Hidden–Variable Models for Discriminative Reranking

Terry Koo and Michael Collins

{maestro|mcollins}@csail.mit.edu
Overview of reranking

The reranking approach

Use a baseline model to get the $N$-best candidates
“Rerank” the candidates using a more complex model

Parse reranking

Collins (2000): 88.2% $\Rightarrow$ 89.8%
Charniak and Johnson (2005): 89.7% $\Rightarrow$ 91.0%
Talk by Brooke Cowan in 7B: 83.6% $\Rightarrow$ 85.1%

Also applied to

MT (Och and Ney, 2002; Shen et al., 2004)
NL Generation (Walker et al., 2001)
Representing NLP structures

Proper representation is critical to success

Hand-crafted feature vector representations

$$\Phi\left(\begin{array}{c}0, 1, 2, 0, 0, 3, 0, 1\end{array}\right)$$

Features defined through kernels

$$K\left(\begin{array}{c}0, 1, 2, 0, 0, 3, 0, 1\end{array}\right) = \Phi\left(\begin{array}{c}0, 1, 2, 0, 0, 3, 0, 1\end{array}\right) \cdot \Phi\left(\begin{array}{c}0, 1, 2, 0, 0, 3, 0, 1\end{array}\right)$$

This talk: A new approach using hidden variables
Two facets of lexical items

Different lexical items can have similar meanings, e.g. *president* and *chairman*

Clustering: \( president, chairman \in \text{NounCluster}_4 \)

A single lexical item can have different meanings, e.g. [river] *bank* vs [financial] *bank*

Refinement: \( bank_1, bank_2 \in \text{bank} \)

Model clusterings and refinements as hidden variables that support the reranking task
Highlights of the approach

Conditional log–linear model with hidden variables

Dynamic programming is used for training and decoding

Clustering and refinement done automatically using a discriminative criterion
Overview of talk

Motivation

Design

General form of the model

Training and decoding efficiently

Creating specific instantiations

Results

Discussion

Conclusion
The parse reranking framework

Sentences $s_i$ for $1 \leq i \leq n$

$s_1$: Pierre Vinken, 61 years old, will join ...  
$s_2$: Mr. Vinken is chairman of Elsevier N.V. ...  
$s_3$: Big Board Chairman John Phelan said yesterday ...

Each $s_i$ has candidate parses $t_{i,j}$ for $1 \leq j \leq n_i$

$t_{i,1}$ is the best candidate parse for $s_i$
The parse reranking framework

$t_{i,j}$ has phrase structure and dependency tree

Mr. Vinken is chairman of Elsevier N.V.
The parse reranking framework

\( t_{i,j} \) has phrase structure and dependency tree

Mr. Vinken is chairman of Elsevier N.V.
Adding hidden variables

Hidden–value domains $H_w(t_{i,j})$ for $1 \leq w \leq \text{len}(s_i)$
Adding hidden variables

Assignment $h \in H_1(t_{i,j}) \times \ldots \times H_{\text{len}(s_i)}(t_{i,j})$
Marginalized probability model

$\Phi(t_{i,j}, h)$ produces a descriptive vector of feature occurrence counts, e.g.

$\Phi_2(t_{i,j}, h) = \text{Count}(\text{chairman has hidden value NN}_1)$

$\Phi_{13}(t_{i,j}, h) = \text{Count}(\text{NNP}_2 \text{ is a direct object of VB}_1)$

$\Phi_{19}(t_{i,j}, h) = \text{Count}(\text{NN}_1 \text{ coordinates with NN}_2)$
Marginalized probability model

Log–linear distribution over \((t_{i,j}, h)\) with parameters \(\Theta\):

\[
p(t_{i,j}, h \mid s_i, \Theta) = \frac{e^{\Phi(t_{i,j}, h) \cdot \Theta}}{\sum_{j', h'} e^{\Phi(t_{i,j'}, h') \cdot \Theta}}
\]

