Low-Dimensional Discriminative Reranking

Jagadeesh Jagarlamudi  and Hal Daume III
University of Maryland, College Park
### Discriminative Reranking

- Useful for many NLP tasks
- Enables us to use arbitrary features
- Search space is limited

<table>
<thead>
<tr>
<th>Score</th>
<th>Buyers</th>
<th>stepped</th>
<th>in</th>
<th>to</th>
<th>the</th>
<th>futures</th>
<th>pit</th>
<th>.</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.1947</td>
<td>NNS</td>
<td>VBD</td>
<td>RP</td>
<td>TO</td>
<td>DT</td>
<td>NNS</td>
<td>NN</td>
<td>.</td>
<td>0.12</td>
</tr>
<tr>
<td>-6.8068</td>
<td>NNS</td>
<td>VBD</td>
<td>RB</td>
<td>TO</td>
<td>DT</td>
<td>NNS</td>
<td>NN</td>
<td>.</td>
<td>0.12</td>
</tr>
<tr>
<td>-7.0514</td>
<td>NNS</td>
<td>VBD</td>
<td>IN</td>
<td>TO</td>
<td>DT</td>
<td>NNS</td>
<td>NN</td>
<td>.</td>
<td>0</td>
</tr>
<tr>
<td>-7.1408</td>
<td>NNS</td>
<td>VBD</td>
<td>RP</td>
<td>TO</td>
<td>DT</td>
<td>NNS</td>
<td>VB</td>
<td>.</td>
<td>0.25</td>
</tr>
<tr>
<td>-13.752</td>
<td>NNS</td>
<td>VBD</td>
<td>RB</td>
<td>TO</td>
<td>DT</td>
<td>NNS</td>
<td>VB</td>
<td>.</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Related Work

- **POS Tagging**
  - Collins 2002, Huang et al. 2007

- **Dependency Parsing**
  - Charniak and Johnson 2005, Shen et al. 2004

- **Machine Translation**
  - Watanabe et al. 2007

- **Use joint features defined on input and output pair**
  - Increases the number of features
  - Feature engineering becomes essential
Outline

- Introduction

- Low-Dimensional Reranking Models
  - Generative-Style Model
  - Discriminative-Style Model
  - Softened-Discriminative Model

- Experiments

- Conclusion
Low-dimensional Reranking

1) Run independent feature extractors

Buyers stepped in to the futures pit.

\[ X = [x_1, ..., x_n]; \quad Y = [y_1, \ldots, y_n] \]

2) Use training data to find low-dimensional space

- \( X = [x_1, \ldots, x_n] \);
- \( Y = [y_1, \ldots, y_n] \)
Ranking Models

1) Generative Style Model

2) Discriminative Style Model

3) Softened-Discriminative Model
Ranking Models

1) Generative Style Model
   - Uses only reference outputs i.e. $\{x_i, y_i\} \quad i=1 \ldots n$

2) Discriminative Style Model
   - Uses reference and candidate outputs $\{x_i, y_i, y_{ij}\} \quad i=1 \ldots n$
   - Ensures the highest scoring output is the reference

3) Softened-Discriminative Model
   - Difficult to solve the Discriminative-Style Model
   - Relaxed version which is tractable
1) Generative-Style Model

- Uses only input, ref output pairs \( \{x_i, y_i\} \) \( i = 1 \ldots n \)
- Aim to find \( A \) and \( B \) such that …
  - Projection is nothing but \( A^T x \) and \( B^T y \)
1) Generative-Style Model

- Uses only input, ref output pairs \( \{ x_i, y_i \} \ i = 1 \ldots n \)
- Aim to find \( a \) and \( b \) such that …
  - Projection is nothing but \( a^T X \) and \( b^T Y \)

\[
(a, b) = \arg \max \cos(X^T a, Y^T b)
\]

\[
\arg \max \quad a^T X Y^T b \quad \text{s.t.} \quad \| Y^T a \| = \| Y^T b \| = 1
\]

- Can be solved exactly using Gen. Eigenvalue problem
  - Use the first \( k \) eigenvectors to form \( A \) and \( B \)

