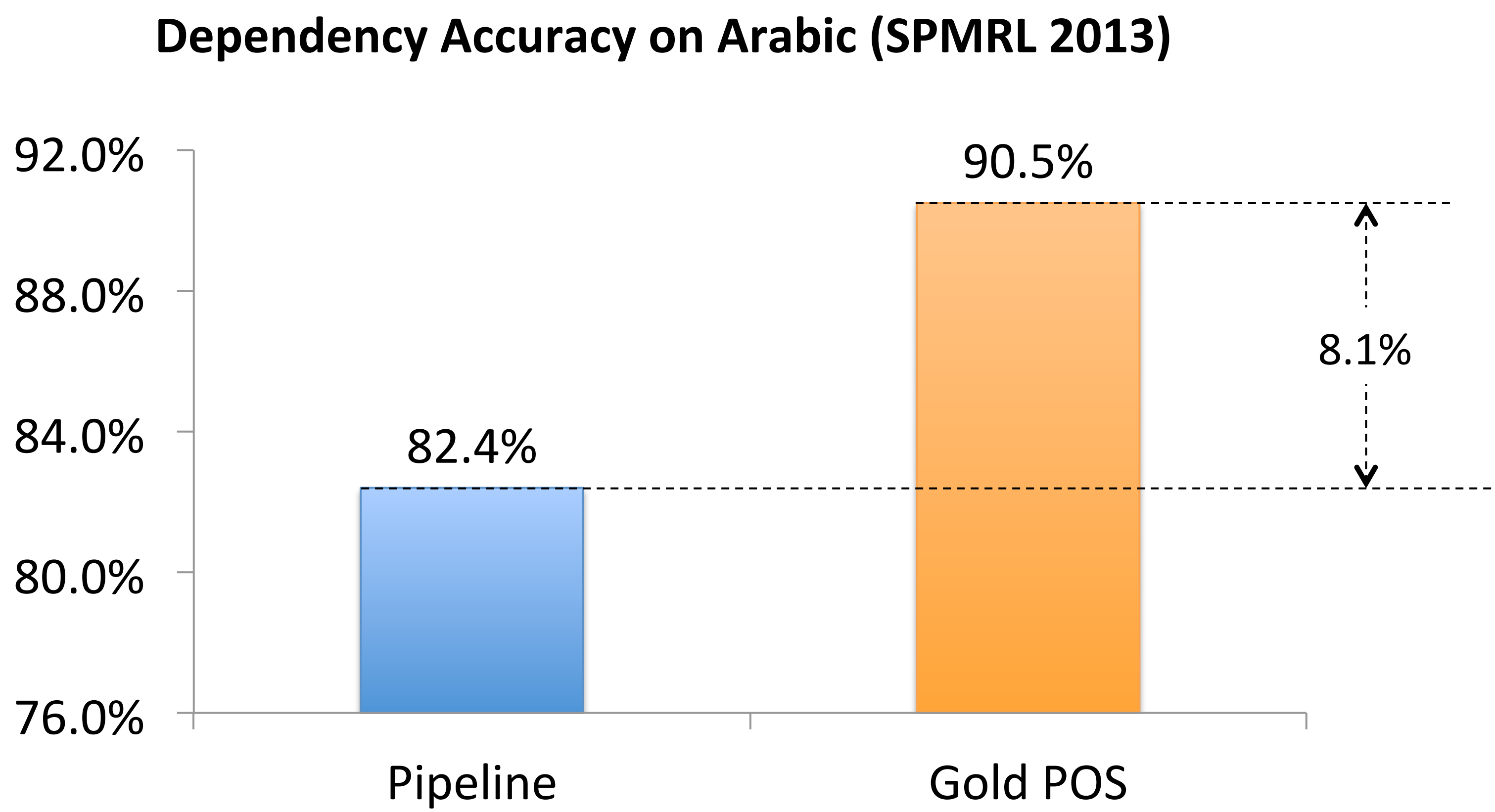
Randomized Greedy Inference for Joint Segmentation, Tagging and Parsing

