DP

• The more non-local, the higher the polynomial order of the DP algorithm.

• Arbitrarily non-local features render exact DP intractable.

• Solution from MT: **cube pruning** (Chiang, 2007; Huang & Chiang, 2007)
  
  • Keep a(n approximate) k-best list of complete structures for each DP item.

  • More naïve, easier to describe: “cube decoding”
Combining Items

• When combining two items \( \mathbf{a} \) and \( \mathbf{b} \) to make \( \mathbf{c} \), the value of \( \mathbf{c} \) is \( \oplus \)-incremented by \( \mathbf{a} \otimes \mathbf{b} \).

Let the values be \( k \)-length vectors of scored partial structures; consider the cross-product.
### k-Best Combinations

<table>
<thead>
<tr>
<th>×</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.08</td>
<td>0.04</td>
<td>0.02</td>
</tr>
<tr>
<td>0.3</td>
<td>0.06</td>
<td>0.03</td>
<td>0.015</td>
</tr>
<tr>
<td>0.02</td>
<td>0.004</td>
<td>0.002</td>
<td>0.001</td>
</tr>
</tbody>
</table>
k-Best Combinations

<table>
<thead>
<tr>
<th></th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.08</td>
<td>0.04</td>
<td>0.02</td>
</tr>
<tr>
<td>0.3</td>
<td>0.06</td>
<td>0.03</td>
<td>0.015</td>
</tr>
<tr>
<td>0.02</td>
<td>0.004</td>
<td>0.002</td>
<td>0.001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0.08</th>
<th>0.06</th>
<th>0.04</th>
</tr>
</thead>
</table>
k-Best Combinations with Non-Local Features

⊗ in the local and non-local features
k-Best Combinations with Non-Local Features

<table>
<thead>
<tr>
<th></th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.008</td>
<td>0.004</td>
<td>0.001</td>
</tr>
<tr>
<td>0.3</td>
<td>0.018</td>
<td>0.009</td>
<td>0.003</td>
</tr>
<tr>
<td>0.02</td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
</tbody>
</table>
k-Best Combinations with Non-Local Features

<table>
<thead>
<tr>
<th></th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.008</td>
<td>0.004</td>
<td>0.001</td>
</tr>
<tr>
<td>0.3</td>
<td>0.018</td>
<td>0.009</td>
<td>0.003</td>
</tr>
<tr>
<td>0.02</td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

0.018  0.009  0.008
k-Best

- Keeps the basic DP “logic” and algorithms

\[ Y_{N,i-1,i} = \psi_{N,i,x} \]

\[ Y_{N,i,k} = \bigoplus_{N',N'' \in \mathcal{N}, j \in \{i+1,\ldots,k-1\}} Y_{N',i,j} \otimes Y_{N'',j,k} \otimes \psi_{N,N',N'',i,j,k,x} \]

\[ \psi_{\ldots} = \exp \mathbf{w}^\top \mathbf{f}(\ldots) \]

- With non-local features, this is approximate (and we don’t have any formal guarantees); **bigger k implies closer approximation**

- What about all the exponentially many other structures?

- Cube *summing*: maintain a residual
### k-Best Combinations with “Residuals”

<table>
<thead>
<tr>
<th></th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
<th>0.03</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.008</td>
<td>0.004</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>0.018</td>
<td>0.009</td>
<td>0.003</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0001</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### k-Best Combinations with “Residuals”

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
<th>0.03</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td></td>
<td>0.008</td>
<td>0.004</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td></td>
<td>0.018</td>
<td>0.009</td>
<td>0.003</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td></td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0001</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### k-Best Combinations with “Residuals”

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
<th>0.03</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td></td>
<td>0.008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td></td>
<td>0.018</td>
<td>0.009</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0084</td>
</tr>
<tr>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
### k-Best Combinations with “Residuals”

<table>
<thead>
<tr>
<th>b</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
<th>0.03</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.008</td>
<td></td>
<td></td>
<td>0.012</td>
</tr>
<tr>
<td>0.3</td>
<td></td>
<td>0.009</td>
<td></td>
<td>0.009</td>
</tr>
<tr>
<td>0.02</td>
<td></td>
<td></td>
<td>0.0084</td>
<td>0.0006</td>
</tr>
<tr>
<td>0.05</td>
<td>0.01</td>
<td>0.005</td>
<td>0.0025</td>
<td>0.0015</td>
</tr>
</tbody>
</table>
### k-Best Combinations with “Residuals”

<table>
<thead>
<tr>
<th></th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
<th>0.03</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.008</td>
<td></td>
<td></td>
<td>0.012</td>
</tr>
<tr>
<td>0.3</td>
<td>0.018</td>
<td>0.009</td>
<td></td>
<td>0.009</td>
</tr>
<tr>
<td>0.02</td>
<td></td>
<td></td>
<td>0.0084</td>
<td>0.0006</td>
</tr>
<tr>
<td>0.05</td>
<td>0.01</td>
<td>0.005</td>
<td>0.0025</td>
<td>0.0015</td>
</tr>
</tbody>
</table>

