Basic SRL Taggers using AdaBoost

Lluís Màrquez and Xavier Carreras
TALP Research Center
Technical University of Catalonia

UIUC, June 9th, 2004
Problem Decomposition and Learning Architecture

- CoNLL-2004 datasets and setting.

- Proposition treated independently: each target verb generates a sequence to be annotated.

- Learning one binary classifier for each label (B-A\{0-5,A\}, B-AM-*, I-A\{0-5,A\}, I-AM-*, B-C-*, I-C-*, I-V, O) for a total of 75 labels. But only the 40 most frequent labels were treated.

• SRL tagging: sliding window based (sw) BIO tagging; left to right; greedy; also recurrent sliding window (rsw) with information about the left context.

• Sequence processing by “tokens”: tokens are the top-most syntactic elements in the region of the sentence considered (always according to the chunk and clause annotation predicted for the sentence). Training set: 19,992 sequences totaling 216,042 tokens.

• The sentence “regions” explored for finding arguments are: 1) left and right contexts of the immediate clause containing the target predicate; 2) Left contexts of the parent clauses containing the target predicate’s clause, up to the most general clause.
• Basic constraints ensured: B-I-O coherence. No embedding nor overlapping arguments. Arguments do not cross clause boundaries nor base chunk boundaries (except verb chunk containing the target predicate.

• Unicity of predicates not checked nor enforced; Continuation tags are not checked for a previous argument of the same type. No postprocessing performed.
An Example

***** development sentence #10
(S) +- (NP) +- [0: NNP Balcor]
  +- [1: , ,]
  +- (S) +- (NP) +- [2: WDT which]
  |   +- (S) +- (VP) +- [3: VBZ has]
  |   +- (NP) +- [4: NNS interests]
  |   +- (PP) +- [5: IN in]
  |   +- (NP) +- [6: JJ real]
  |       +- [7: NN estate]
  +- [8: , ,]
  +- (VP) +- [9: VBD said]
  +- (S) +- (NP) +- [10: DT the]
  |   |   +- [11: NN position]
  |   +- (VP) +- [12: VBZ is]
  |       +- [13: RB newly]
  |       +- [14: VBN created]
  +- [15: . .]

Target predicates are: *have*, *say*, and *create*.
say has arguments: A0_(0 8) and A1_(10 14).
An Example

>>>>>> target 10 3 V *have*: seq. #1
>>> region-PRE (0,1):
  1 B-A0 0 1 type:chunk NP
  2 0 1 1 type:word ,
>>> region-PRE (2,2):
  3 B-R-A0 2 1 type:chunk NP
>>> region-ICL (3,2):
>>> target predicate (3,3):
  4 B-V 3 1 has
>>> region-ICR (4,7):
  5 B-A1 4 1 type:chunk NP
  6 I-A1 5 1 type:chunk PP
  7 I-A1 6 2 type:chunk NP
>>> region-OUT (8,15):
  8 0 8 8 Not-treated
target 10 9 V *say*: seq. #2

region-ICL (0,8):
1 B-A0 0 1 type:chunk NP
2 I-A0 1 1 type:word ,
3 I-A0 2 6 type:clause S
4 I-A0 8 1 type:word ,

region-predicate (9,9):
5 B-V 9 1 said

region-ICR (10,15):
6 B-A1 10 5 type:clause S
7 0 15 1 type:word .
>>>>>> target 10 14 V *create*: seq. #3

>>> region-PRE (0,9):
  1 0 0 1 type:chunk NP
  2 0 1 1 type:word ,
  3 0 2 6 type:clause S
  4 0 8 1 type:word ,
  5 0 9 1 type:chunk VP

>>> region-ICL (10,13):
  6 B-A1 10 2 type:chunk NP
  7 0 12 1 type:word is
  8 B-AM-ADV 13 1 type:word newly

>>> target predicate (14,14):
  9 B-V 14 1 created

>>> region-ICR (15,14):

>>> region-OUT (15,15):
  10 0 15 1 Not-treated
### Upper Bounds on Global Performance

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>$F_{\beta=1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>99.09%</td>
<td>98.28%</td>
<td>98.68</td>
</tr>
<tr>
<td>Development</td>
<td>97.95%</td>
<td>96.22%</td>
<td>97.08</td>
</tr>
<tr>
<td>Test</td>
<td>97.32%</td>
<td>95.57%</td>
<td>96.44</td>
</tr>
</tbody>
</table>
Feature representation

(i) On the verb predicate:
- verb base form
- verb form
- verb POS
- type of VP in which verb is included: single/complex
- verb voice: active/passive

(ii) On the focus token \((s, e)\) and its context:
- type and head
- POS and word forms at positions: \(s - 1, s, e, e + 1\)
• “surrounding” POS tags \((s - 1\%e + 1)\)
• POS sequence from \(s\) to \(e\), and \(\{2, 3\}\)-grams
• \([-2,+2]\) token window. For each position: type, head
• left and right context to the next clause boundary: token-sequence and \(\{2, 3\}\)-grams, lexicalization, bag of words

(iii) On the relation between verb predicate and focus token:
• distance in tokens: \(\{<-2,-2,-1,0,+1,+2,>2\}\)
• path in tokens (and taking into account clause embedding), and \(\{2, 3, 4\}\)-grams

Total number of features: 270,413. Filtered by frequency. Reduced to 55,544 features.
Basic Models Tested

- **rswC.stumps**: recurrent sliding window model. The context considered are the three preceding argument labels assigned (correct labels during training, predicted labels when testing), e.g., “previous label is B-A0”. Weak rules are decision stumps. We trained up to 3,000 weak rules for each of the labels.

