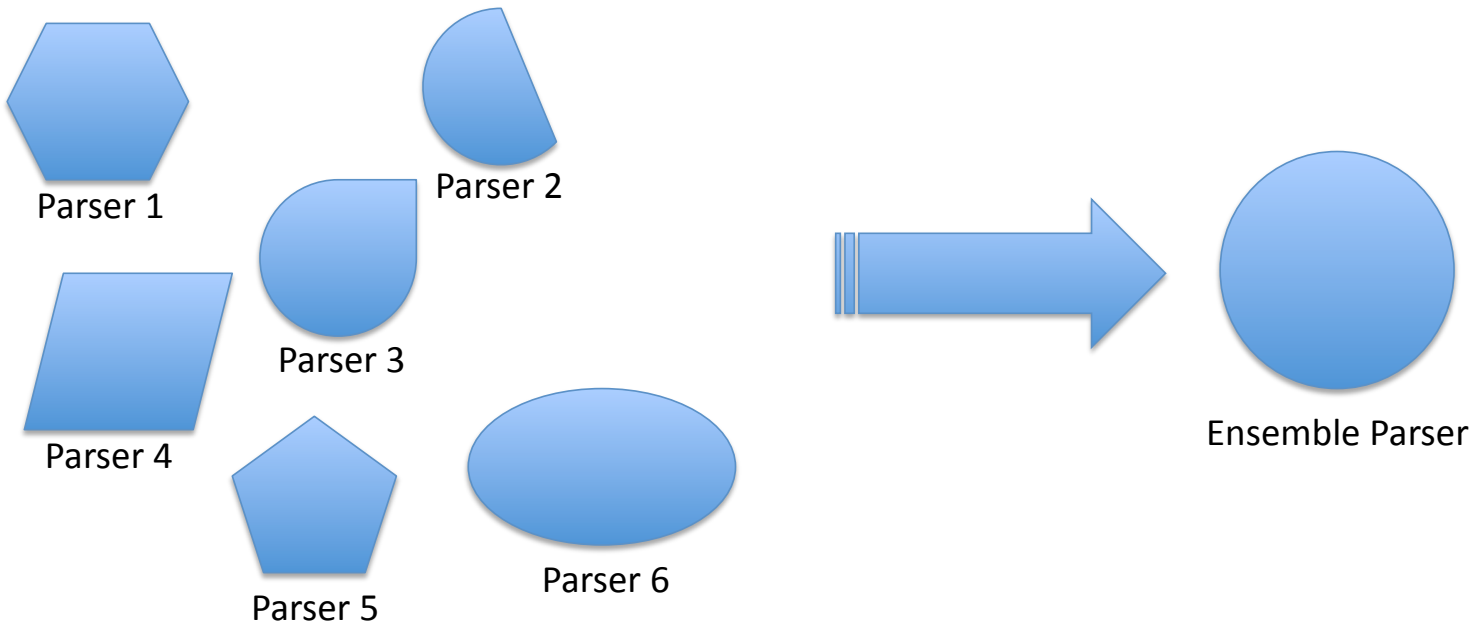
Ensemble Models for Dependency Parsing: Cheap and Good?

