Exploring an Auxiliary Distribution based approach to Domain Adaptation of a Syntactic Disambiguation Model

Barbara Plank and Gertjan van Noord

University of Groningen (RUG), The Netherlands
COLING Workshop on Cross-Framework and Cross-Domain Parser Evaluation

August 23, 2008
Research area: Parsing of Natural Language

A parser - Conceptual view

Disambiguation Component

- Selects parse from the (many) alternative hypotheses
- Statistical in nature; bases its decisions on a hand-parsed treebank
The Problem: Domain dependence of Parsing

- Disambiguation component highly dependent on training data
- Problem: Whenever test and training data differ, the performance of such a supervised system degrades considerably (Gildea, 2001)

<table>
<thead>
<tr>
<th>PCFG parsing / English</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSJ (newspaper)</td>
<td>89.5</td>
</tr>
<tr>
<td>Brown (fiction/non-fiction)</td>
<td>83.4 ↓</td>
</tr>
<tr>
<td>GENIA (biomedical)</td>
<td>76.3 ↓</td>
</tr>
</tbody>
</table>
The Problem: Domain dependence of Parsing

Possible solutions approaches:

1. Build a model for every domain we encounter.

Need for training data → expensive & unsatisfactory solution
The Problem: Domain dependence of Parsing

Possible solutions approaches:

1. Build a model for every domain we encounter.

   Need for training data → expensive & unsatisfactory solution

2. Adapt parsers from a source domain (e.g., news) to a target domain (e.g., biomedical)
   → Domain Adaptation
Recently gained attention - Approaches:

a. **Supervised**: Limited annotated resources in new domain
   - (Hara, 2005)
   - (Daume III, 2007)

b. **Semi-supervised**: No annotated resources in new domain
   - (Blitzer et al., 2006)
   - (McClosky et al., 2006)

→ In this study we focus on the *supervised* scenario.
Supervised Domain Adaptation

- **Out-domain** ≫ **In-domain**, both labeled

  - In-domain (Target data)
  - Out-domain (Source data)

- **Goal**: To overcome limited in-domain data, exploit already trained out-of-domain/general model

In this study:

- Exploit auxiliary distributions (Johnson & Riezler, 2000)
- Auxiliary distributions: originally suggested to incorporate lexical selectional preferences
Background: Alpino Parser

- Wide-coverage dependency parser for Dutch
- HPSG-style grammar rules, large hand-crafted lexicon
- Maximum Entropy Disambiguation Model:
  - Feature functions / weights
  - Estimation based on *Informative samples* (Osborne, 2000)

\[
p_{\theta}(\omega|s) = \frac{1}{Z_\theta} q_0 \exp \sum_{j=1}^{m} \theta_j f_j(\omega)
\]

- Output: Dependency Structure

*B. Plank & G. J. M. van Noord (RUG)*

Exploring Auxiliary Distributions

August 23, 2008

7 / 16
What are Auxiliary Distributions?

- Probability distribution(s) estimated from larger corpus
- Additional, real-valued feature(s) (auxiliary features):
  \[ f_{m+i} = \log Q_{i}(\omega) \]

  - Value/count of aux feature: logarithm of aux distribution
  - Several aux features can be integrated
  - Contribution scaled through estimated weight(s)
Auxiliary Distributions for Domain Adaptation

- Incorporate more general model, out-of-domain model into specific
- Add the probability it assigns to a given parse as auxiliary feature:

\[ f_{m+1} = -\log P_{OUT}(\omega|s) \]

Train a new model on the target data, augmented with this new feature.
Auxiliary Distributions for Domain Adaptation

- Incorporate more general model, out-of-domain model into specific
- Add the probability it assigns to a given parse as auxiliary feature:
  \[ f_{m+1} = -\log P_{OUT}(\omega|s) \]

Train a new model on the target data, augmented with this new feature.

