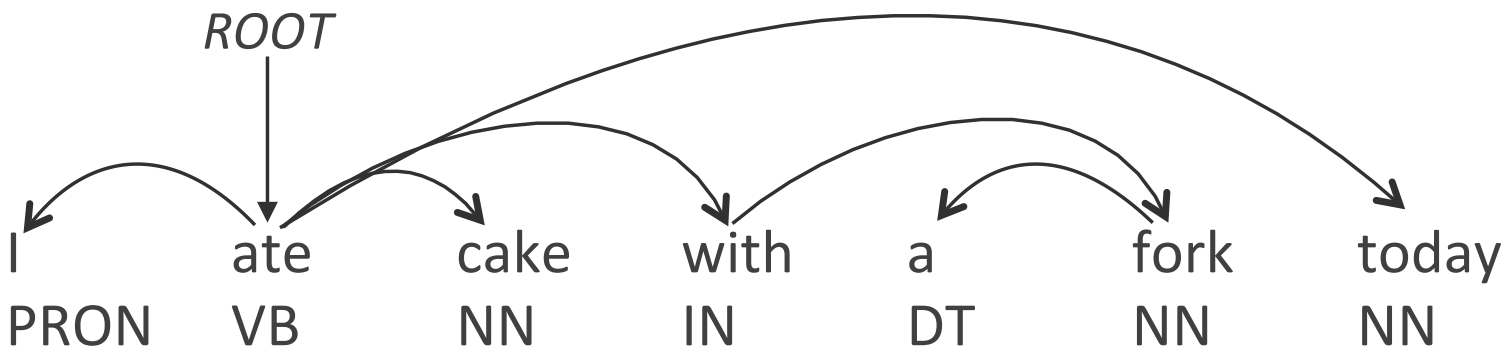
Progress on Dependency Parsing

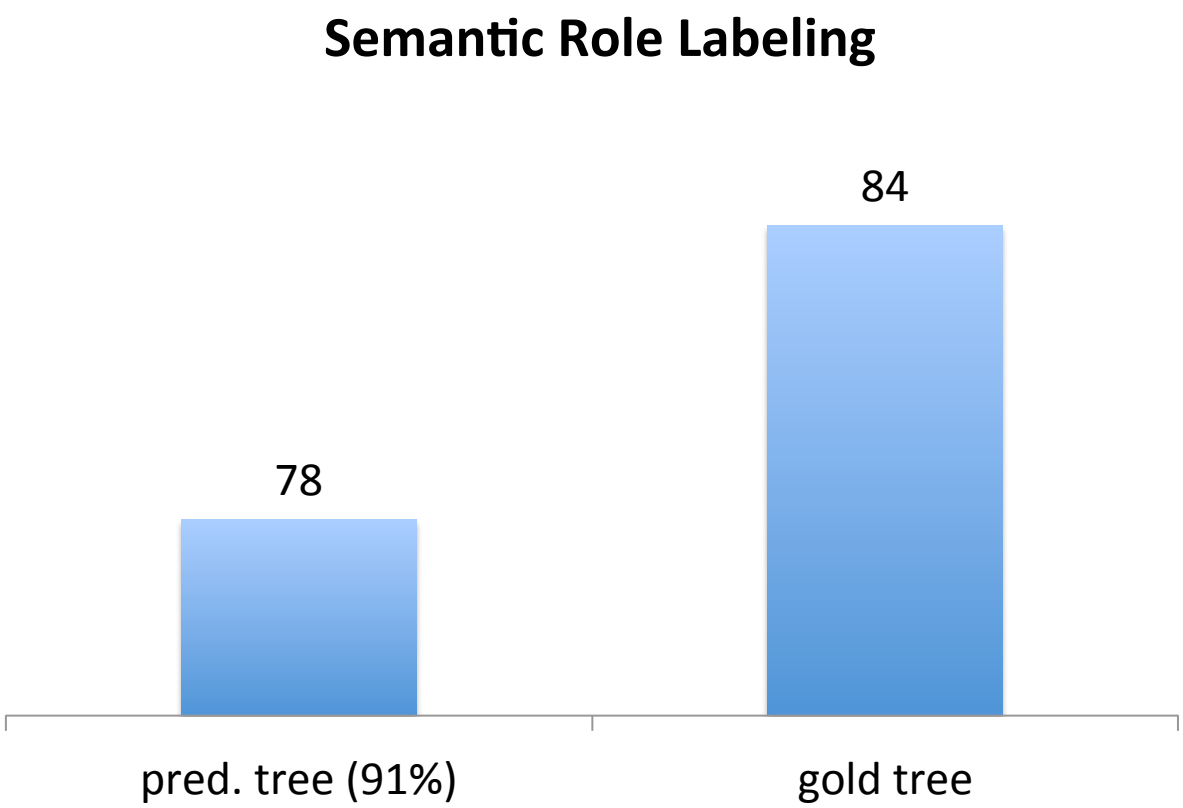
Regina Barzilay
CSAIL, MIT

Joint work with Tao Lei,
Yuan Zhang, and Tommi Jaakkola

Dependency Parsing



Why Dependency Parsing?

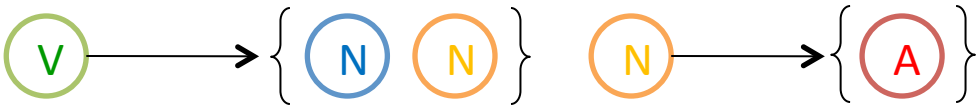


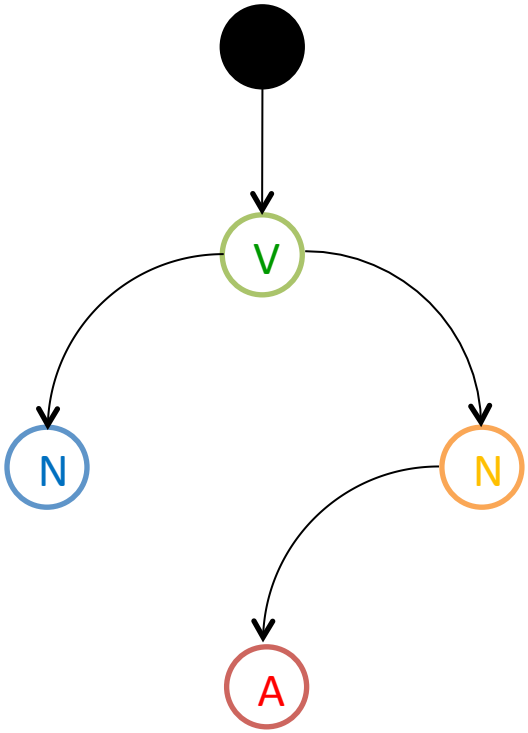
Syntactic parsing accuracy impacts applications

Progress

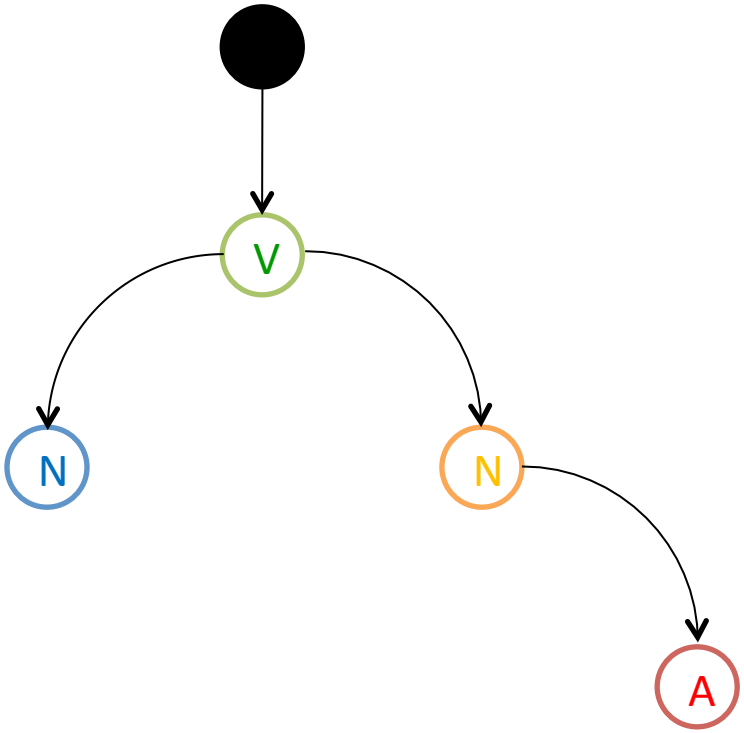
			Malagasy	*Kinyawanda*
2012	Language Universals	(MIT)	*54.5*	*63.3*
2013	Turbo Parser *3rd order, dual decomposition*	(CMU)	*72.4*	*70.1*
2014	Tensor	(MIT)	*74.4*	*70.2*
2014	Tensor + Greedy*	(MIT)	*75.1*	*79.2*