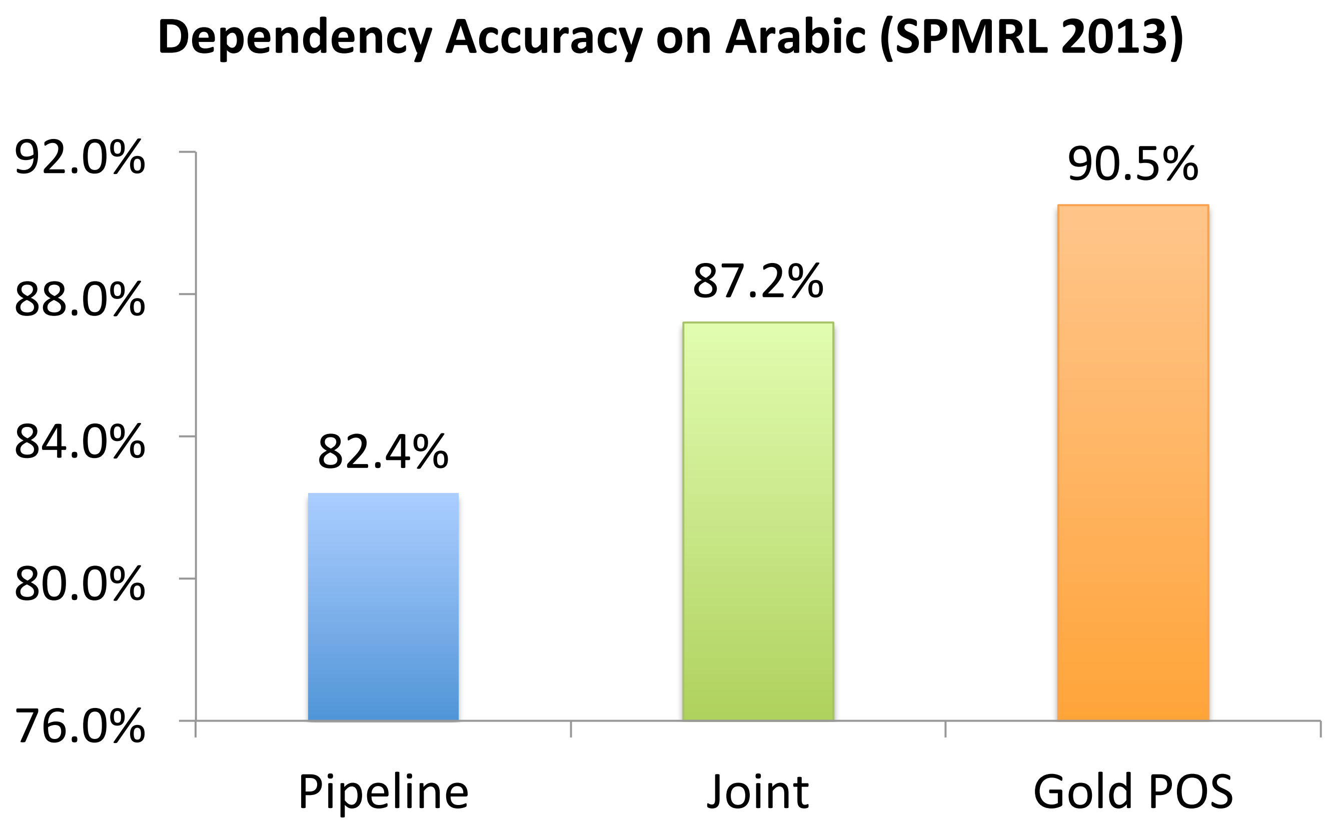
Yuan Zhang, Chengtao Li, Regina Barzilay, Kareem Darwish

MIT, QCRI

Error Propagation in Pipeline Models



Our Approach: Joint Model with Randomized Greedy

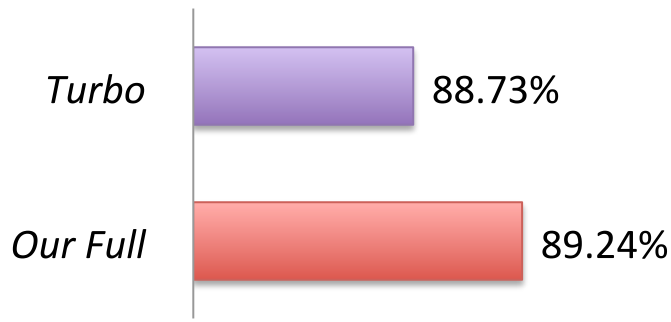


Randomized Greedy in Dependency Parsing

- Key idea: greedy hill-climbing with random restarts
- Highly effective inference procedure