Similarity measured using Cosine Angle
Discriminative-Style Models – Intuition

- Use ref. \((y_1)\) and candidate \((\hat{y}_1)\) outputs
- Aim to find A and B s.t. ref. output is highest scoring

\[
y_i = \arg\max_{\hat{y}_{ij}} \cos(A^T x_i, B^T \hat{y}_{ij})
\]

Similarity measured using Cosine Angle
Discriminative-Style Models

- Use ref. $y_1$ and candidate $\hat{y}_1$ outputs
- Aim to find $A$ and $B$ s.t. ref. output is highest scoring

$$y_i = \arg\max_{\hat{y}_{ij}} \cos(A^T x_i, B^T \hat{y}_{ij})$$
Discriminative-Style Models

- Use ref. \( y_1 \) and candidate \( \hat{y}_1 \) outputs
- Aim to find \( A \) and \( B \) s.t. ref. output is highest scoring

\[
y_i = \arg\max_{\hat{y}_{ij}} \cos(A^T x_i, B^T \hat{y}_{ij})
\]

1) Observe that the score is: \( a^T x_i \hat{y}_{ij} b \)

2) Add constraints to penalize incorrect ones

\[
m_{ij} = a^T x_i y_{ij} b - a^T x_i \hat{y}_{ij} b \quad m_{ij} \geq 1 - \frac{\xi_i}{L_{ij}}
\]
2) Discriminative-Style Model

- Enforce the individual constraints

\[
\begin{align*}
\arg \max_{a,b} \quad & \frac{1 - \lambda}{\lambda} a^T X Y^T b \\
\text{s.t.} \quad & a^T XX^T a = 1 \\
& b^T YY^T b = 1
\end{align*}
\]
2) Discriminative-Style Model

- Enforce the individual constraints

\[
\arg\max_{a,b} \quad \frac{1 - \lambda}{\lambda} a^T X Y^T b
\]

\[
a^T X X^T a = 1 \quad b^T Y Y^T b = 1
\]

\[
m_{ij} \geq 1
\]
2) Discriminative-Style Model

- Enforce the individual constraints

\[
\arg \max_{a,b} \quad \frac{1 - \lambda}{\lambda} \ a^T \ X \ Y^T \ b
\]
\[
a^T \ X \ X^T \ a \ = \ 1 \quad b^T \ Y \ Y^T \ b \ = \ 1
\]
\[
m_{ij} \ \geq \ 1 - \frac{\xi_i}{L_{ij}}
\]
2) Discriminative-Style Model

- Enforce the individual constraints

\[
\arg \max_{a,b} \frac{1-\lambda}{\lambda} a^T X Y^T b - \sum_i \xi_i \\
a^T X X^T a = 1 \quad b^T Y Y^T b = 1 \\
m_{ij} \geq 1 - \frac{\xi_i}{L_{ij}}
\]
2) Discriminative-Style Model

- Enforce the individual constraints

\[
\arg\max_{a,b,\xi \geq 0} \frac{1 - \lambda}{\lambda} a^TXY^Tb - \sum_i \xi_i
\]

\[
a^TXX^Ta = 1 \quad b^TYY^Tb = 1
\]

\[
m_{ij} \geq 1 - \frac{\xi_i}{L_{ij}}
\]

- Harder to solve exactly
3) Softened-Discriminative Model

- Discriminative Model
  \[
  \arg \max_{a, b, \xi \geq 0} \frac{1-\lambda}{\lambda} a^T X Y^T b - \sum_i \xi_i
  \]
  \[
  \text{s.t. length constraints} \quad m_{ij} \geq 1 - \frac{\xi_i}{L_{ij}}
  \]

- Softened-Discriminative Model
  \[
  \arg \max_{a, b} \left(1-\lambda\right) a^T X Y^T b + \lambda \sum_{ij} L_{ij} m_{ij}
  \]
  \[
  \text{s.t. length constraints}
  \]

- Similar to Generative Model it can be solved exactly
  - We use it to solve Discriminative-Style Model
Optimizing Discriminative-Style Model

- \( \alpha_{ij} = L_{ij} \) // Initialization

- Repeat

\[
[A^{(i)}, B^{(i)}] = \text{Softened-Disc}(X, Y, \alpha_{ij}) \]

// Get the current soln.

\[
m_{ij} = a^T x_i y_i^T b - a^T x_i \hat{y}_{ij} b
\]