Dependent "*Selection*" is universal

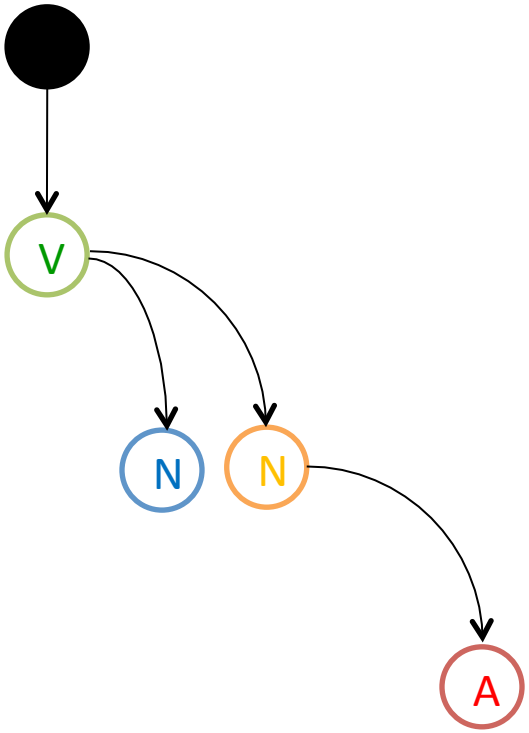




English & Dutch

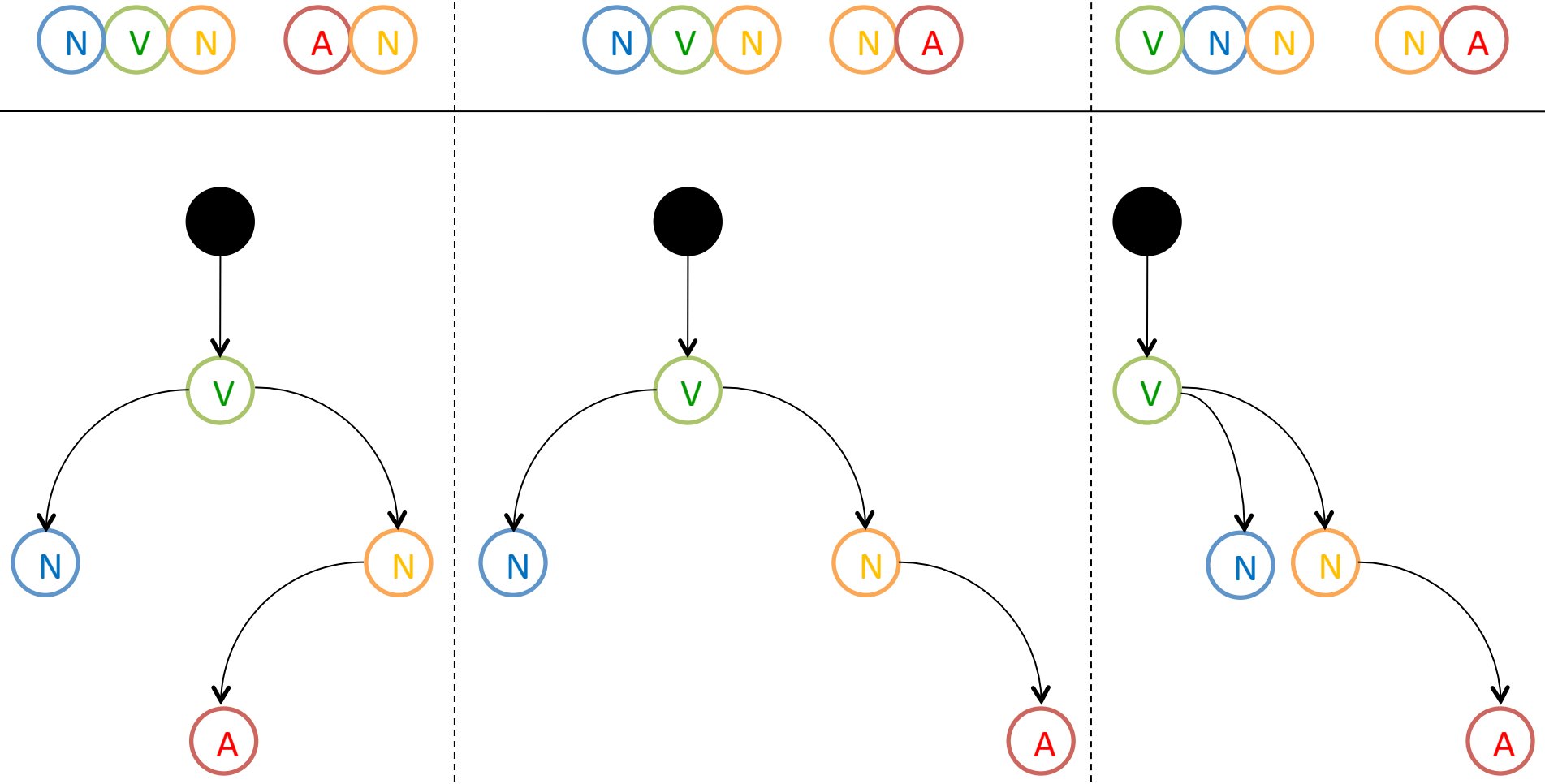


Portuguese

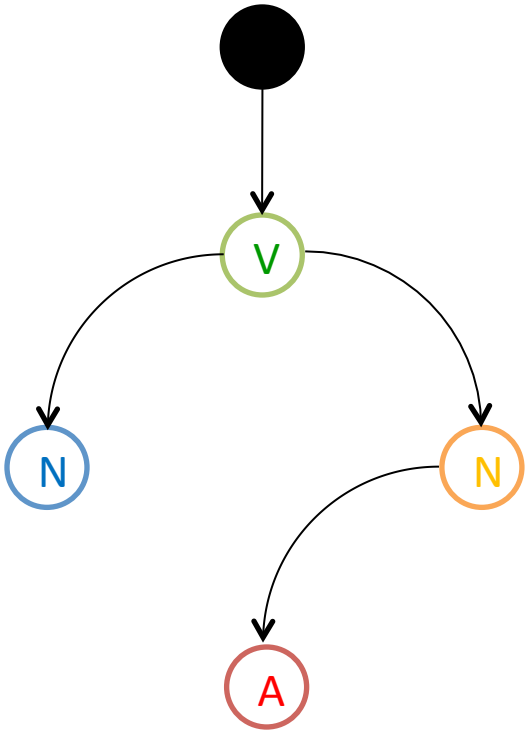


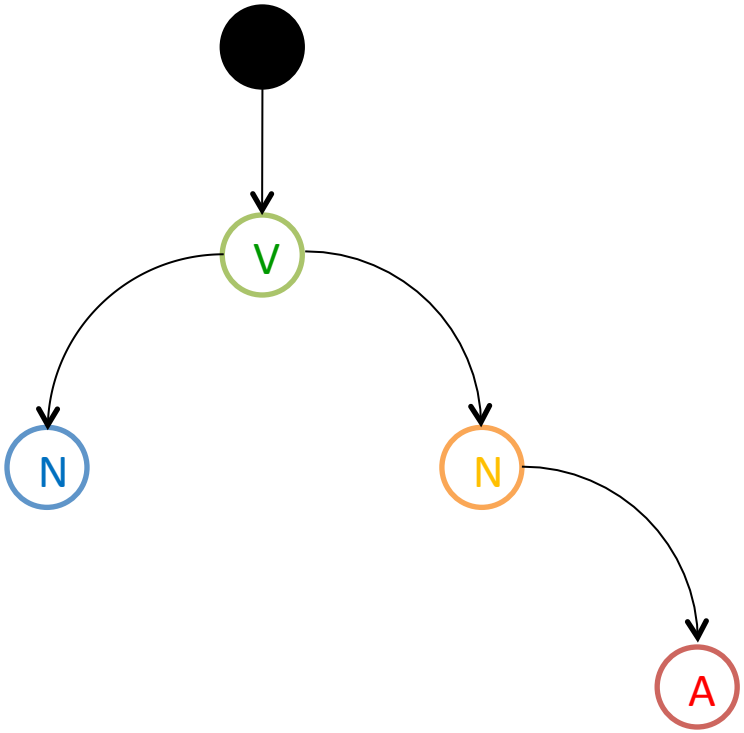
Arabic

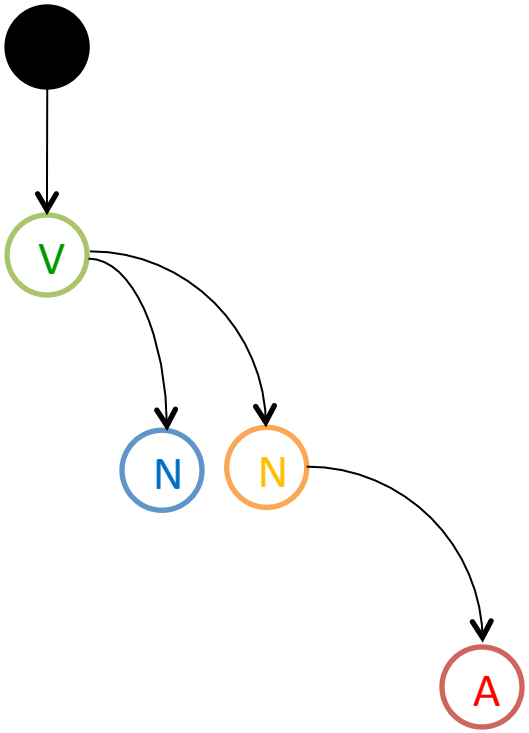
"Ordering" is language specific



"Ordering" typological features can guide





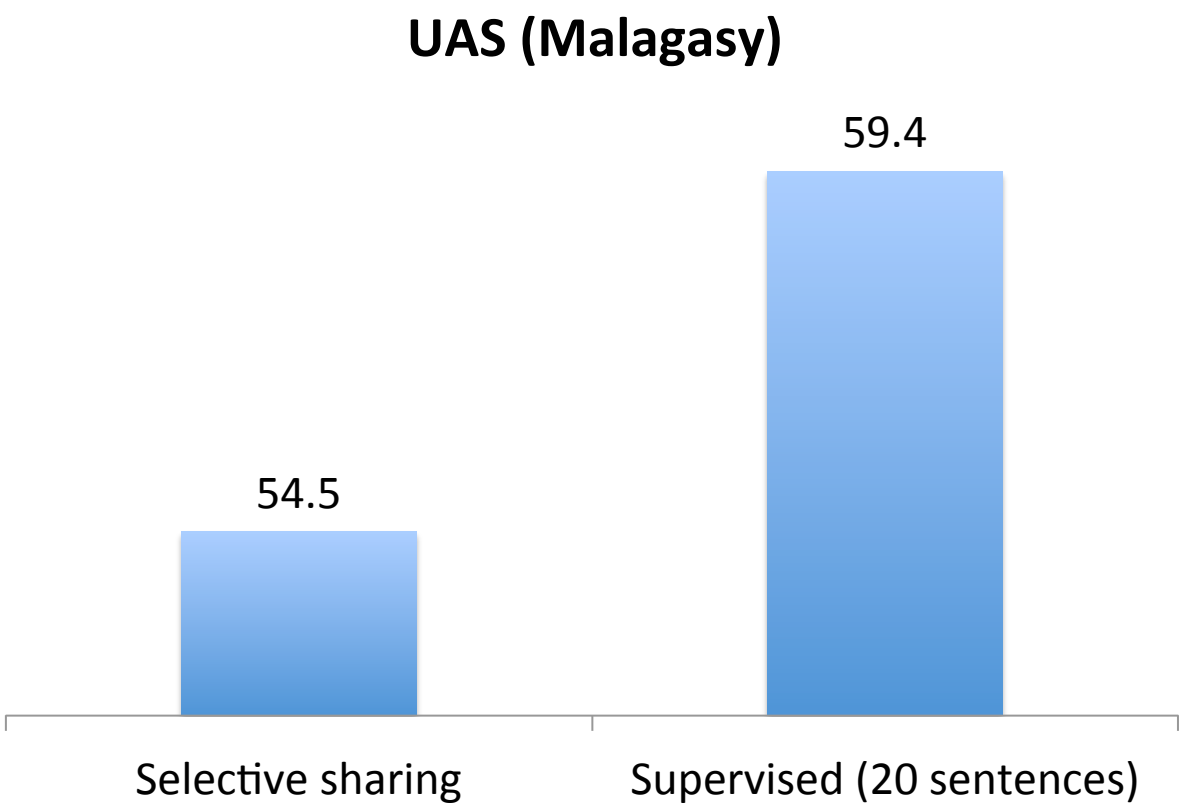


English

Portuguese

Arabic

Selective Sharing vs. Monolingual

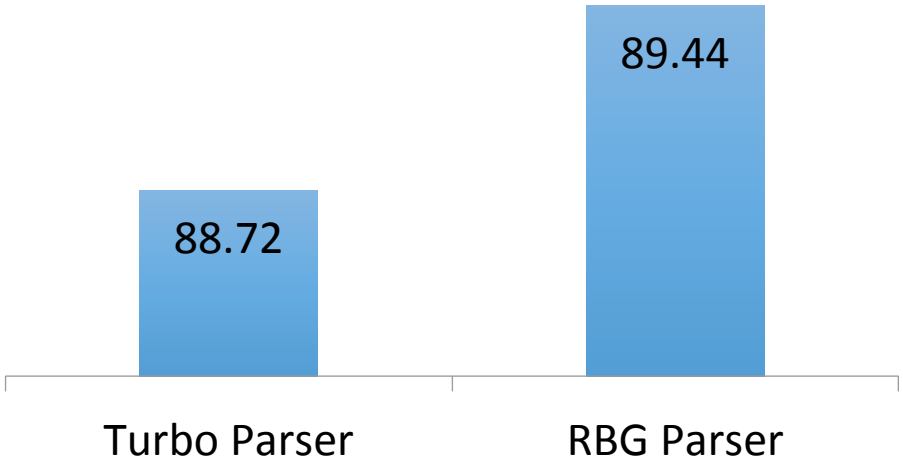


Advances in Monolingual Parsing

New ideas:

- Representations by low-rank tensor factorization
- Inference by randomized greedy hill-climbing

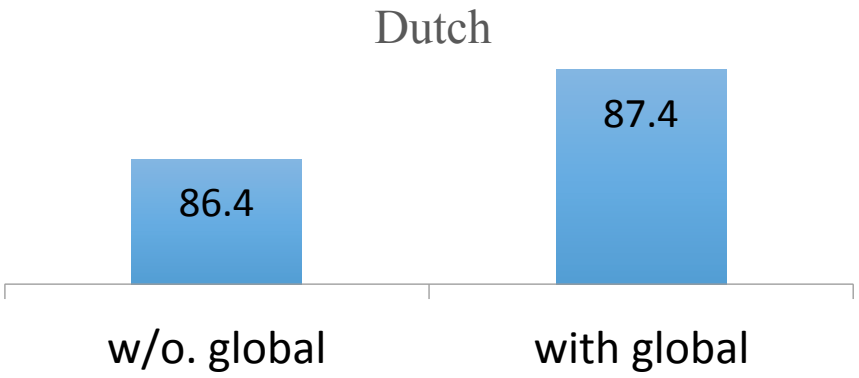
Top performance on CoNLL datasets across 14 languages



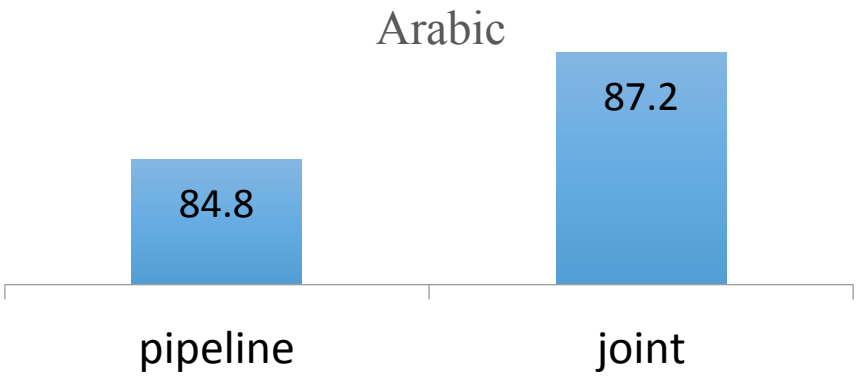
Best student paper award in ACL 2014

Randomized Greedy: Desiderata

- Handles any type of features



- Enables joint morphology, tagging and parsing



Real Questions behind this Research

- High-order parsing is NP-hard (McDonald et al., 2006)
- Dual decomposition finds certificates > 90% of the cases
- Hypothesis: parsing is easy on average
- Many NP-hard problems are easy on average
  - MAX-SAT (Resende et al., 1997)
  - Set cover (Hochbaum, 1982)

Therefore we provide:

- analysis on average parsing complexity
- a simple inference algorithm based on the analysis

Core Idea

- Climb to the optimal tree in a few small greedy steps

Randomized Hill-climbing

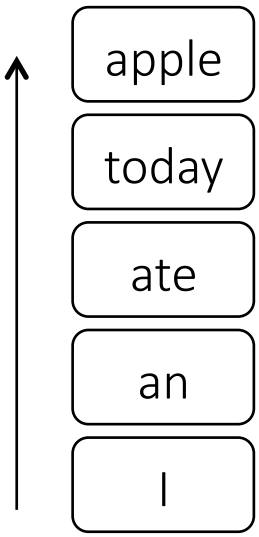
For k = 1 to K

1) Randomly sample a dependency tree
2) Greedily improve the tree one edge at a time
3) Repeat (2) until converge

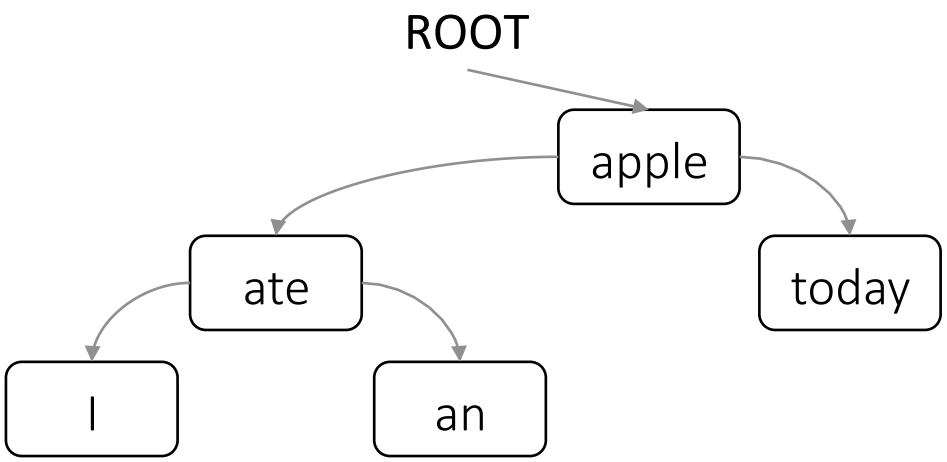
Select the tree with highest score

That's it!

Example

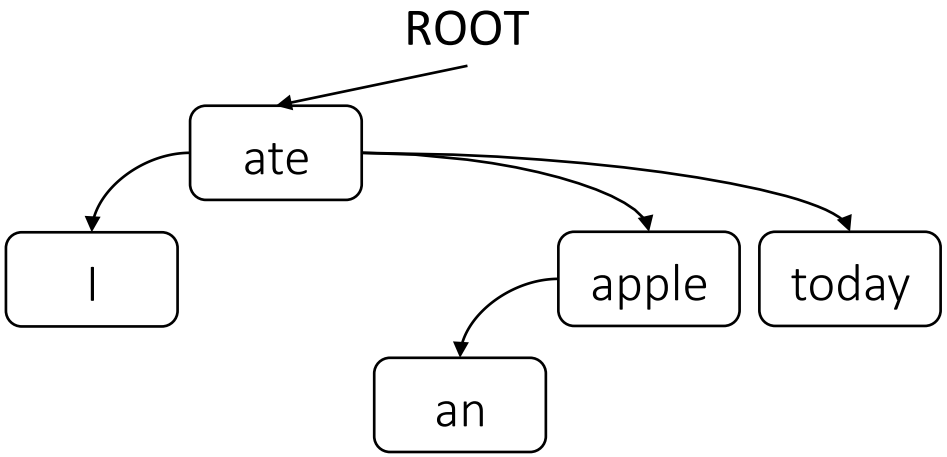


Initial tree

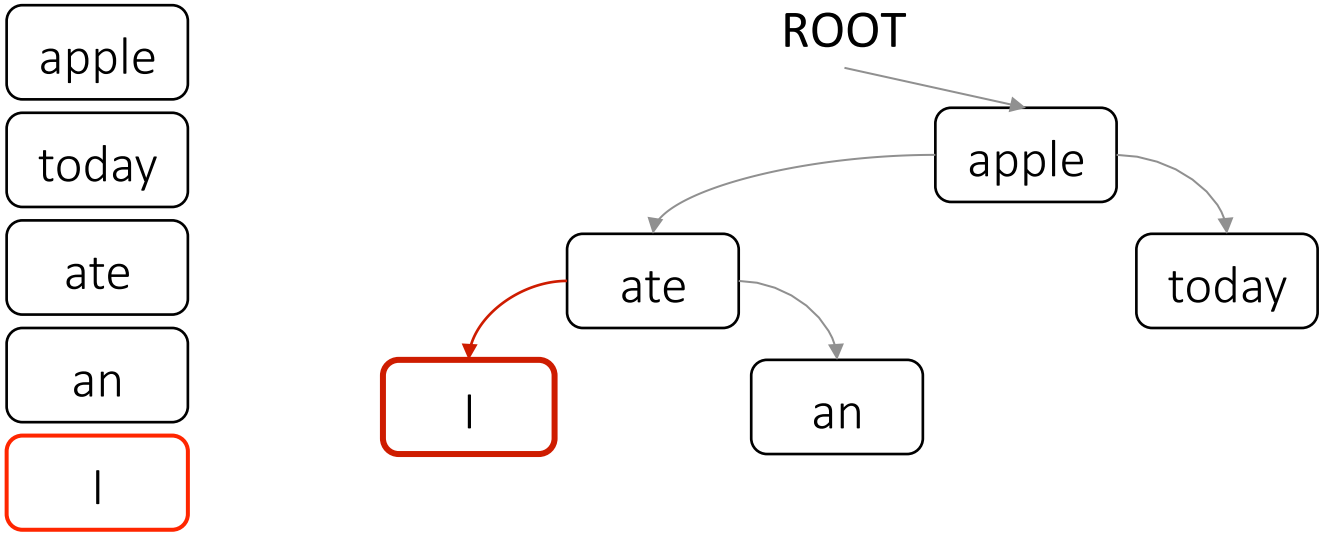


" I ate an apple today"