An alternative: Model combination

Keep only two features under the MaxEnt framework:

\[ f_1 = -\log P_{OUT}(\omega|s), \quad f_2 = -\log P_{IN}(\omega|s) \]
Experimental design

Data

- General, out-of-domain: Alpino (newspaper text; 7,000 sentences)
- Domain-specific:
  - CLEF corpus (questions; 1,800 sentences)
  - CGN (spoken corpus; size varies, from 17 to 1,193 sentences)

Evaluation metric: Concept Accuracy

- Proportion of correct dependencies
- Similar to named dependency accuracy
- Allowing mismatches between the number of returned and treebank dependencies
Experiments & Results

Experiments on the QA data

- **In-domain**: train on CLEF (baseline)
- **Out-domain**: train on Alpino
- **Data combination**: train on CLEF ∪ Alpino
- **Auxiliary distribution**: add Alpino model as aux feature to CLEF
- **Model combination**: keep only two features
## Experiments and Results

### Experiments on the QA data

<table>
<thead>
<tr>
<th>Dataset size (#sents)</th>
<th>In-dom. CLEF</th>
<th>Out-dom. Alp</th>
<th>Data Comb. CLEF+Alp</th>
<th>Aux.distr. CLEF+Alp</th>
<th>Model Comb. C_aux+A_aux</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEF 2003 (446)</td>
<td>97.01</td>
<td>94.02</td>
<td>97.21</td>
<td>97.01</td>
<td>97.14</td>
</tr>
<tr>
<td>CLEF 2004 (700)</td>
<td>96.60</td>
<td>89.88</td>
<td>95.14</td>
<td>96.60</td>
<td>97.12</td>
</tr>
<tr>
<td>CLEF 2005 (200)</td>
<td>97.65</td>
<td>87.98</td>
<td>93.62</td>
<td>97.72</td>
<td>97.99</td>
</tr>
<tr>
<td>CLEF 2006 (200)</td>
<td>97.06</td>
<td>88.92</td>
<td>95.16</td>
<td>97.06</td>
<td>97.00</td>
</tr>
<tr>
<td>CLEF 2007 (200)</td>
<td>96.20</td>
<td>92.48</td>
<td>97.30</td>
<td>96.33</td>
<td>96.33</td>
</tr>
</tbody>
</table>

- Data combination could help in some cases
- Adding auxiliary feature does not help; achieves in-domain performance
- Simple Model combination performed better

Why did the auxiliary feature not work?
Examining possible causes

- **Ignored?** No, compared to other features, quite influential weight
Examining possible causes

- **Ignored?** No, compared to other features, quite influential weight
- **Not modeling properly out-of-domain model?** No, performance of model having only the aux feature is identical to out-domain model performance
Examining possible causes

- **Ignored?** No, compared to other features, quite influential weight
- **Not modeling properly out-of-domain model?** No, performance of model having only the aux feature is identical to out-domain model performance
- **Single feature to weak?** No, adding several auxiliary features did not help, either
Examining possible causes

- **Ignored?** No, compared to other features, quite influential weight

- **Not modeling properly out-of-domain model?** No, performance of model having only the aux feature is identical to out-domain model performance

- **Single feature to weak?** No, adding several auxiliary features did not help, either

- **What if we have smaller amounts of in-domain training data?**
Experiments & Results

Varying amount of training data

Varying amount of training data (CLEF 2004)

CA

% training data

Aux.distr. (CLEF+Alp_aux)
Out-dom (Alpino)
Mod.Comb. (CLEF_aux+Alpino_aux)
Results

- Performance even falls below the out-domain baseline (e.g., 1%)
- Reason for this drop: available amount of in-domain training data and the corresponding scaling of the feature’s weight
- CLEF domain too ’easy’? → examine approach on CGN (spoken data). Results were confirmed (details in paper).
Conclusions

- Exploiting a more general model to overcome the limited amount of in-domain data through auxiliary distributions does not help.
- Better results were obtained either without adaptation or by simple model combination.
- As soon as we have a reasonable (often small) amount of in-domain training data, just use that!
- Future work:
  - Investigate other approaches to parser adaptation, especially the semi-supervised case (no labeled target data).
  - What is meant by domain?
Thank you for your attention.