// Compute margins

\[
\psi_{ij} = (1 - m_{ij}) L_{ij}
\]

// Potential Slack

\[
\xi_i = \min \{0, \psi_{ij} \text{ s.t. } \psi_{ij} > 0\}
\]

// Compute Slack

If \( \xi_i > 0 \)

\[
d_{ij} = m_{ij} - \left(1 - \frac{\xi_i}{L_{ij}}\right)
\]

// Update the Lagrangian variables

\[
\alpha_{ij} \leftarrow \alpha_{ij} - \gamma d_{ij}
\]

End if

- Until convergence // Slack doesn't change
Outline

- Introduction
- Low-Dimensional Reranking Models
  - Generative-Style Model
  - Discriminative-Style Model
  - Softened-Discriminative Model
- Experiments
- Conclusion
Reranking for POS Tagging

- HMM Trigram tagger is used to generate 10-best list
  - Use multi-fold cross validation
- Training
  - Use one of our models to find mappings A and B

<table>
<thead>
<tr>
<th>Score</th>
<th>Buyers stepped in to the futures pit</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.1947</td>
<td>NNS VBD RP TO DT NNS NN .</td>
<td>0.12</td>
</tr>
<tr>
<td>-6.8068</td>
<td>NNS VBD RB TO DT NNS NN .</td>
<td>0.12</td>
</tr>
<tr>
<td>-7.0514</td>
<td>NNS VBD IN TO DT NNS NN .</td>
<td>0</td>
</tr>
<tr>
<td>-7.1408</td>
<td>NNS VBD RP TO DT NNS VB .</td>
<td>0.25</td>
</tr>
<tr>
<td>-13.752</td>
<td>NNS VBD RB TO DT NNS VB .</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Reranking for POS Tagging

- HMM Trigram tagger is used to generate 10-best list
  - Use multi-fold cross validation

- Training
  - Use one of our models to find mappings A and B

- Reranking

<table>
<thead>
<tr>
<th>Score</th>
<th>Buyers stepped in to the futures pit .</th>
<th>Score</th>
<th>Buyers stepped in to the futures pit .</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.1947</td>
<td>NNS  VBD  RP  TO  DT  NNS  NN  .</td>
<td>-6.8068</td>
<td>NNS  VBD  RB  TO  DT  NNS  NN  .</td>
</tr>
<tr>
<td>-7.0514</td>
<td>NNS  VBD  IN  TO  DT  NNS  NN  .</td>
<td>-7.1408</td>
<td>NNS  VBD  RP  TO  DT  NNS  VB  .</td>
</tr>
<tr>
<td>-13.752</td>
<td>NNS  VBD  RB  TO  DT  NNS  VB  .</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Reranking for POS Tagging

- HMM Trigram tagger is used to generate 10-best list
  - Use multi-fold cross validation
- Training
  - Use one of our models to find mappings A and B
- Reranking
  - For every candidate tag sequence
    → Compute cosine similarity in the sub-space
    → Combine this score with Viterbi decoding score
  - Interpolation parameter is tuned
Experimental Results

• Data Sets
  → CoNLL shared task on Dependency Parsing
  → Use the fine-grained POS tags

• Baseline system
  → HMM Trigram Tagger
    → Thede and Harper, 1999

• Baseline Rerankers
  → Collins and Koo, 2005
  → Its regularized version Huang et al. 2007

<table>
<thead>
<tr>
<th>Language</th>
<th># sent.</th>
<th># words</th>
<th>Train.</th>
<th>Dev.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>15K</td>
<td>362K</td>
<td>2K</td>
<td>47K</td>
<td>1791</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>43K</td>
</tr>
<tr>
<td>Chinese</td>
<td>50K</td>
<td>292K</td>
<td>4K</td>
<td>26K</td>
<td>3647</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25K</td>
</tr>
<tr>
<td>French</td>
<td>9K</td>
<td>254K</td>
<td>2K</td>
<td>57K</td>
<td>1351</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40K</td>
</tr>
<tr>
<td>Swedish</td>
<td>8K</td>
<td>137K</td>
<td>2K</td>
<td>31K</td>
<td>1431</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>28K</td>
</tr>
</tbody>
</table>
Features