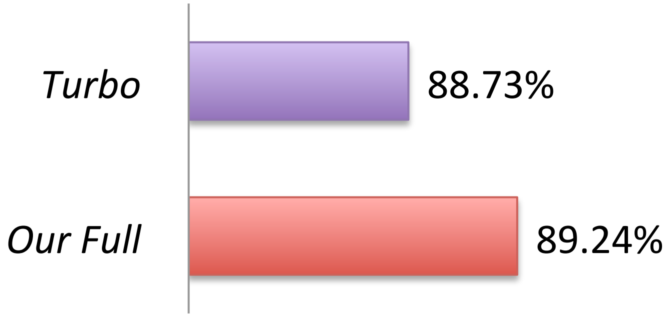
Randomized Greedy in Dependency Parsing

- Key idea: greedy hill-climbing with random restarts
- Highly effective inference procedure



Randomized Greedy in Dependency Parsing

- Key idea: greedy hill-climbing with random restarts
- Highly effective inference procedure

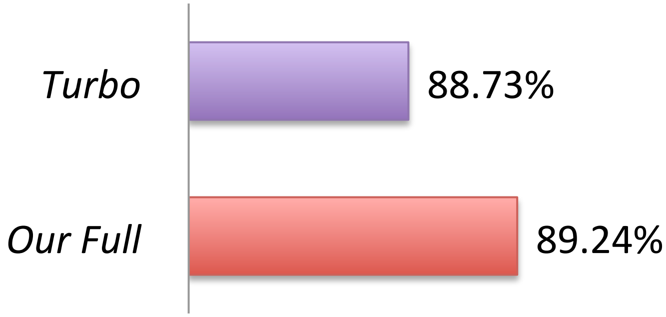


Finding Global Optimum on English

Len. ≤ 15	Len. > 15
100%	99.3%

Randomized Greedy in Dependency Parsing

- Key idea: greedy hill-climbing with random restarts
- Highly effective inference procedure

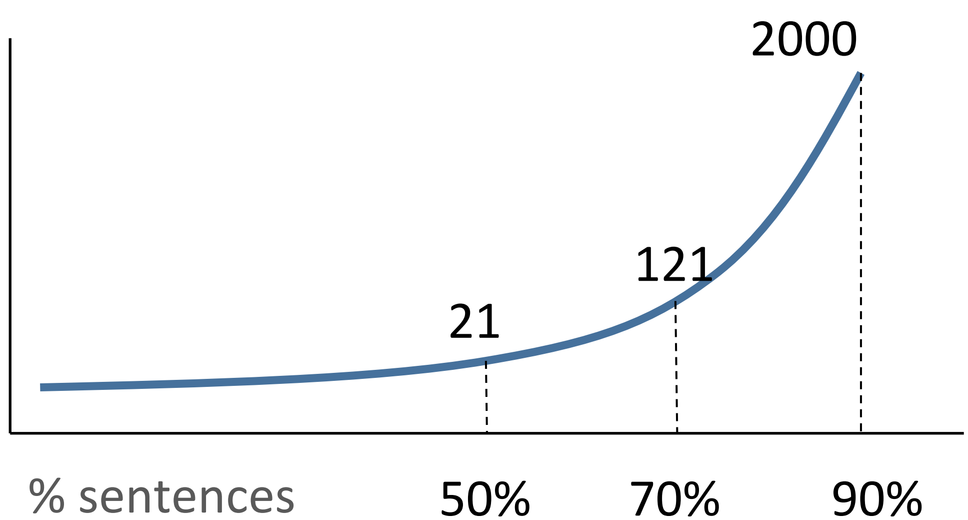


Finding Global Optimum on English

Len. ≤ 15	Len. > 15
100%	99.3%

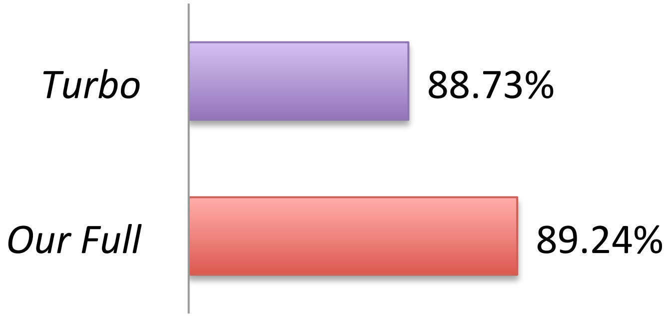
- Analysis: parsing is easy on average

 # Optima on English Dataset



Randomized Greedy in Dependency Parsing

- Key idea: greedy hill-climbing with random restarts
- Highly effective inference procedure