Marginalize over assignments \(h\):

\[
p(t_{i,j} \mid s_i, \Theta) = \sum_h p(t_{i,j}, h \mid s_i, \Theta)
\]
Optimizing the parameters

Define loss as negative log-likelihood

\[ L(\Theta) = -\sum_{i=1}^{n} \log p(t_{i,1} | s_i, \Theta) \]

Minimize \( L(\Theta) \) through gradient descent

\[
\frac{\partial L}{\partial \Theta} = -\sum_{i}^{n} \sum_{h} \left[ p(h | t_{i,1}, s_i, \Theta) \Phi(t_{i,1}, h) \right] \\
+ \sum_{i,j} p(t_{i,j} | s_i, \Theta) \sum_{h} p(h | t_{i,j}, s_i, \Theta) \Phi(t_{i,j}, h)
\]
Overview of talk

Motivation

Design

- General form of the model
- Training and decoding efficiently
- Creating specific instantiations

Results

Discussion

Conclusion
Problems with efficiency

\[ |H_1(t_{i,j}) \times \ldots \times H_{\text{len}(s_i)}(t_{i,j})| \text{ grows exponentially, so training the model is intractable:} \]

\[
\frac{\partial L}{\partial \Theta} = - \sum_i \sum_h p(h \mid t_{i,1}, s_i, \Theta) \Phi(t_{i,1}, h) + \sum_{i,j} p(t_{i,j} \mid s_i, \Theta) \sum_h p(h \mid t_{i,j}, s_i, \Theta) \Phi(t_{i,j}, h)
\]

Decoding the model is also intractable:

\[
p(t_{i,j} \mid s_i, \Theta) = \sum_h p(t_{i,j}, h \mid s_i, \Theta)
\]
Problems with efficiency

\[ |H_1(t_{i,j}) \times \ldots \times H_{\text{len}(s_i)}(t_{i,j})| \text{ grows exponentially, so training the model is intractable:} \]

\[
\frac{\partial L}{\partial \Theta} = - \sum_i \sum_h p(h \mid t_{i,1}, s_i, \Theta) \Phi(t_{i,1}, h) \\
+ \sum_{i,j} p(t_{i,j} \mid s_i, \Theta) \sum_h p(h \mid t_{i,j}, s_i, \Theta) \Phi(t_{i,j}, h)
\]

Decoding the model is also intractable:

\[
p(t_{i,j} \mid s_i, \Theta) = \sum_h p(t_{i,j}, h \mid s_i, \Theta)
\]
Locality constraint on features

Features have pairwise local scope on hidden variables

Features still have global scope on non-hidden information

$\Phi$ can be factored into local feature vectors, allowing dynamic programming
Local feature vectors

Define two kinds of local feature vector $\phi$:

Single-variable $\phi(t_{i,j}, w, h_w)$ look at a single hidden variable

Pairwise $\phi(t_{i,j}, u, v, h_u, h_v)$ look at two hidden variables in a dependency relationship
Local feature vectors

\( \Phi(t_{i,j}, h) \) looks at every hidden variable

Mr. Vinken is chairman of Elsevier N.V.
\[ \phi(t_{i,j}, \text{chairman}, \text{NN}_3) \text{ only sees } \text{NN}_3 \]
Local feature vectors

\( \phi(t_{i,j}, \textit{chairman}, \textit{of}, \text{NN}_3, \text{IN}_2) \) sees \text{NN}_3 and \text{IN}_2

Mr. Vinken is chairman of Elsevier N.V.
Local feature vectors

Rewrite global $\Phi$ as a sum over local $\phi$

$$\Phi(t_{i,j}, h) = \sum_{w \in t_{i,j}} \phi(t_{i,j}, w, h_w) + \sum_{(u,v) \in D(t_{i,j})} \phi(t_{i,j}, u, v, h_u, h_v)$$
Local feature vectors