Mihai Surdeanu and Christopher D. Manning

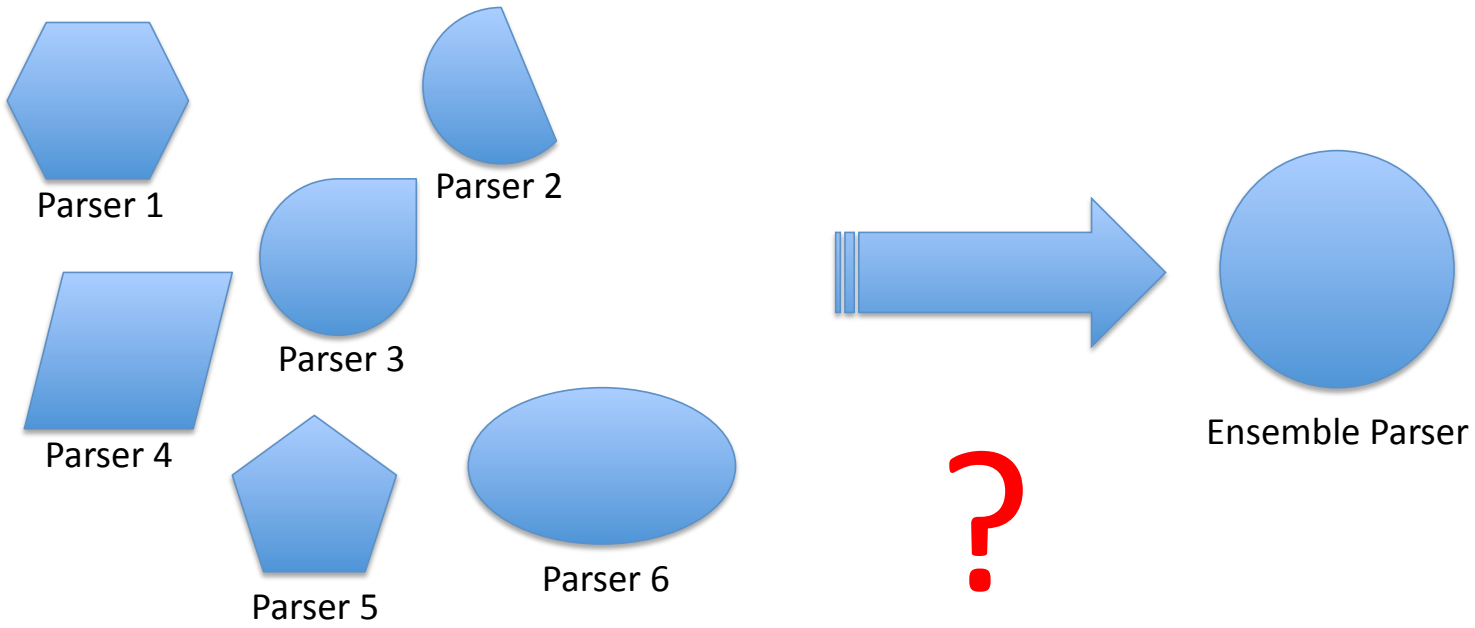
Stanford University

June 3, 2010

Ensemble Parsing



Ensemble Parsing



Many questions still unanswered despite all the previous work