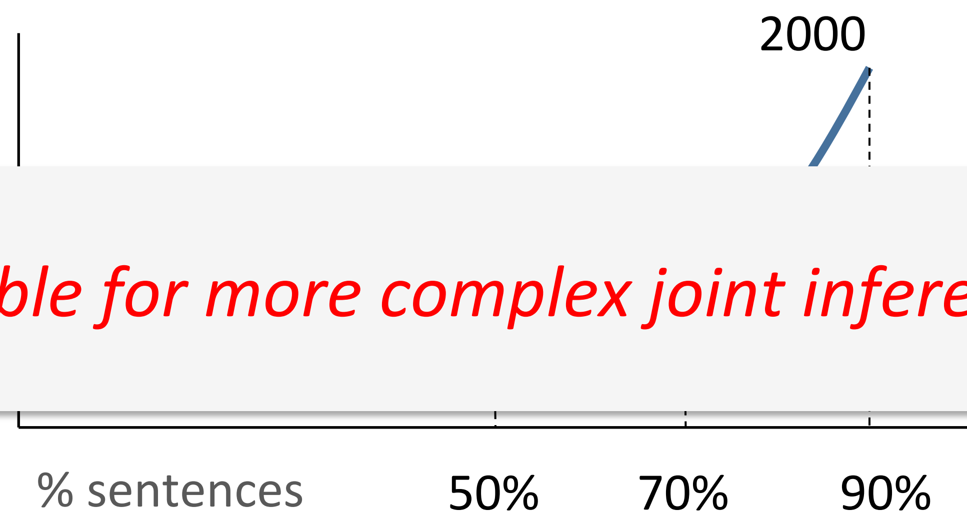


Finding Global Optimum on English

Len. ≤ 15	Len. > 15
100%	99.3%

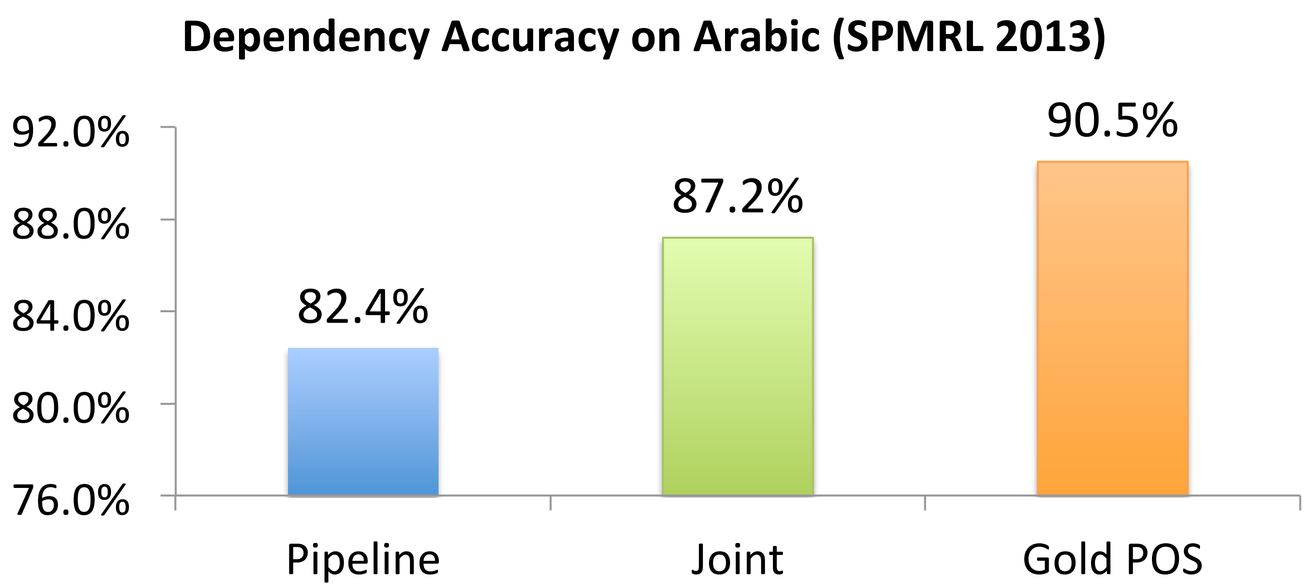
- Analysis: parsing is easy on average

 # Optima on English Dataset



Scalable for more complex joint inference?