⊗ in the local features only
### k-Best Combinations with “Residuals”

<table>
<thead>
<tr>
<th></th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
<th>0.03</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>0.018</td>
<td>0.009</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td></td>
<td></td>
<td>0.0084</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td>0.0203</td>
</tr>
</tbody>
</table>
k-Best Combinations with “Residuals”

\[
\begin{array}{cccc}
0.2 & 0.1 & 0.05 & 0.03 \\
0.4 & 0.008 & \text{...} & \text{...} \\
0.3 & 0.018 & 0.009 & \text{...} \\
0.02 & \text{...} & \text{...} & \text{...} \\
0.05 & \text{...} & \text{...} & \text{...} \\
\end{array}
\]
k-Best Combinations with “Residuals”

\[ a \times b \]
Formally ...

• With only local features, the k-best list and residual are exact. And we have a semiring!

• With non-local features, we lose associativity and distributivity; no longer a semiring.

• For any features, if $k \to \infty$, the k-best list is exact and the residual goes to 0.

• Generalizes the sum-product semiring and the k-best semiring.

• Exact “outside” values (related to first derivatives) are straightforward for the approximate function.

• See Gimpel and Smith (EACL 2009) for more details.
Application to Machine Translation

• Two main approaches to data-driven translation:
  • Source sentence breaks into contiguous **phrases**, each is translated, then they are reordered into a target sentence.
  • Synchronous grammars; two sentences share an isomorphic **syntactic** derivation (up to reordering of siblings).

• Our approach: phrases and grammar rules (and many other things) are features; many are non-local (e.g., language model).
  • DP “backbone” based on lattice dependency parsing. Syntax is local, phrase features are not.
  • Log-linear model with hidden variables. Trained using pseudolikelihood.

• See Gimpel and Smith (EMNLP 2009) for many more details!
MT Performance (German-English BTEC)

The diagram shows the BLEU score as a function of the value of $k$ for decoding. There are four different methods represented:

- **Phrase + Syntactic** (solid blue diamonds)
- **Phrase** (dashed blue squares)
- **Syntactic** (dotted red triangles)
- **Neither** (dashed orange circles)

The BLEU score increases with increasing values of $k$ for all methods, but the rate of increase and the final score vary depending on the method used.
MT Conclusion

• Having rich (non-local) features is more important than exact inference with those features.
Talk Outline

✓ Natural language processing and parsing in a perfect world

✓ Generalizing dynamic programming with semirings and beyond

3. Parsing as integer linear programming and relaxations

Primarily work by André Martins
The “Projectivity” Constraint

- Context-freeness (and the use of CKY-like inference) corresponds to a
  projectivity constraint on the trees.

- Nonprojective structures are not hard to find, even in English:

\[
\text{\$ I saw a talk Saturday that inspired me} \]

\[
\text{\$ I saw a talk Saturday that inspired me} \]

\[
\text{\$ I saw a talk Saturday that inspired me} \]

\[
\text{\$ I saw a talk Saturday that inspired me} \]

\[
\text{\$ I saw a talk Saturday that inspired me} \]

\[
\text{\$ I saw a talk Saturday that inspired me} \]

\[
\text{\$ I saw a talk Saturday that inspired me} \]

\[
\text{\$ I saw a talk Saturday that inspired me} \]
The “Projectivity” Constraint

- Context-freeness (and the use of CKY-like inference) corresponds to a projectivity constraint on the trees.

- Nonprojective structures are not hard to find, even in English:

$ I saw a talk Saturday that inspired me

- With edges conditionally independent (given that $Y$ is a tree):
  - Max-inference: maximum directed spanning tree and Chu-Liu-Edmonds algorithm (McDonald et al., 2005)
  - Sum-inference: Tutte’s matrix-tree theorem (Smith & Smith, 2007; McDonald & Satta, 2007, Koo et al., 2007)
Nonprojective Parsing

• We can only use spanning trees and matrix inversion for inference when the features are all \textit{arc-local}.

• McDonald and Satta (2007): second-order features make nonprojective parsing NP-hard.

• Goal: efficient nonprojective parsing with arbitrary features
  
  • Max-inference and loss-augmented-max-inference only.
Concise ILP Formulation

• Riedel and Clarke (2006): nonprojective parsing as ILP, with exponentially many constraints. Cutting plane solution. Core “attachment” variables:

\[ z_{i,j} = 1 \text{ iff word } i \text{ is the child of word } j \]

• Martins, Smith, and Xing (2009): nonprojective parsing as concise ILP
  • Replace “acyclic” with “connected” constraints
  • Single commodity flow to impose the tree constraint (Magnanti and Wolsey, 1994).

• Loss function (attachment accuracy) also factors well, so max- and loss-augmented inference are both possible.