Target tree

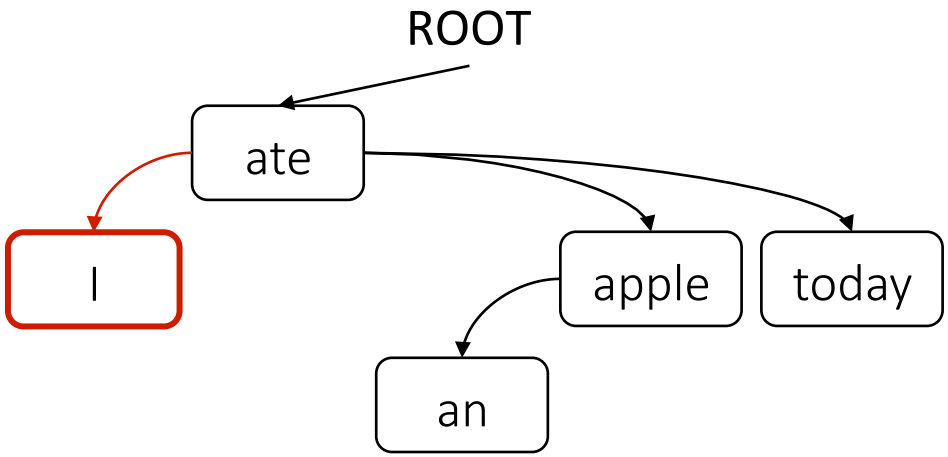


Example

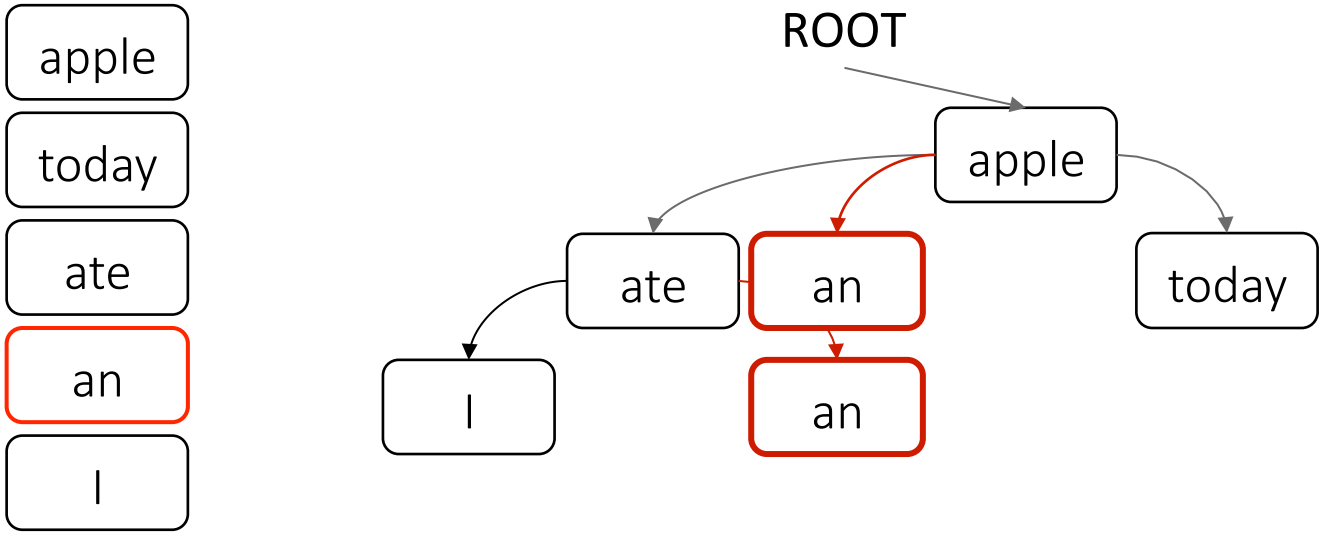


" I ate an apple today"

Target tree

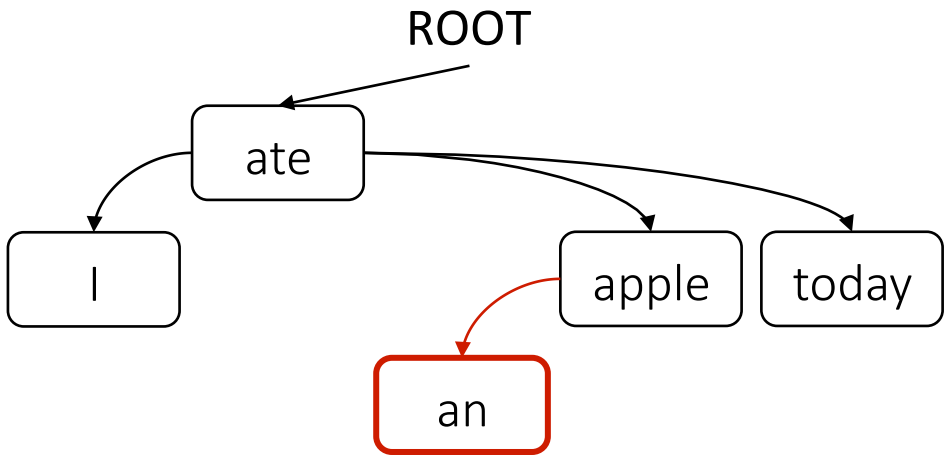


Example

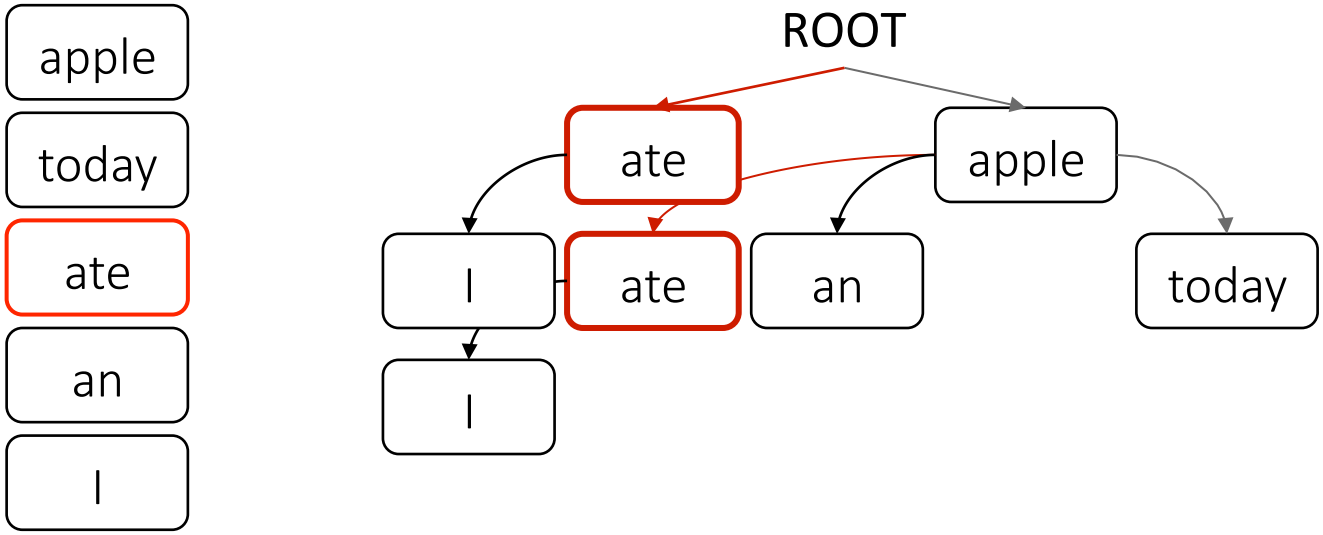


" I ate an apple today"

Target tree

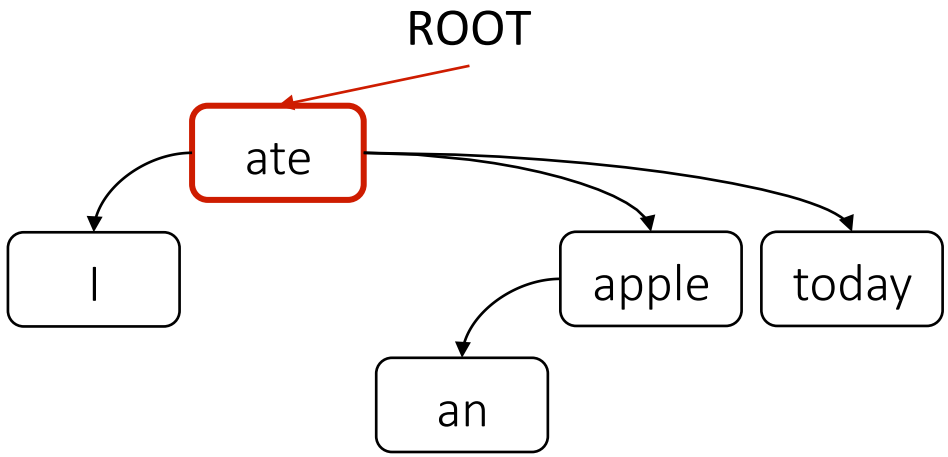


Example

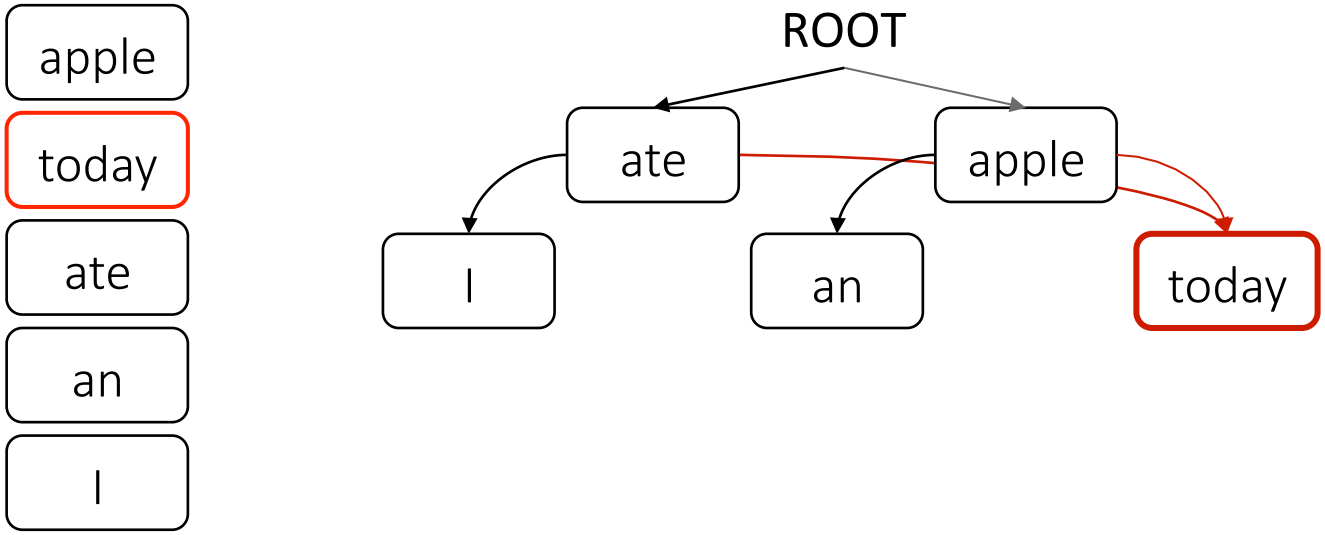


" I ate an apple today"