This work: empirical answers for projective English dependency parsing

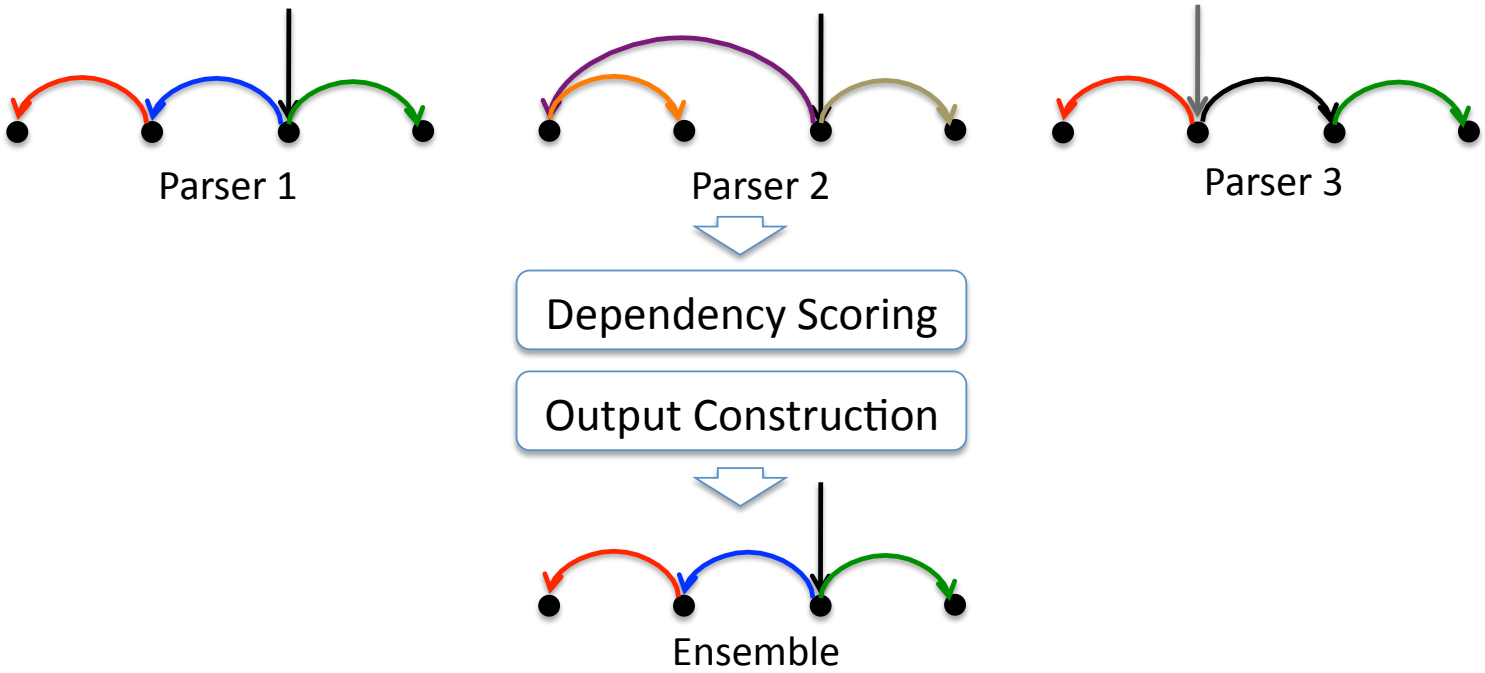
Setup

Corpus: syntactic dependencies of the CoNLL 2008-09 shared tasks

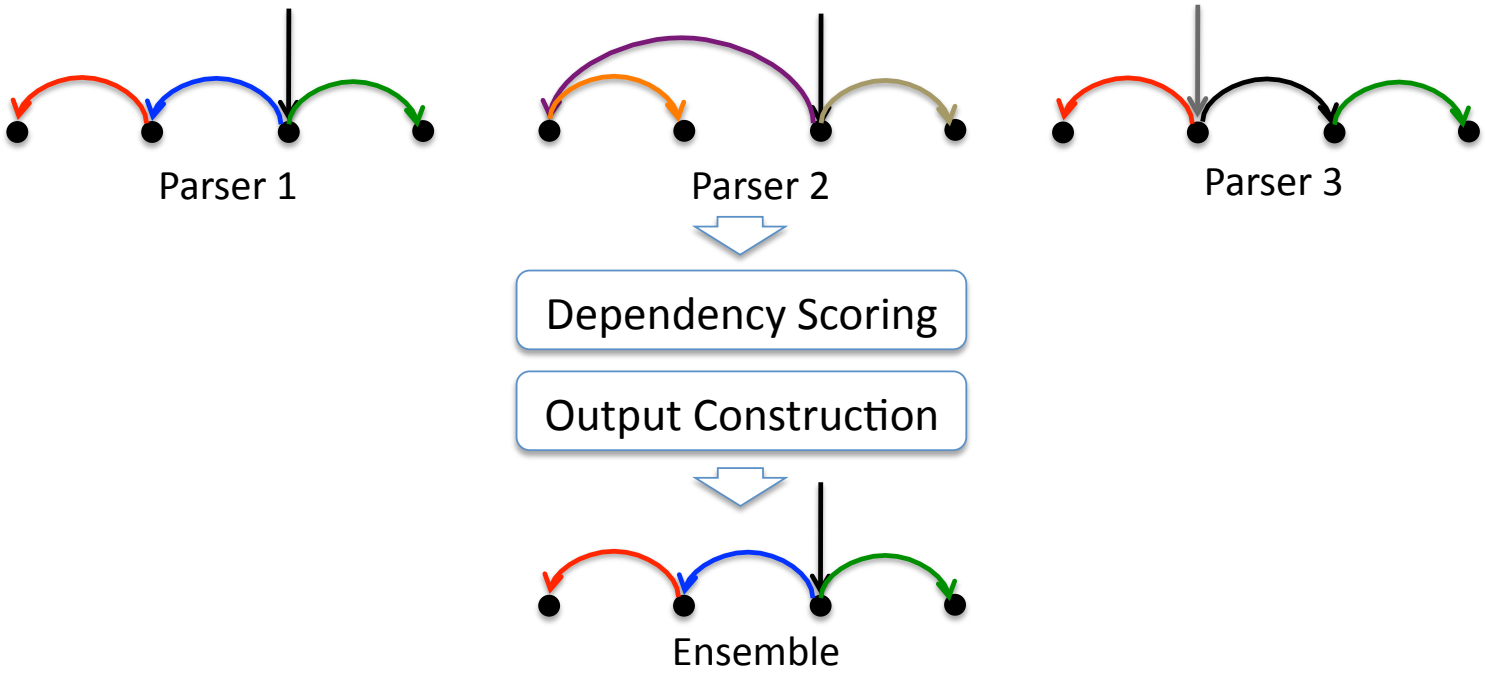
7 individual parsing models:

	Devel	In domain	Out of domain
	LAS	LAS	LAS
MST	85.36	87.07	80.48
$\text{Malt}^{\rightarrow}_{AE}$	84.24	85.96	78.74
$\text{Malt}^{\rightarrow}_{CN}$	83.75	85.61	78.55
$\text{Malt}^{\rightarrow}_{AS}$	83.74	85.36	77.23
$\text{Malt}^{\leftarrow}_{AS}$	82.43	83.90	76.69
$\text{Malt}^{\leftarrow}_{CN}$	81.75	83.53	77.29
$\text{Malt}^{\leftarrow}_{AE}$	80.76	82.51	76.18

Scoring Models for Parser Combination



Scoring Models for Parser Combination



Which scoring model is best?

- → Unweighted voting?
- → Weighted voting? Weighted by what?
- → Meta-classification?

Scoring Models: Voting

	Unweighted	Weighted by POS of modifier	Weighted by label of dep.	Weighted by dep. length	...
	LAS	LAS	LAS	LAS	
3	86.03	86.02	85.53	85.85	
4	86.79	86.68	86.38	86.46	
5	86.98	86.95	86.60	86.87	
6	87.14	**87.17**	86.74	86.91	
7	86.81	86.82	86.50	86.71	

Weighting does not really make a difference!

More individual parsers helps, but up to a point.

Scoring Models: Meta-classification

- Can we improve dependency scoring through meta-classification?

Scoring Models: Meta-classification

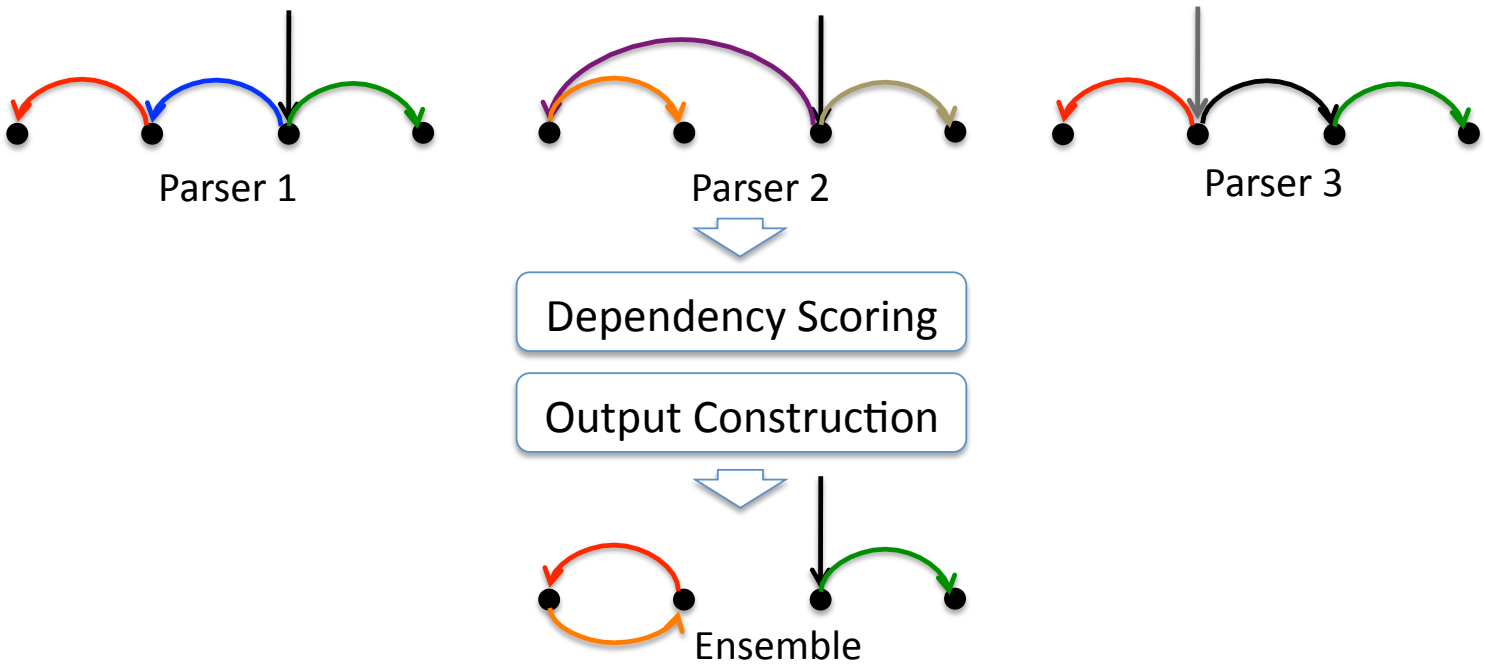
- Can we improve dependency scoring through meta-classification?
- No.
  - → We implemented a L2-regularized logistic regression classifier using as features: identifiers of the base models, POS tags of head and modifier, labels of dependencies, length of dependencies, length of sentence, and combinations of the above.
  - → No improvement over the unweighted voting approach.