Randomized Greedy for Joint Prediction



- Advantages:
  - No constraints on the scoring function
  - Easy language adaptation
  - Easy parallelization

Core Idea

- Climb to the optimal assignment for (s, t, y) in a few small greedy steps

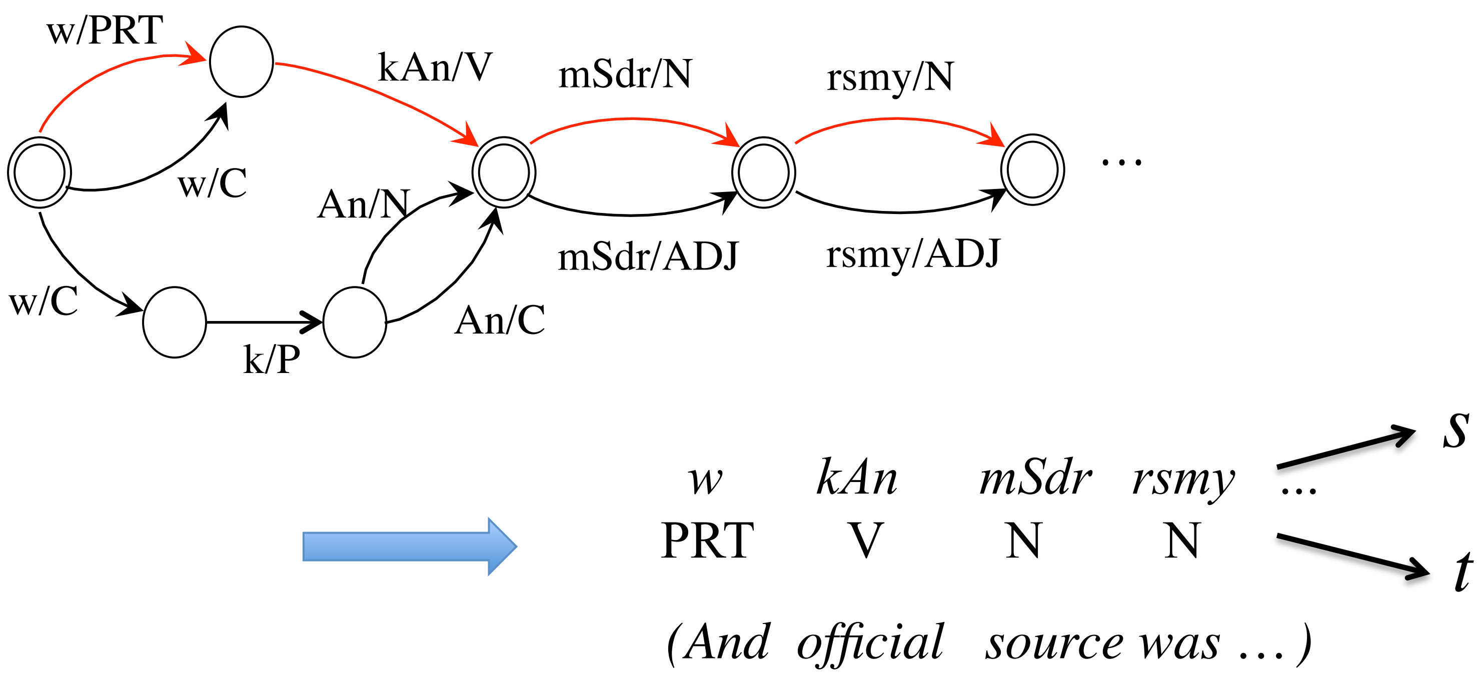
Randomized Hill-climbing

For *k* = 1 to *K*

1) Sample segmentation *s*, POS tags *t* and a dependency tree *y*
2) Greedily improve the POS tags and the tree
3) Repeat (2) until converge

