

CoNLL-2005 shared task spotlight: Partial vs full parsing in SRL

Lluís Màrquez

TALP Research Center, Software Department
Universitat Politècnica de Catalunya

Ann Arbor, June 30, 2005

Motivation

- One of the conclusions from the 2004 edition:
 - ★ State-of-the art SRL systems based on full parsing perform close to 80 (F_1), while systems based on partial parsing perform close to 70.
- Also, results from the 2005 edition are ~ 10 points better than those of 2004
- How much of this difference is due to the transition from partial to full parsing?

Setting₁

- The **marquez** system at CoNLL-2005 presents a comparison between two single SRL systems:
 - ★ **PP**: based on partial parsing (chunks + clauses)
 - ★ **FP_{CHA}**: based on full parsing (Charniak)
 - ★ Fixed setting for both systems
 - ★ The only difference is input information
 - ★ SRL is approached as a B-I-O sequential tagging task

Setting₂

- The **marquez** system at CoNLL-2005 presents a comparison between two single SRL systems:
 - ★ Sequentialization performed by selecting the top-most syntactic elements in the sentence spans defined by clause boundaries
 - ★ Rich set of features based on state-of-the-art
 - ★ Most features common to PP and FP
 - ★ But some complex features make sense only for FP
 - ★ AdaBoost was used to train independent local classifiers for B, I, and O labels

Results₁

- Overall results of the systems on the development set

	Perfect props	Precision	Recall	$F_{\beta=1}$
PP	47.38%	76.86%	70.55%	73.57
FP _{CHA}	51.51%	78.08%	73.54%	75.75

- FP_{CHA} results are better but not that much. Overall PP results are “only” 2.2 points below the best single system on the 2005 task
- FP results are structurally better (% of perfect props)

Results₂

- A more detailed analysis reveals that the two systems predict better different arguments

	FP _{CHA}	PP
A0	84.03	79.79
A1	77.88	74.78
A2	62.36	65.10
A3	59.02	60.22
A4	67.86	67.27

- FP performs better in A0–A1 arguments. PP performs better in A2–A4 and some AM arguments.
- This is good for system combination

Results₃

- Some evidence: overall results of a combined system on the development set (a very naïve combination scheme was used)

	Perfect props	Precision	Recall	$F_{\beta=1}$
PP	47.38%	76.86%	70.55%	73.57
FP _{CHA}	51.51%	78.08%	73.54%	75.75
Combin.	51.39%	78.39%	75.53%	76.93

- Recall improves 2%

Other observations

- FP generates shorter token sequences (21% less training examples than PP). Thus, FP trains faster using less memory
- PP was expected to be more robust on the Brown corpus...
 - ★ But, actually it is worse than FP :-(
★ Explanation: the clause splitter module degrades more than the full parser