Meta-classification Analysis

- Minority dependencies (MD): dependencies that disagree with the majority vote.
- Precision of MDs: ratio of MDs in a given context (e.g., POS of modifier is NN and parser is MST) that are correct.
- Meta-classification can outperform majority vote only when the number of MDs in contexts with precision $> 50\%$ is large.

  $\rightarrow$But these are less than 0.7% of total dependencies!

Re-parsing Algorithms



How common are badly-formed trees for word-by-word combination?

Which is the best re-parsing strategy?

Re-parsing Algorithms

	In domain	Out of domain
Zero roots	0.83%	0.70%
Multiple roots	3.37%	6.11%
Cycles	4.29%	4.23%
Total	7.46%	9.64%

Percentage of badly-formed trees for word-by-word combination

Re-parsing Algorithms

	In domain	Out of domain
Zero roots	0.83%	0.70%
Multiple roots	3.37%	6.11%
Cycles	4.29%	4.23%
Total	7.46%	9.64%

Percentage of badly-formed trees for word-by-word combination

	In domain	Out of domain
	LAS	LAS
Word by word ($O(N)$)	88.89	82.13*
Eisner (exact – $O(N^3)$)	88.83*	81.99
Attardi (approximate – $O(N)$)	88.70	81.82

Performance of re-parsing algorithms

Badly-formed trees are common! But approximate re-parsing algorithms perform as well as exact ones!

* indicates statistical significance over the next lower ranked model

Combination Strategies

How important is it to combine parsers at *learning* time?

$\rightarrow$ E.g., stacking: $\text{MST}_{\textit{Malt}} = \text{MST} + \text{Malt features}$

Combination Strategies

How important is it to combine parsers at *learning* time?

$\rightarrow$ E.g., stacking: MST_{Malt} = MST + Malt features

	In domain	Out of domain
	LAS	LAS
$\text{ensemble}^{3}_{100\%}$	88.83*	81.99*
$\text{ensemble}^{1}_{100\%}$	88.01*	80.78
$\text{ensemble}^{3}_{50\%}$	87.45	81.12
MST_{Malt}	87.45*	80.25*
$\text{ensemble}^{1}_{50\%}$	86.74	79.44

The advantages gained from combining parsers at learning time can be easily surpassed by runtime combination models that have access to more base parsers!

The ensemble models are more robust out of domain

Comparison with State of the Art Parsers

	In domain	Out of domain
	LAS	LAS
CoNLL 2008 #1 (Johansson and Nugues)	90.13*	82.81*
$\text{ensemble}^{3}_{100\%}$	88.83*	81.99*
CoNLL 2008 #2 (Zhang et al.)	88.14	80.80
$\text{ensemble}^{1}_{100\%}$	88.01	80.78

Our best ensemble model is second

In the out-of-domain corpus, performance is within 1% LAS of a parser that uses second-order features and is $O(N^4)$

The ensemble models are more robust out of domain

Conclusion: Less Is More

- The diversity of base parsers is more important than complex learning models for parser combination (e.g., meta-classification, stacking)
- Well-formed dependency trees can be guaranteed without significant performance loss by linear-time approximate re-parsing algorithms
- Unweighted voting performs as well as weighted voting for the re-parsing of candidate dependencies
- Ensemble parsers that are both accurate and fast can be rapidly developed with minimal effort

Thank you!

- Many thanks to Johan Hall, Joakim Nivre, Ryan McDonald, and Giuseppe Attardi
- Code: `www.surdeanu.name/mihai/ensemble/`
- Questions?