Rewrite global $\Phi$ as a sum over local $\phi$

$$\Phi(t_{i,j}, h) = \sum_{w \in t_{i,j}} \phi(t_{i,j}, w, h_w) + \sum_{(u,v) \in D(t_{i,j})} \phi(t_{i,j}, u, v, h_u, h_v)$$
Applying belief propagation

New restrictions enable dynamic–programming approaches, e.g. belief propagation

BP generalizes the forward–backward algorithm from a chain to a tree

Runtime $O(\text{len}(s_i)H^2)$, $H = \max |H_w(t_{i,j})|$

BP efficiently computes

$$\sum_{h} p(t_{i,j}, h \mid s_i, \Theta)$$

$$\sum_{h} p(h \mid t_{i,j}, s_i, \Theta) \Phi(t_{i,j}, h)$$
Overview of talk

Motivation

Design

General form of the model

Training and decoding efficiently

Creating specific instantiations

Results

Discussion

Conclusion
Two areas for choice in the model

Definition of the hidden–value domains $H_w(t_{i,j})$

Definition of the feature vectors $\phi$
Hidden–value domains

Lexical domains allow word refinement

Mr. \_1 \hspace{1cm} \text{Vinken}_1 \hspace{1cm} \text{is}_1 \hspace{1cm} \text{chairman}_1 \hspace{1cm} \text{of}_1 \hspace{1cm} \text{Elsevier}_1 \hspace{1cm} \text{N.V.}_1

Mr. \_2 \hspace{1cm} \text{Vinken}_2 \hspace{1cm} \text{is}_2 \hspace{1cm} \text{chairman}_2 \hspace{1cm} \text{of}_2 \hspace{1cm} \text{Elsevier}_2 \hspace{1cm} \text{N.V.}_2

Mr. \_3 \hspace{1cm} \text{Vinken}_3 \hspace{1cm} \text{is}_3 \hspace{1cm} \text{chairman}_3 \hspace{1cm} \text{of}_3 \hspace{1cm} \text{Elsevier}_3 \hspace{1cm} \text{N.V.}_3
Hidden–value domains

Lexical domains allow word refinement

Mr. Vinken is chairman of Elsevier N.V.
Hidden–value domains

Part-of-speech domains allow word clustering
Hidden–value domains

Part-of-speech domains allow word clustering

Mr. Vinken is chairman of Elsevier N.V.
Hidden–value domains

Part-of-speech domains allow word clustering

Mr. Vinken is chairman of Elsevier N.V.
Hidden–value domains

Supersense domains model WordNet ontology
(Ciaramita and Johnson, 2003; Miller et al., 1993)

Mr. Vinken is chairman of Elsevier N.V.
Hidden–value domains

Supersense domains model WordNet ontology
(Ciaramita and Johnson, 2003; Miller et al., 1993)

Mr. Vinken

is

chairman

of Elsevier N.V.
Hidden–value domains

Supersense domains model WordNet ontology
(Ciaramita and Johnson, 2003; Miller et al., 1993)
Hidden–value domains

Supersense domains model WordNet ontology
(Ciaramita and Johnson, 2003; Miller et al., 1993)
Hidden–value domains

Hidden–value domains that didn’t work well

Word clustering without part-of-speech subdivisions

WordNet hyper/hyponym ontology

Domains containing mixed values
Examples of features

The highest nonterminal headed by chairman

(\text{NN}_3, \text{Word}=\text{chairman}, \text{Highest Nonterminal}=\text{NP}) \in \phi(t_{i,j}, \text{chairman}, \text{NN}_3)
Examples of features

(\(VB_1, NN_3, \text{Rule}= VP \rightarrow VB \, NP\)) \(\in \phi(t_{i,j}, is, \text{chairman}, VB_1, NN_3)\)

The governing rule

(S)

(VP)

(NP)

(PP)

(NP)
Overview of talk

Motivation
Design
Results
Discussion
Conclusion
Experimental Setup

$N$-best lists generated by Collins parser, $n_i \approx 30$

Training set: WSJ sections 2–21
Development set: WSJ section 0
Test set: WSJ sections 22-24
Final test models