Select the assignment with the highest score

Sample Segmentation and POS Tag

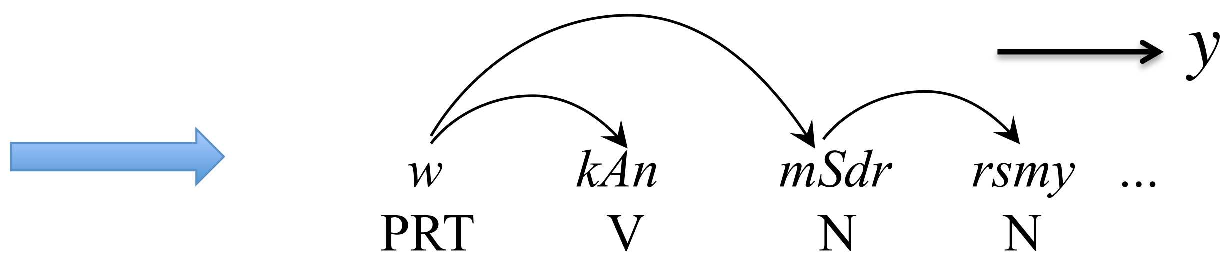


- Sample from first-order distribution

$$p(s) \propto \exp\{\theta \cdot f(s)\}, p(t) \propto \exp\{\theta \cdot f(s,t)\}$$

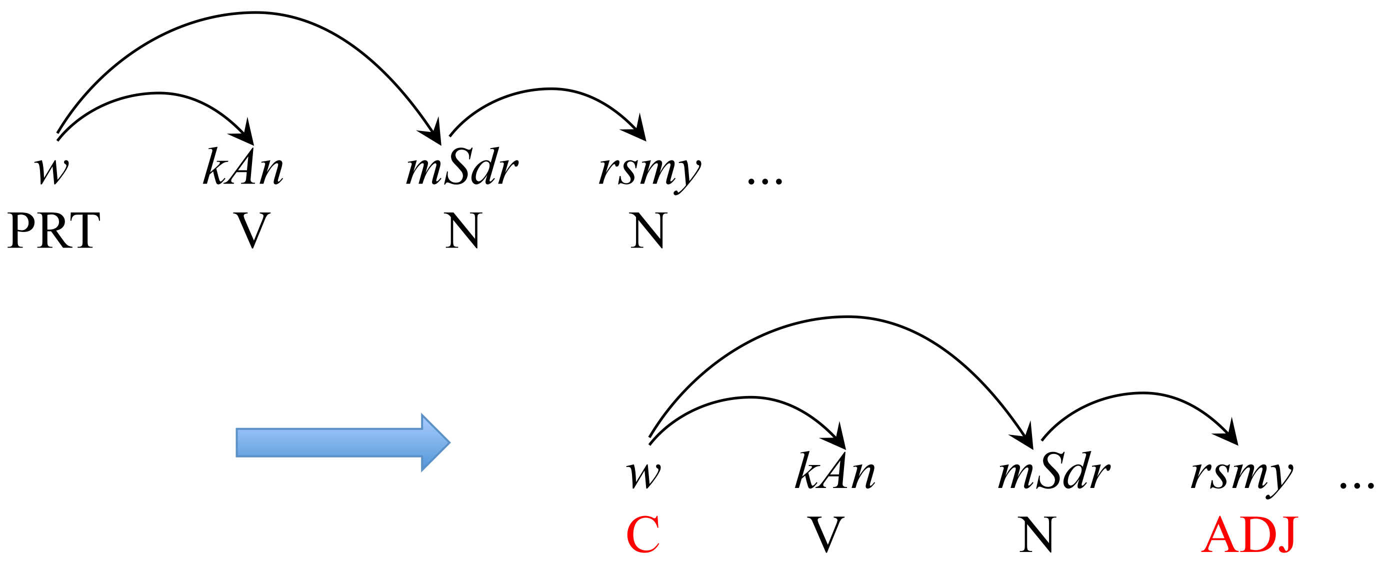
Sample Tree

w	*kAn*	*mSdr*	*rsmy*	…
PRT	V	N	N	



Sample using a random walk-based algorithm (Wilson, 1996)