Target tree

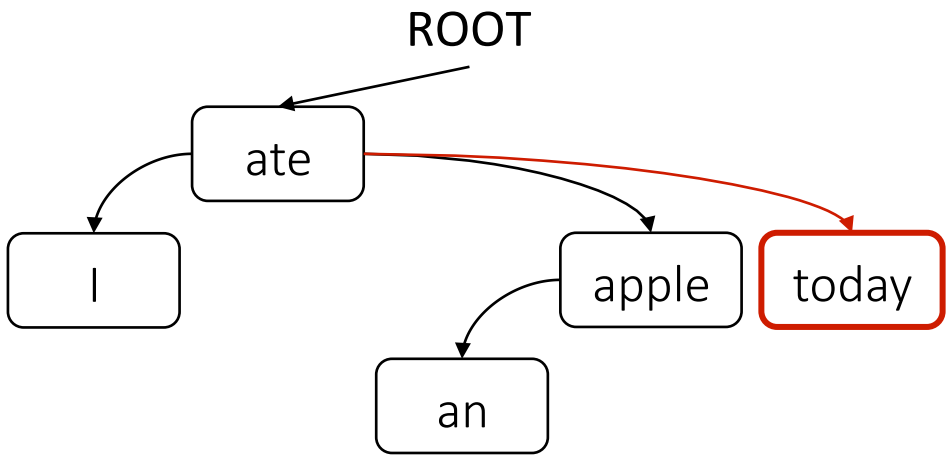


Example

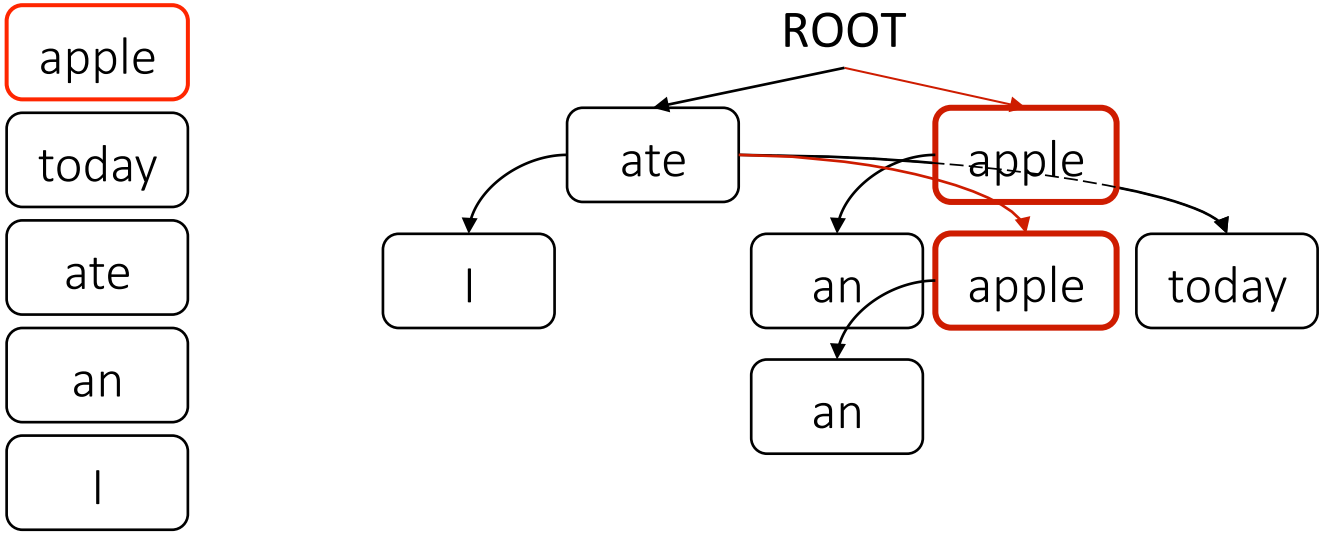


“ I ate an apple today”

Target tree

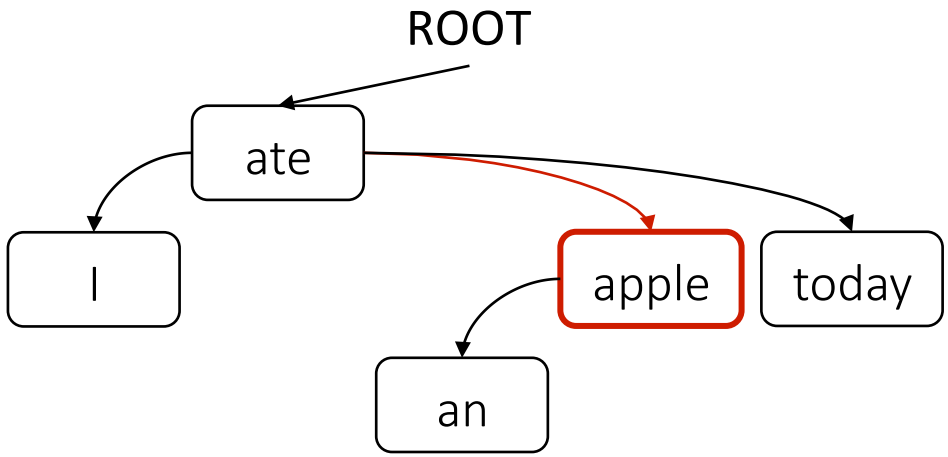


Example

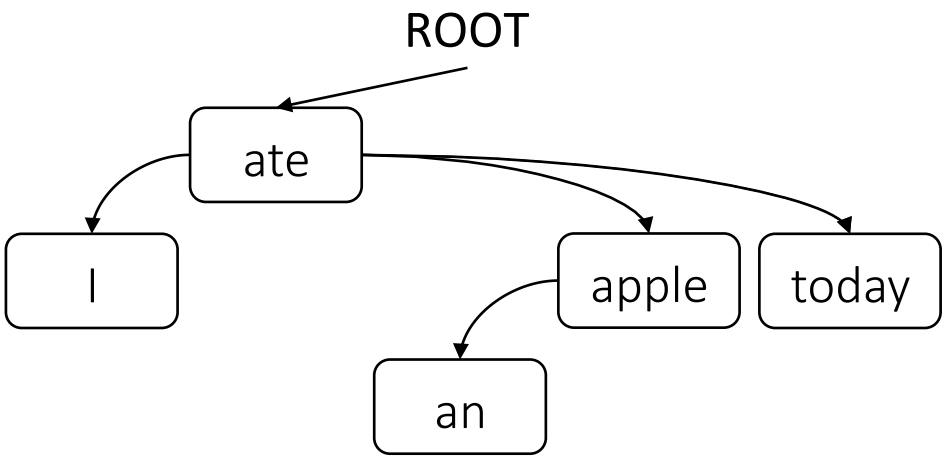


“ I ate an apple today”

Target tree

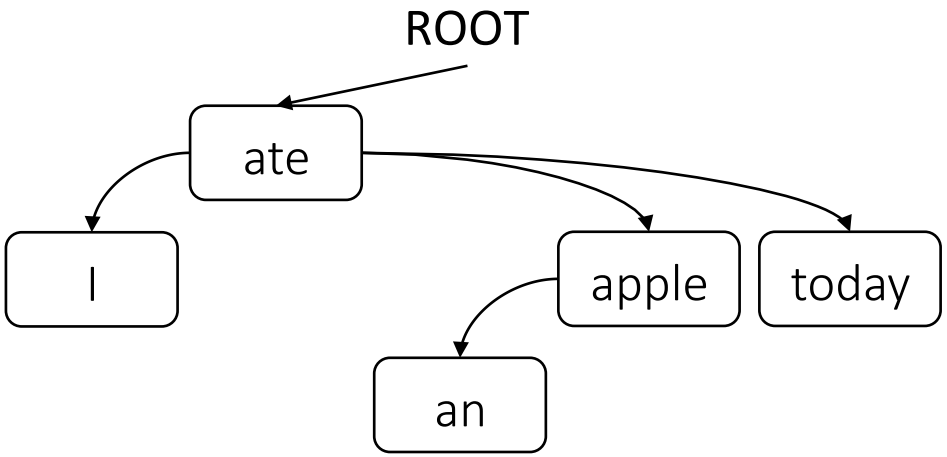


Example

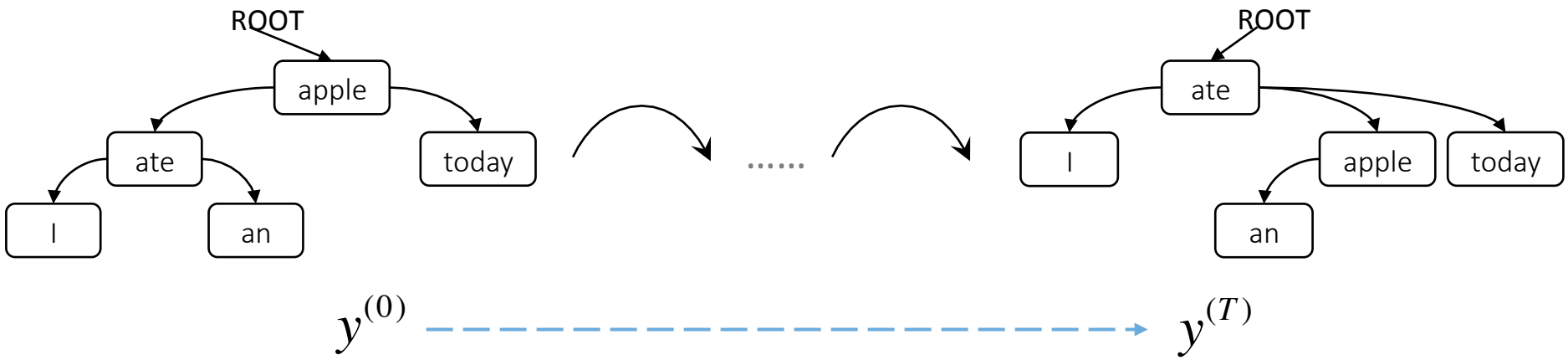


“ I ate an apple today”

Target tree



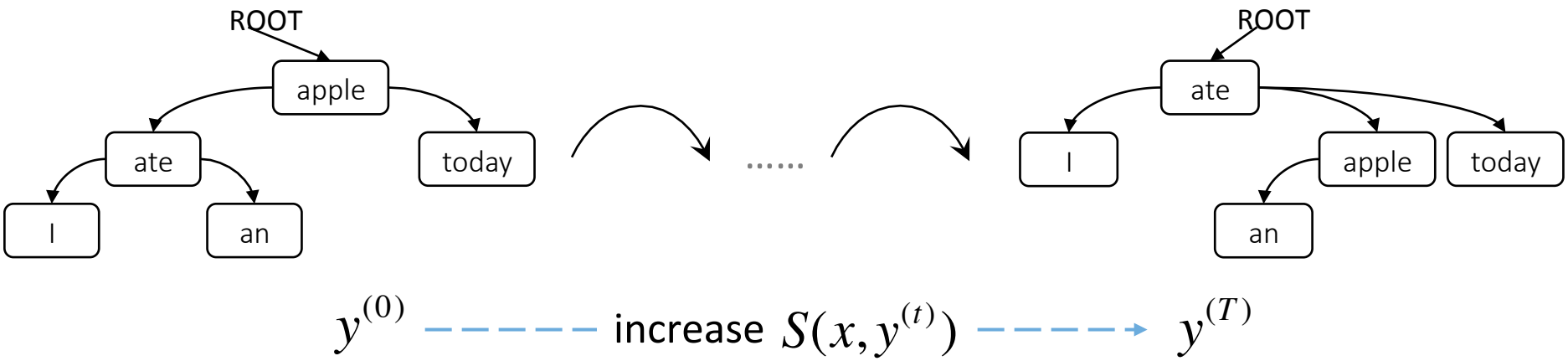
Why Greedy Has a Chance to Work



Reachability: transforming any tree to any other tree

- maintaining the structure a valid tree at any point
- using as few as *d* steps (*d* : head differences/hamming distance)