- **rswC.d3**: same model as rswC.stumps but now the weak rules are decision trees of a maximum depth of three. We trained up to 3,000 weak rules for each of the labels.
• **rswA.d3**: recurrent sliding window model. The context considered include the three preceding argument labels (as in the rswC.d3 model) and the “bag-of-labels” of the left context of the token into consideration (from the estart of the sequence), e.g., “B-A1 appears in the left context”. the labels.

• **sw.d3**: sliding window model. Independent tagging decisions. Weak rules are decision trees of a maximum depth of three. We trained up to 2,000 weak rules for each of the labels.
Overall prec/rec/F1 of \textbf{rswC.d3} on the development set with respect to the number of rounds
Experimental Results

Overall $F_1$ results of \texttt{rswC.stumps} and \texttt{rswC.d3} on the development set with respect to the number of rounds
Overall results of $rswC.d3$ and $rswC.stumps$ on the training set with respect to the number of rounds = training convergence
Overall F\textsubscript{1} results of sw.d3, rswC.d3, and rswA.d3 on the development set with respect to the number of rounds
Overall F$_1$ results of $rswC.d3$ and $rswA.d3$ systems assuming perfect context knowledge on the development set = upper bounds
# Comparative Results

<table>
<thead>
<tr>
<th>development</th>
<th>Precision</th>
<th>Recall</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>hacioglu</td>
<td>74.18%</td>
<td>69.43%</td>
<td>71.72</td>
</tr>
<tr>
<td><strong>rswC.d3</strong></td>
<td><strong>74.54%</strong></td>
<td><strong>68.06%</strong></td>
<td><strong>71.15</strong></td>
</tr>
<tr>
<td>punyakanok</td>
<td>71.96%</td>
<td>64.93%</td>
<td>68.26</td>
</tr>
<tr>
<td>carreras</td>
<td>73.40%</td>
<td>63.70%</td>
<td>68.21</td>
</tr>
<tr>
<td><strong>sw.d3</strong></td>
<td><strong>69.92%</strong></td>
<td><strong>65.05%</strong></td>
<td><strong>67.40</strong></td>
</tr>
<tr>
<td>lim</td>
<td>69.78%</td>
<td>62.57%</td>
<td>65.97</td>
</tr>
<tr>
<td>park</td>
<td>67.27%</td>
<td>64.36%</td>
<td>65.78</td>
</tr>
<tr>
<td><strong>rswC.stumps</strong></td>
<td><strong>67.96%</strong></td>
<td><strong>62.95%</strong></td>
<td><strong>65.36</strong></td>
</tr>
<tr>
<td>higgins</td>
<td>65.59%</td>
<td>60.16%</td>
<td>62.76</td>
</tr>
<tr>
<td>van den bosch</td>
<td>69.06%</td>
<td>57.84%</td>
<td>62.95</td>
</tr>
<tr>
<td>kouchnir</td>
<td>44.93%</td>
<td>63.12%</td>
<td>52.50</td>
</tr>
<tr>
<td>baldewein</td>
<td>64.90%</td>
<td>41.61%</td>
<td>50.71</td>
</tr>
<tr>
<td>williams</td>
<td>53.37%</td>
<td>32.43%</td>
<td>40.35</td>
</tr>
<tr>
<td>baseline</td>
<td>50.63%</td>
<td>30.30%</td>
<td>37.91</td>
</tr>
</tbody>
</table>
## Comparative Results

<table>
<thead>
<tr>
<th>Test</th>
<th>Precision</th>
<th>Recall</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>hacioglu</td>
<td>72.43%</td>
<td>66.77%</td>
<td>69.49</td>
</tr>
<tr>
<td>rswC.d3</td>
<td>73.33%</td>
<td>65.35%</td>
<td>69.11</td>
</tr>
<tr>
<td>punyakanok</td>
<td>70.07%</td>
<td>63.07%</td>
<td>66.39</td>
</tr>
<tr>
<td>sw.d3</td>
<td>69.71%</td>
<td>62.84%</td>
<td>66.09</td>
</tr>
<tr>
<td>carreras</td>
<td>71.81%</td>
<td>61.11%</td>
<td>66.03</td>
</tr>
<tr>
<td>lim</td>
<td>68.42%</td>
<td>61.47%</td>
<td>64.76</td>
</tr>
<tr>
<td>park</td>
<td>65.63%</td>
<td>62.43%</td>
<td>63.99</td>
</tr>
<tr>
<td>rswC.stumps</td>
<td>67.06%</td>
<td>60.94%</td>
<td>63.85</td>
</tr>
<tr>
<td>higgins</td>
<td>64.17%</td>
<td>57.52%</td>
<td>60.66</td>
</tr>
<tr>
<td>van den bosch</td>
<td>67.12%</td>
<td>54.46%</td>
<td>60.13</td>
</tr>
<tr>
<td>kouchpinr</td>
<td>56.86%</td>
<td>49.95%</td>
<td>53.18</td>
</tr>
<tr>
<td>baldewein</td>
<td>65.73%</td>
<td>42.60%</td>
<td>51.70</td>
</tr>
<tr>
<td>williams</td>
<td>58.08%</td>
<td>34.75%</td>
<td>43.48</td>
</tr>
<tr>
<td>baseline</td>
<td>54.60%</td>
<td>31.39%</td>
<td>39.87</td>
</tr>
</tbody>
</table>
Open Problems

- Determine the role of the segmentation and features on the final performance
- Proof that a “complex” learning architecture can perform better than the simple RSW approach. A fair comparison is needed.
- Exploit interactions between arguments of different verbs when learning and tagging
- Better exploit semantic information of the problem
Thank you very much for your attention!