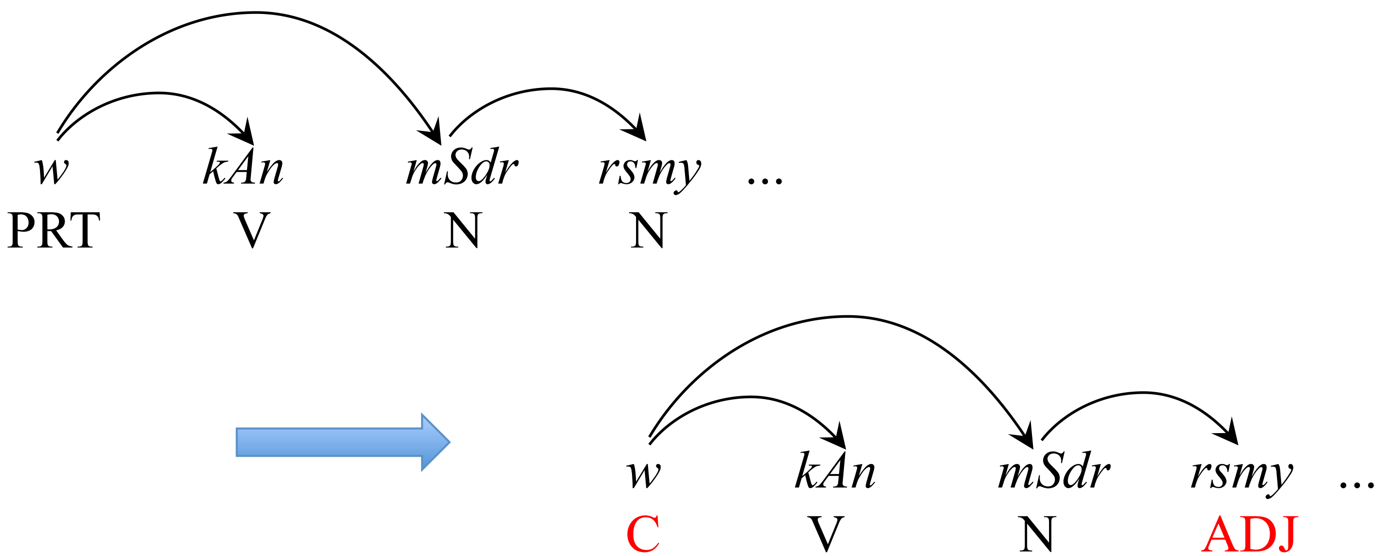
Improve POS Tag



- Update each POS to maximize the full scoring function

$$t_{i,j} \leftarrow \underset{t_{i,j}}{\operatorname{argmax}}\{\theta \cdot f(s, t_{i,j}, t_{-(i,j)}, y)\}$$