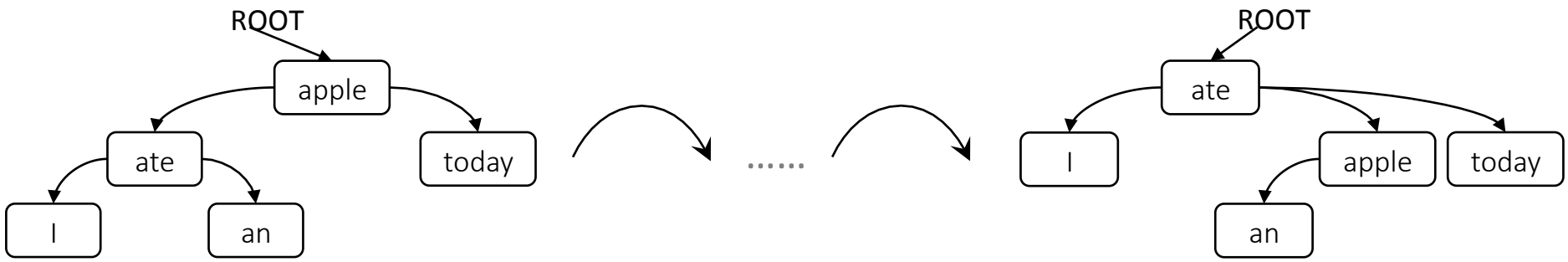
Greedy Hill-climbing

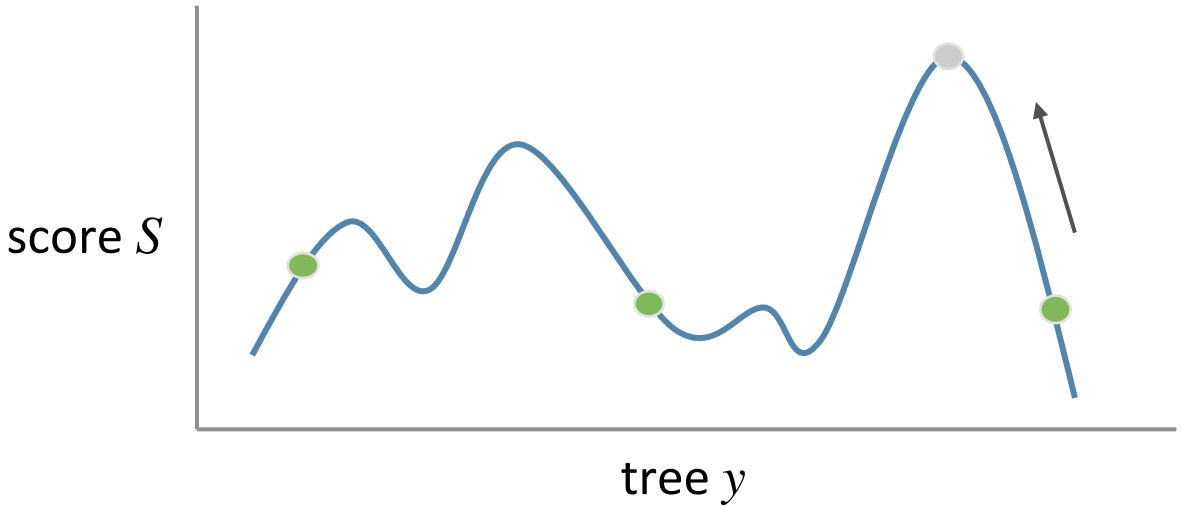


$y^{(0)}$ ------ increase $S(x, y^{(t)})$ ------→ $y^{(T)}$

Arbitrary features in the scoring function

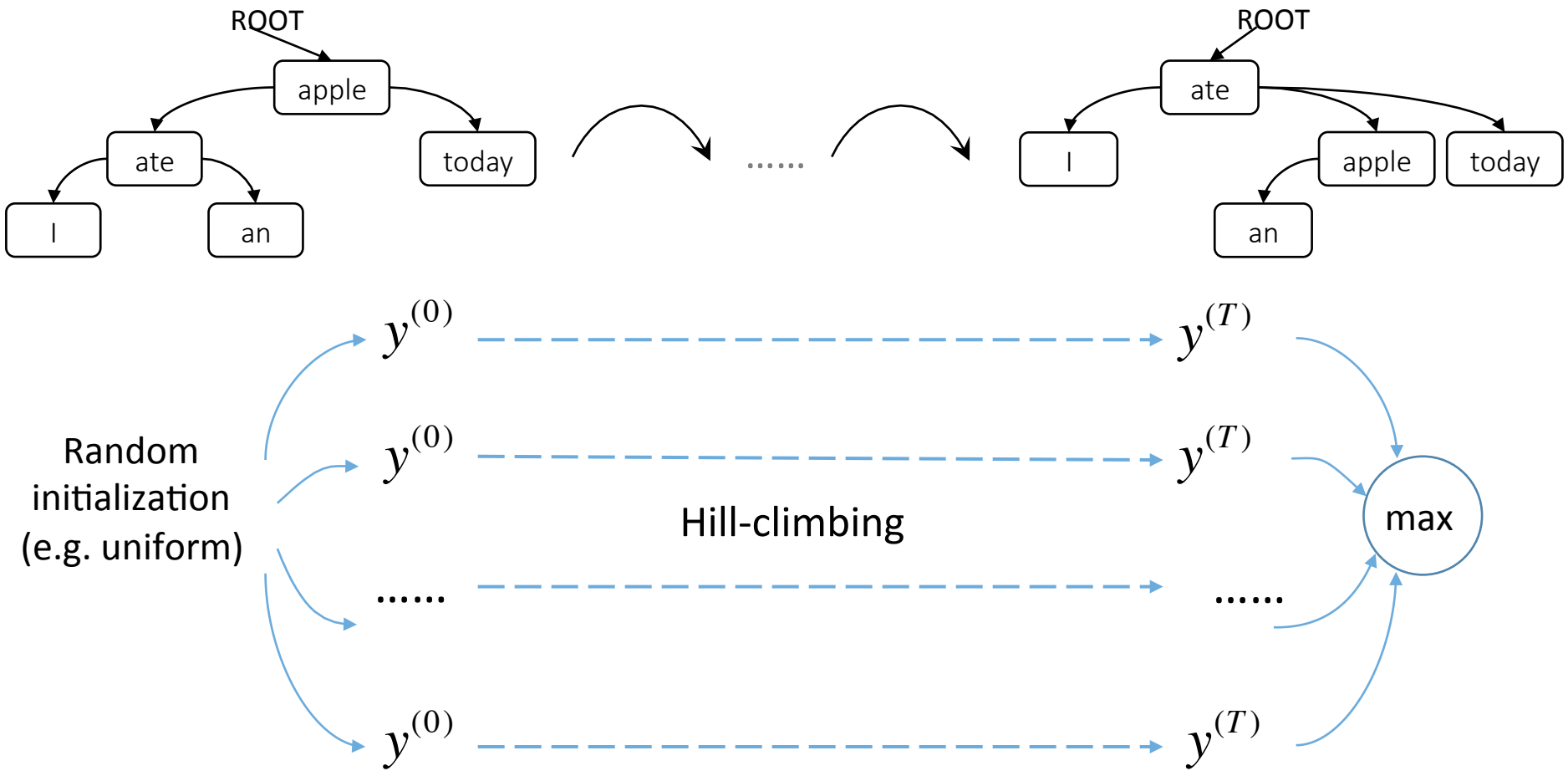
Hill-climbing with Restarts





Overcome local optima via restarts

Hill-climbing with Restarts



Overcome local optima via restarts

Learning Algorithm

- Follow common max-margin framework

$$\forall y \in T(x) \qquad S(x,\hat{y}) \geq S(x,y) + \mathrm{l}\,\hat{y} - y\,\mathrm{l} - \xi$$

  - $\hat{y}$ is the gold tree

- Adopt passive-aggressive online learning framework (Crammer et al. 2006)

- Decode with our randomized greedy algorithm

Analysis

	Theoretical	Empirical
First-order	✓	✓
High-order	?	✓

Search Space Complexity: First-order

≈ 2 billion trees

10 words

< 512 local optima

Search Space Complexity: First-order

Theorem: For any first-order scoring function:

- there are at most 2^{n-1} locally optimal trees
- this upper bound is **tight**

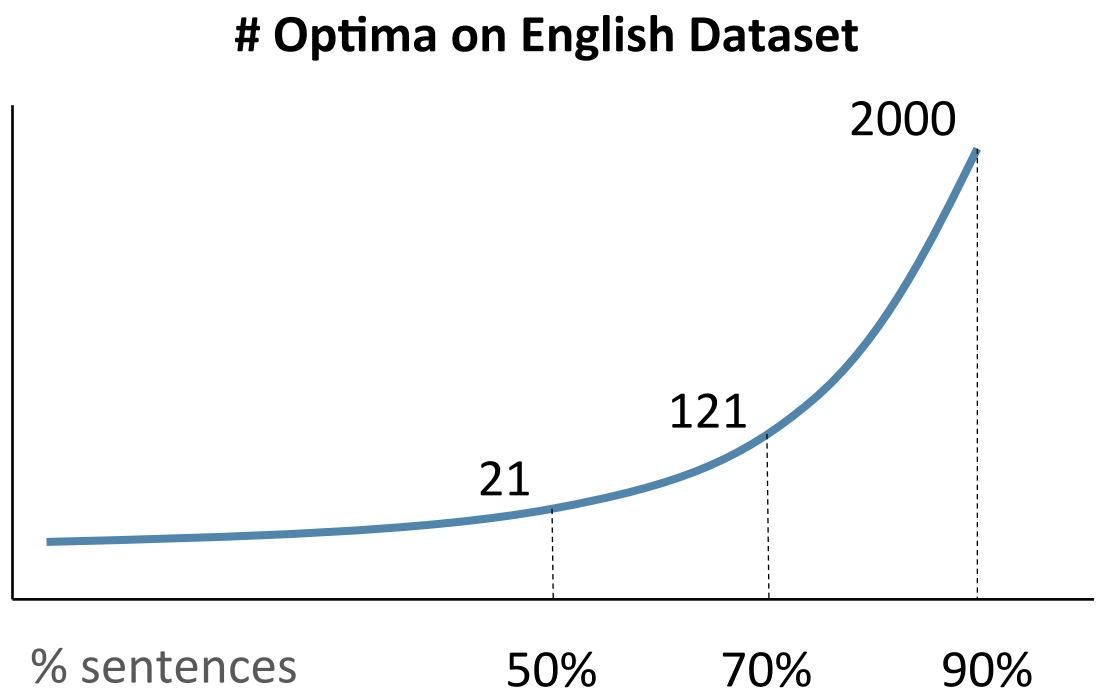
2^{n-1} is still a lot, but it is the worst case

Average Case Analysis:

- New algorithm for counting local optima
- Based on Chu-Liu-Edmonds algorithm

Empirical Results: First-order

How many local optima in real data?



Empirical Results: First-order

Does the hill-climbing find the argmax?

Finding Global Optimum on English

Len. ≤ 15	Len. > 15
100%	99.3%

Easy search space leads to successful decoding

Empirical Results: High-order

Does the hill-climbing find the argmax?