• For our models:
  → Sentence is represented as bag of suffixes of length 2 to 4 ($x$)
  → For POS tag-sequences, we use
    → Unigram and Bigram POS tags as features ($y$)

• For baseline Rerankers
  → Feature of the form (suffix, Tag$_0$), (suffix,Tag$_0$,Tag$_{-1}$)
## Results

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Chinese</th>
<th>French</th>
<th>Swedish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>96.15</td>
<td>92.31</td>
<td>97.41</td>
<td>93.23</td>
</tr>
<tr>
<td>Collins</td>
<td>96.06</td>
<td>92.81</td>
<td>97.35</td>
<td>93.44</td>
</tr>
<tr>
<td>Regularized</td>
<td>96.00</td>
<td>92.88</td>
<td>97.38</td>
<td>93.35</td>
</tr>
<tr>
<td>Oracle</td>
<td>98.39</td>
<td>98.19</td>
<td>99.00</td>
<td>96.48</td>
</tr>
</tbody>
</table>

- Oracle improvements are less compared to Parsing
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>English</th>
<th>Chinese</th>
<th>French</th>
<th>Swedish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>96.15</td>
<td>92.31</td>
<td>97.41</td>
<td>93.23</td>
</tr>
<tr>
<td>Collins</td>
<td>96.06</td>
<td>92.81</td>
<td>97.35</td>
<td>93.44</td>
</tr>
<tr>
<td>Regularized</td>
<td>96.00</td>
<td>92.88</td>
<td>97.38</td>
<td>93.35</td>
</tr>
<tr>
<td>Generative</td>
<td>96.24</td>
<td>92.95</td>
<td>97.43</td>
<td>93.26</td>
</tr>
<tr>
<td>Softened-Disc</td>
<td>96.32</td>
<td>92.87</td>
<td>97.53</td>
<td>93.36</td>
</tr>
<tr>
<td>Discriminative</td>
<td>96.3</td>
<td>92.91</td>
<td>97.53</td>
<td>93.36</td>
</tr>
<tr>
<td>Oracle</td>
<td>98.39</td>
<td>98.19</td>
<td>99.00</td>
<td>96.48</td>
</tr>
</tbody>
</table>

- One of our models performs better
Results

- Performance is not very sensitive to hyperparams
- Combination with Viterbi Score is crucial

<table>
<thead>
<tr>
<th>Type</th>
<th>English</th>
<th>Chinese</th>
<th>French</th>
<th>Swedish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generative</td>
<td>96.24</td>
<td>92.95</td>
<td>97.43</td>
<td>93.26</td>
</tr>
<tr>
<td>Softened-Disc</td>
<td>96.32</td>
<td>92.87</td>
<td>97.53</td>
<td>93.36</td>
</tr>
<tr>
<td>Discriminative</td>
<td>96.3</td>
<td>92.91</td>
<td>97.53</td>
<td>93.36</td>
</tr>
<tr>
<td>Generative</td>
<td>94.83</td>
<td>89.89</td>
<td>96.10</td>
<td>91.89</td>
</tr>
<tr>
<td>Softened-Disc</td>
<td>95.04</td>
<td>89.61</td>
<td>95.97</td>
<td>91.95</td>
</tr>
<tr>
<td>Discriminative</td>
<td>94.95</td>
<td>89.76</td>
<td>95.82</td>
<td>92.11</td>
</tr>
</tbody>
</table>

- Performance drops by more than 2%
Conclusions

• Family of rerankers based on dim. reduction
  - Max-margin CCA [Szedmak et al. 2007]
  - Max-margin regression [Szedmak et al. 2006, Wang et al. 2007]
  - Supervised Semantic Hashing [Bai et al. 2010, Grangier et al. 2006]

• Learns a linear classifier in the joint feature space
  - But weights are constrained
  - Perfect when defining the joint features is not clear

• Scalability
  - Online SVD algorithms [Halko et al. 2009]
  - Adaptive variants of CCA [Yger et al. 2012]
Thank You !!
Performance – with regularization param. (\(\text{tau}\))

Discriminative Softened-Disc Generative Baseline

- Discriminative
- Softened-Disc
- Generative
- Baseline
Performance – with weight param. (lambda)