• Extensions:
  • Projectivity: use multi-commodity flow (constraint or features)
  • Higher-order features using linearization trick for Boolean formulas: “grandchild,” “sibling,” “valence” features
LP Relaxation

convex hull; vertices are the set of valid trees

“fractional” parse

valid parse

outer polytope (concise)
Parsing

• First solve relaxed LP.

• If the solution is integral, it is the best tree; we’re done.

• If not, project using Chu-Liu-Edmonds; we get a nearby approximate best tree.
  
  • (This is the more expensive case we’d like to avoid.)
Relaxation Gap

- Original max margin objective: \( \min_w \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{t=1}^{m} r_t(w) \)

- Hypothesis: relaxation gap approximates computational cost.
Relaxation Gap

• Original max margin objective:
  \[
  \min_w \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{t=1}^{m} r_t(w)
  \]

• Hypothesis: relaxation gap approximates computational cost.

• New objective:
  \[
  \min_w \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{t=1}^{m} r_t(w) + \eta \left( \frac{1}{m} \sum_{t=1}^{m} \bar{r}_t(w) - r_t(w) \right)
  \]
  \[
  = \min_w \frac{\lambda}{2} \|w\|^2 + \frac{1 - \eta}{m} \sum_{t=1}^{m} r_t(w) + \frac{\eta}{m} \sum_{t=1}^{m} \bar{r}_t(w)
  \]

• Stochastic online solution (building on Ratliff et al., 2006): solve approximate (with probability \(\eta\)) or exact (with probability \(1 - \eta\)) loss-augmented inference
Encouraging Integer Solutions

• A geometric interpretation of the new objective:

\[ \tilde{Z}_\eta = (1 - \eta) \tilde{Z} + \eta \tilde{Z} \]

• We still have fractional solutions, but they are closer to the integral ones.
Dependency Parsing State of the Art

Baselines:

- **MST** (McDonald, Lerman, & Pereira, 2006)
- **Stacking** Malt and MST (Martins, Das, Smith, & Xing, 2008)

Our method:

- **ILP** (Martins, Smith, & Xing, 2009)

<table>
<thead>
<tr>
<th>Language</th>
<th>MST</th>
<th>Stacking</th>
<th>ILP exact</th>
<th>ILP approx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danish</td>
<td>90.6</td>
<td>91.5</td>
<td>91.2</td>
<td>91.0</td>
</tr>
<tr>
<td>Dutch</td>
<td>84.1</td>
<td>84.8</td>
<td>85.6</td>
<td>85.4</td>
</tr>
<tr>
<td>English</td>
<td>90.9</td>
<td>-</td>
<td>91.2</td>
<td>91.1</td>
</tr>
<tr>
<td>Portuguese</td>
<td>91.4</td>
<td>92.1</td>
<td>91.4</td>
<td>91.4</td>
</tr>
<tr>
<td>Slovene</td>
<td>83.7</td>
<td>85.1</td>
<td>85.6</td>
<td>85.4</td>
</tr>
<tr>
<td>Swedish</td>
<td>89.1</td>
<td>90.5</td>
<td>90.6</td>
<td>90.5</td>
</tr>
<tr>
<td>Turkish</td>
<td>75.3</td>
<td>76.4</td>
<td>76.3</td>
<td>76.3</td>
</tr>
</tbody>
</table>
Increasing $\eta$ Improves Test-Time Relaxation Gap

Relaxation Gap at Test Time vs $\eta$ at Train Time

- $\approx$ More Exact
- $\eta$ @Train
- More Relaxed $\gg$

- Line: Relaxation Gap @Test
Increasing $\eta$ Improves Test-Time Speed
Example

- Nonprojective arc “when ← frightened” should be dispreferred

\$ When the little guy gets frightened, the big guys hurt badly \$
Example

- “learned → lesson” and “learned → in” are likely to be siblings

$ He added: “We learned a lesson in 1987 about volatility.”
Conclusion

• Natural affinity: probabilistic models and linguistic structures
  • Synthesis lecture in 2010 (hopefully): *Linguistic Structure Prediction*

• Today’s linguistic models **require** approximate inference

• Important to remember: even our annotated data aren’t perfect or uncontroversial.
  • Computational representations of linguistic structure are always evolving!

• Much more to be done in developing generic, declarative frameworks for approximating hard NLP problems.
Acknowledgments

- Kevin Gimpel
- André Martins
- Eric Xing
- NSF, DARPA, IBM, Google, HP Labs, Yahoo, Hertz Foundation
Publications Discussed in this Talk

• **Cube Summing, Approximate Inference with Non-Local Features, and Dynamic Programming without Semirings**
  Kevin Gimpel and Noah A. Smith

• **Feature-Rich Translation by Quasi-Synchronous Lattice Parsing**
  Kevin Gimpel and Noah A. Smith

• **Concise Integer Linear Programming Formulations for Dependency Parsing**
  André F. T. Martins, Noah A. Smith, and Eric P. Xing

• **Polyhedral Outer Approximations with Application to Natural Language Parsing**
  André F. T. Martins, Noah A. Smith, and Eric P. Xing