Overall Comparison on English

$S_{DD} = S_{HC}$
98.7%

$S_{DD} < S_{HC}$
1.0%

$S_{DD} > S_{HC}$
0.3%

Experimental Setup

Datasets

- 14 languages in CoNLL 2006 & 2008 shared tasks

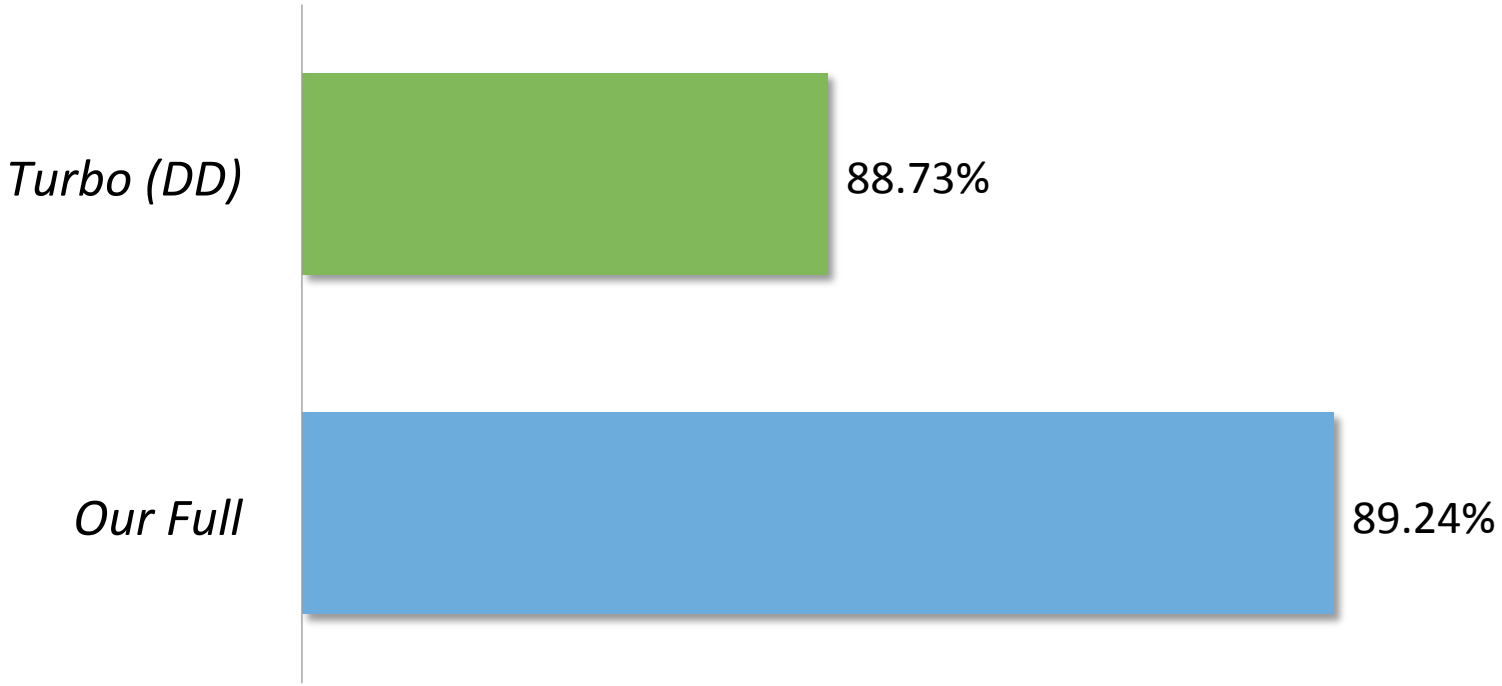
Features

- Turbo Parser: Dual Decomposition with 3rd-order features (Martins et al., 2013)