Two baseline models
- The Collins (1999) base parser
- The Collins (2000) reranker

Two mixed models
- MIX combined clustering, refinement, and WordNet
- MIX+ augments MIX with features of Collins reranker
## Results on Sections 22–24

<table>
<thead>
<tr>
<th></th>
<th>LR</th>
<th>LP</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collins parser</td>
<td>88.19</td>
<td>88.60</td>
<td>88.39</td>
</tr>
<tr>
<td>MIX</td>
<td>89.41</td>
<td>89.87</td>
<td>89.64</td>
</tr>
<tr>
<td>Collins reranker</td>
<td>89.46</td>
<td>90.07</td>
<td>89.76</td>
</tr>
<tr>
<td>MIX+</td>
<td>89.78</td>
<td>90.29</td>
<td>90.03</td>
</tr>
</tbody>
</table>

All comparisons except MIX vs. Collins reranker are significant at $p \leq 0.01$ using the sign test.
### Results on Sections 22–24

<table>
<thead>
<tr>
<th></th>
<th>LR</th>
<th>LP</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collins parser</td>
<td>88.19</td>
<td>88.60</td>
<td>88.39</td>
</tr>
<tr>
<td><strong>MIX</strong></td>
<td>89.41</td>
<td>89.87</td>
<td>89.64</td>
</tr>
<tr>
<td>Collins reranker</td>
<td>89.46</td>
<td>90.07</td>
<td>89.76</td>
</tr>
<tr>
<td><strong>MIX+</strong></td>
<td>89.78</td>
<td>90.29</td>
<td>90.03</td>
</tr>
</tbody>
</table>

All comparisons except **MIX** vs. Collins reranker are significant at $p \leq 0.01$ using the sign test.
## Results on Sections 22–24

<table>
<thead>
<tr>
<th></th>
<th>LR</th>
<th>LP</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collins parser</td>
<td>88.19</td>
<td>88.60</td>
<td>88.39</td>
</tr>
<tr>
<td><em>MIX</em></td>
<td>89.41</td>
<td>89.87</td>
<td>89.64</td>
</tr>
<tr>
<td>Collins reranker</td>
<td>89.46</td>
<td>90.07</td>
<td>89.76</td>
</tr>
<tr>
<td><em>MIX</em>+</td>
<td>89.78</td>
<td>90.29</td>
<td>90.03</td>
</tr>
</tbody>
</table>

All comparisons except *MIX* vs. Collins reranker are significant at $p \leq 0.01$ using the sign test.
Overview of talk

Motivation
Design
Results
Discussion
Conclusion
Previous work

Parsing approaches that use hidden variables

Riezler et al. (2002)
Matsuzaki et al. (2005)

Differences with our approach

Use of reranking
Definition of hidden variables
Use of belief propagation
Using packed representations

Candidates $t_{i,j}$ represented as a packed forest

Compact representation of many parse trees

Packed representation forces local scope

Features would become locally scoped on non-hidden information

Decoding becomes NP-hard, must approximate with Viterbi (cf. Matsuzaki et al., 2005):

$$
\underset{t_{i,j}}{\arg\max} \ p(t_{i,j} \mid s_i, \Theta) \approx \underset{t_{i,j}, h}{\arg\max} \ p(t_{i,j}, h \mid s_i, \Theta)
$$
Empirical analysis of hidden values

The model makes hidden–value assignments on the basis of the reranking criterion

\[
\text{i.e. maximize } \sum \log p(t_{i,1} \mid s_i, \Theta)
\]

The empirical distribution of assignments shows linguistically reasonable trends
Concluding remarks

The hidden–variable model defines a new representation for NLP structures

Conditional log–linear model with hidden variables

BP enables efficient and exact training and decoding

Significant improvement over Collins (2000)
Example

I [will [give/VB₂ an example]]

I expected [to [give/VB₁ an example]]

I expected [to/TO₄ [give an example]]

You expected [me [to/TO₁,₅ [give an example]]]