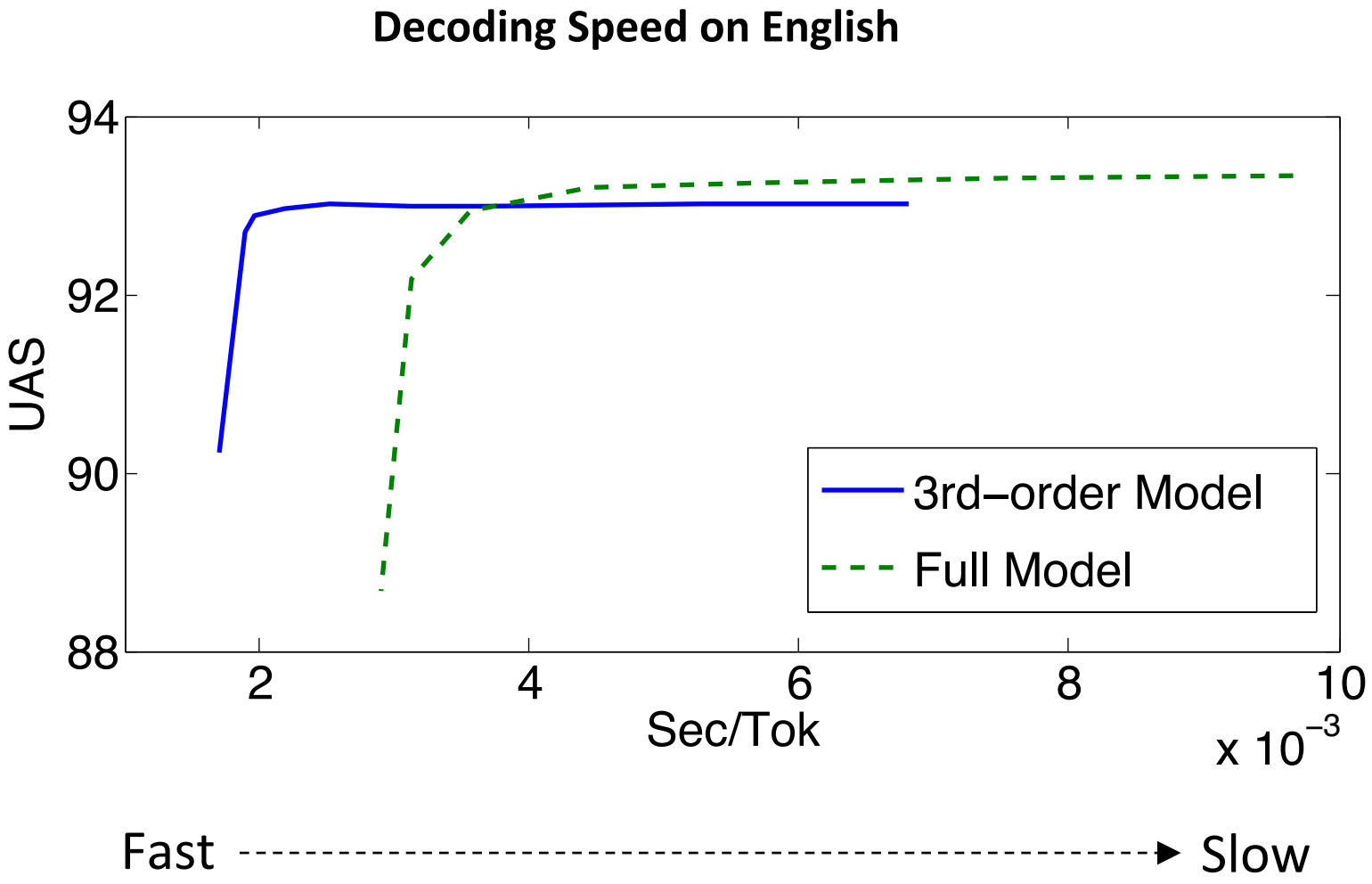
Implementation

- Adaptive restarting strategy with $K = 300$

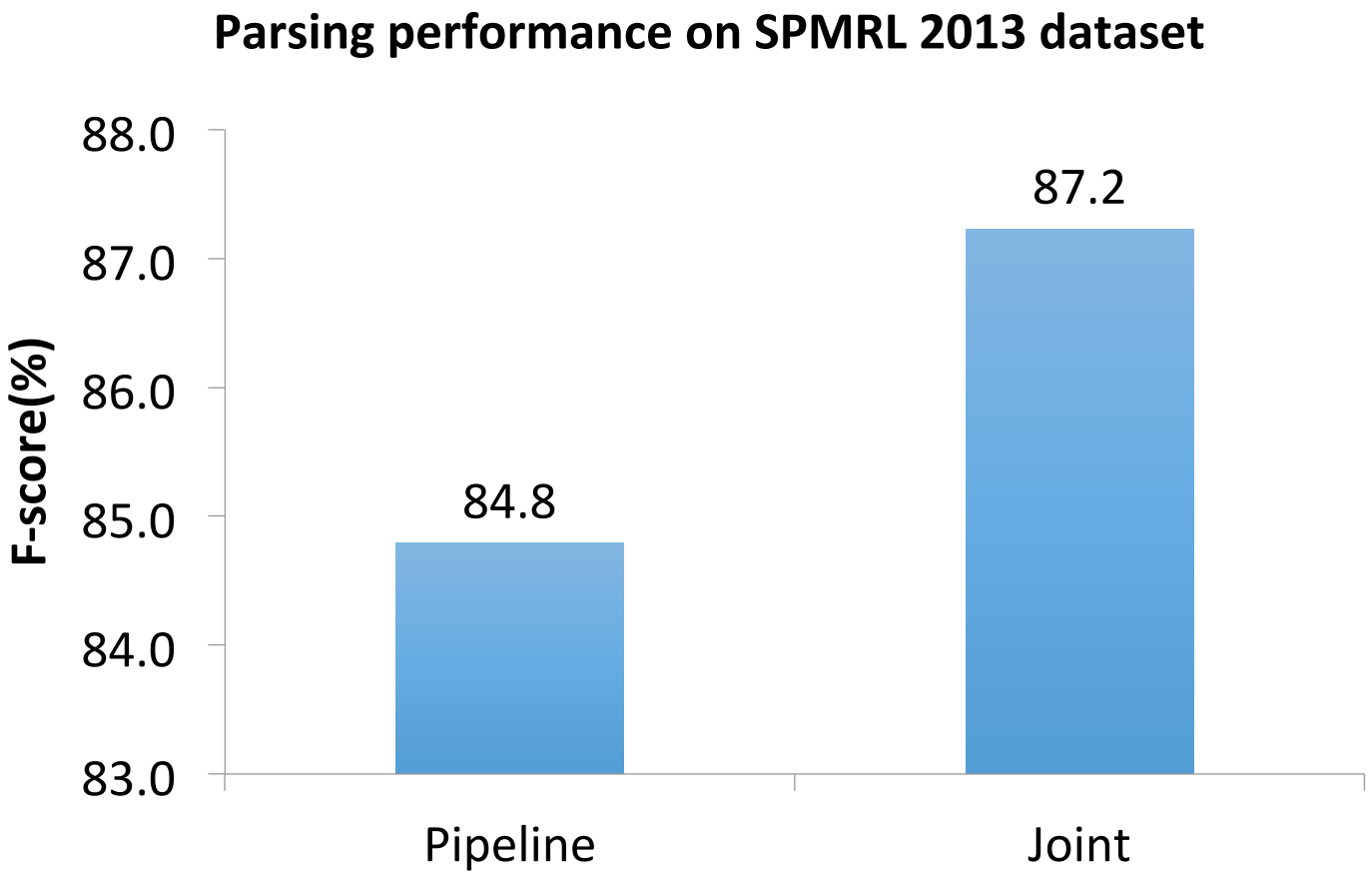
Comparing with Baselines



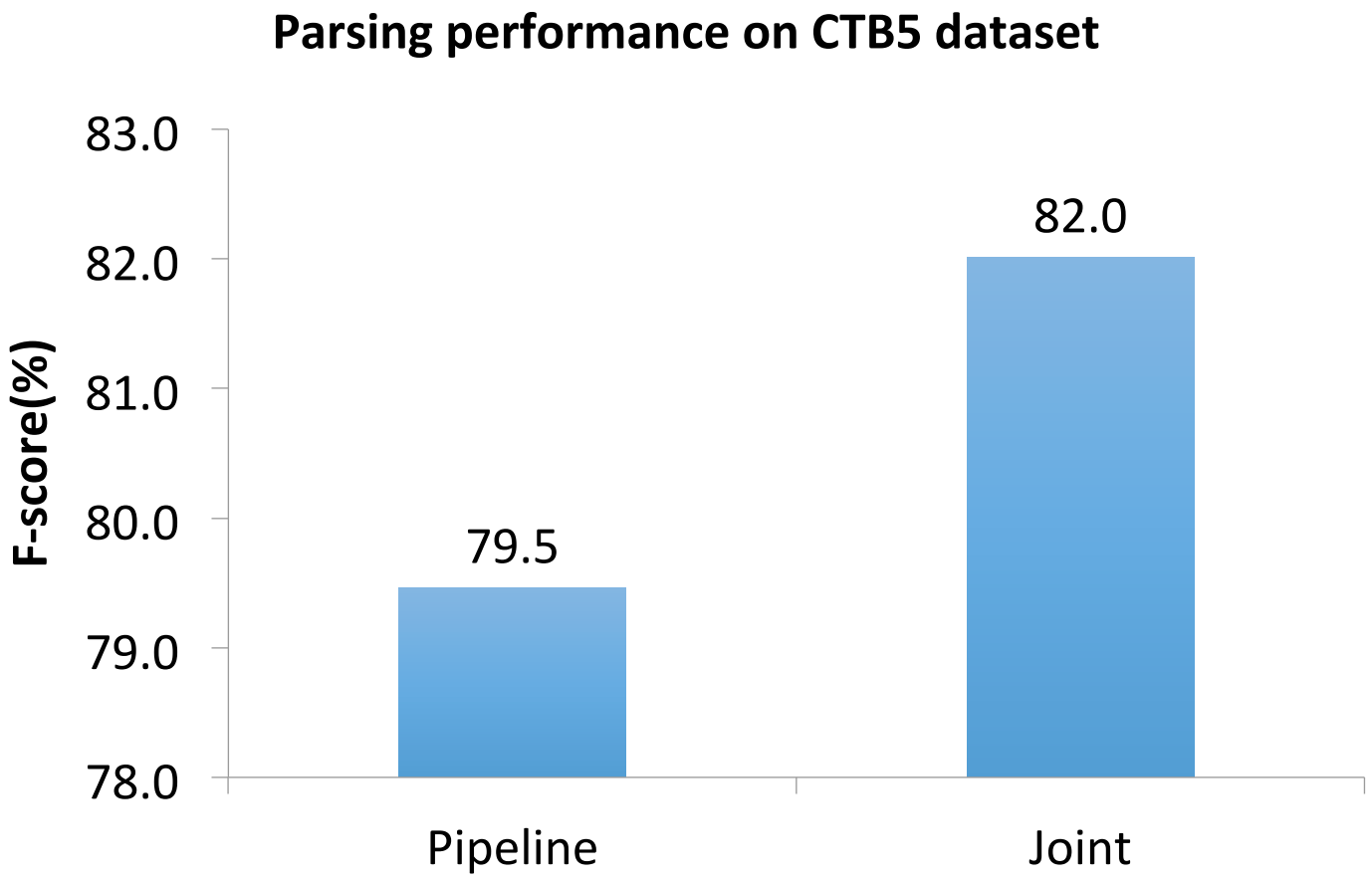
Trade-off between Speed and Performance



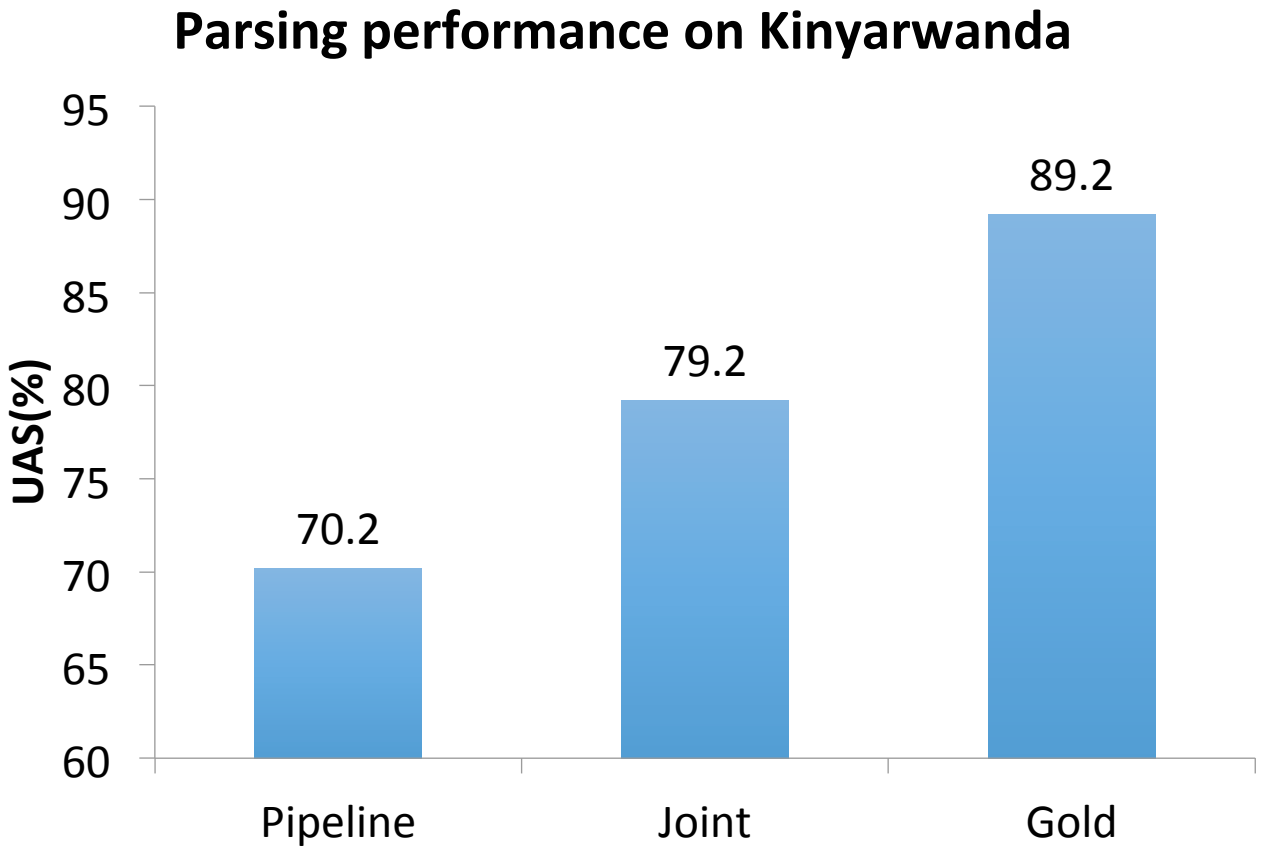
Results on Arabic: Joint Morphology, Tagging and Parsing



Results on Chinese: Joint Word Segmentation, Tagging and Parsing

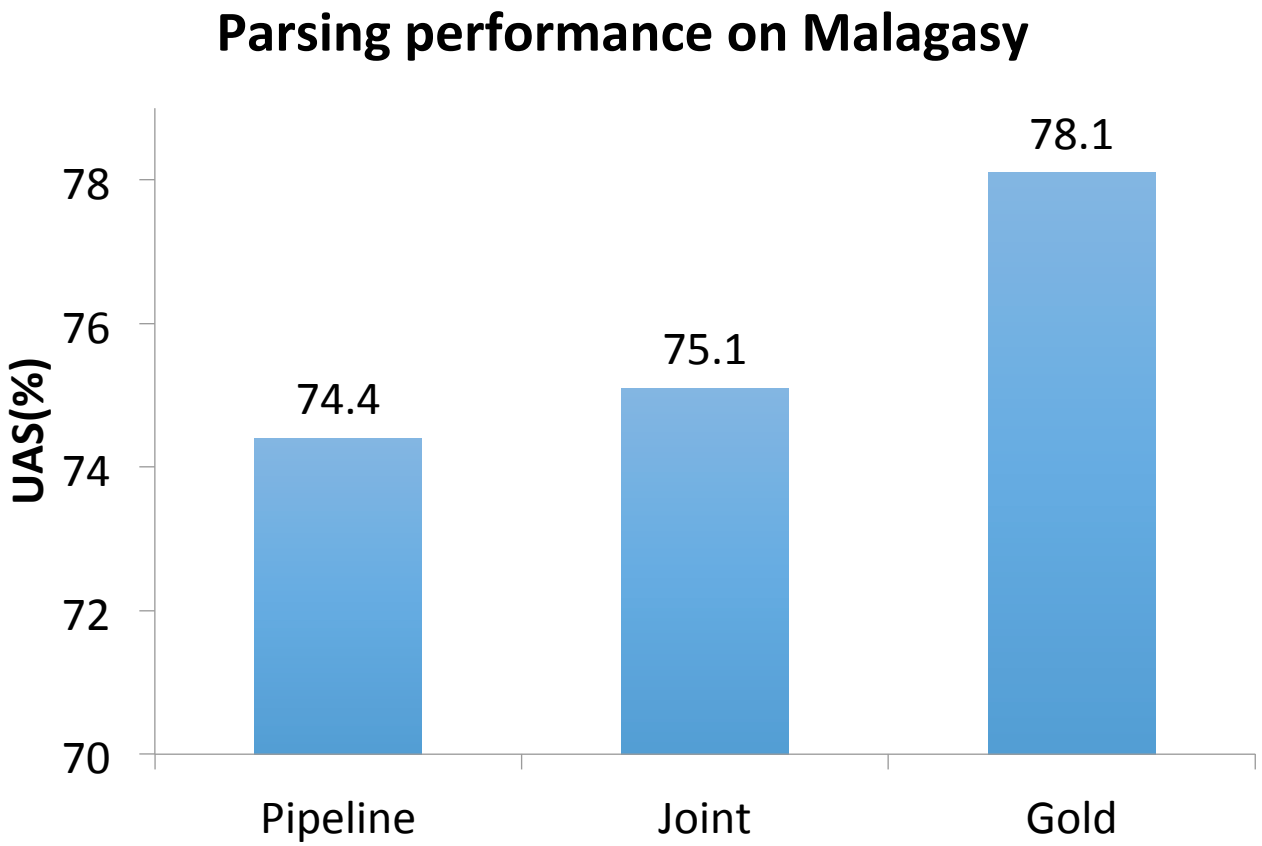


Results on Kinyarwanda: Joint Tagging and Parsing



- Tagging: select top-k including the gold

Results on Malagasy: Joint Tagging and Parsing



- Tagging: select top-k including the gold

Conclusion

- Strong monolingual parsers are effective in a low-resource setting
- The key research issue in parsing is representation, not inference
- Randomized greedy inference delivers top performance in dependency parsing and joint morphology/tagging/parsing tasks

Source code available at:
https://github.com/taolei87/RBGParser