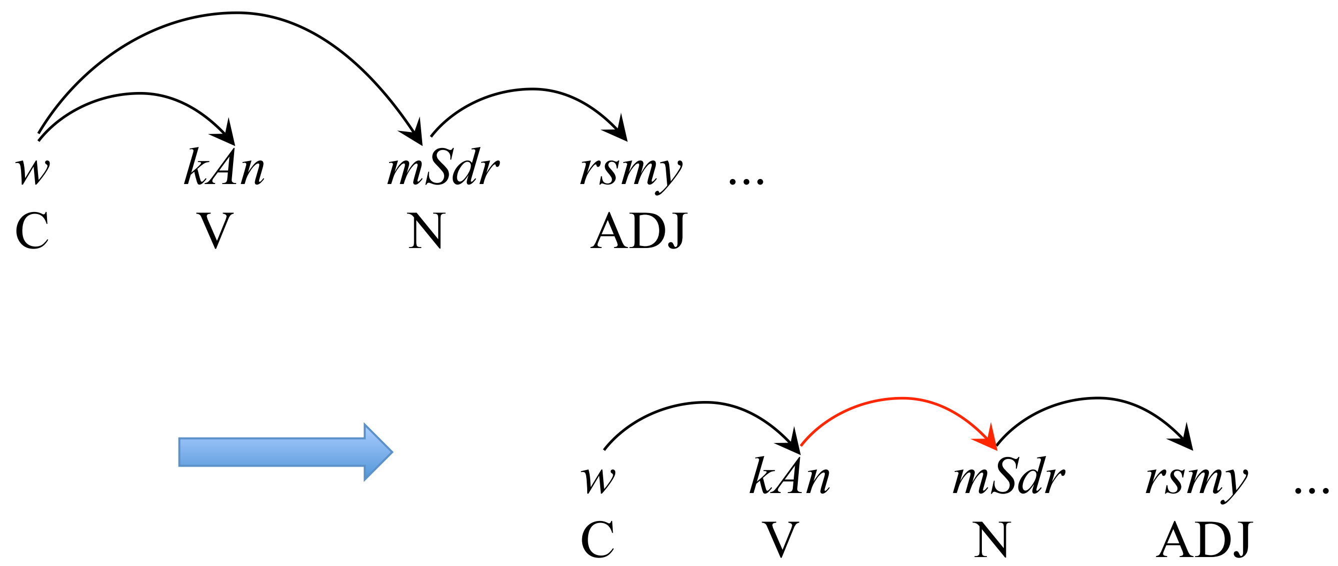
Improve POS Tag



- Update each POS to maximize the full scoring function

$$t_{i,j} \leftarrow \underset{t_{i,j}}{\operatorname{argmax}} \{\theta \cdot f\boxed{(s, t_{i,j}, t_{-(i,j)}, y)}\}$$

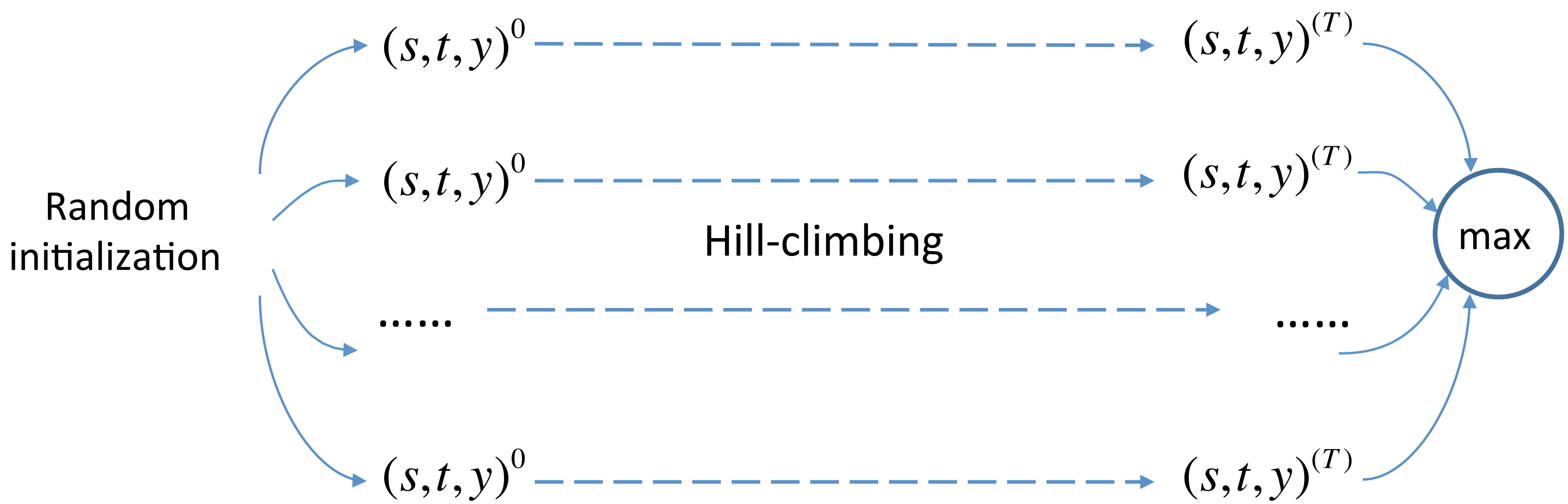
Improve Tree



- Update each dependency to maximize the full scoring function

$$y_{i,j} \leftarrow \operatorname*{argmax}_{y_{i,j}} \{\theta \cdot f\boxed{(s,t,y_{i,j},y_{-(i,j)})}\}$$

Hill-climbing with Restarts



- Overcome local optima via restarts
- Parallelize each run during hill-climbing

Learning Algorithm

- Follow common max-margin framework

$$\theta \cdot f(x, \hat{s}, \hat{t}, \hat{y}) \geq \theta \cdot f(x, s, t, y) + Err(s, t, y) - \xi$$

  - $\hat{s}, \hat{t}, \hat{y}$ are gold values of segmentation, POS tags and dependencies

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

- Decode with our randomized greedy algorithm

Generating Lattice Structure: Arabic

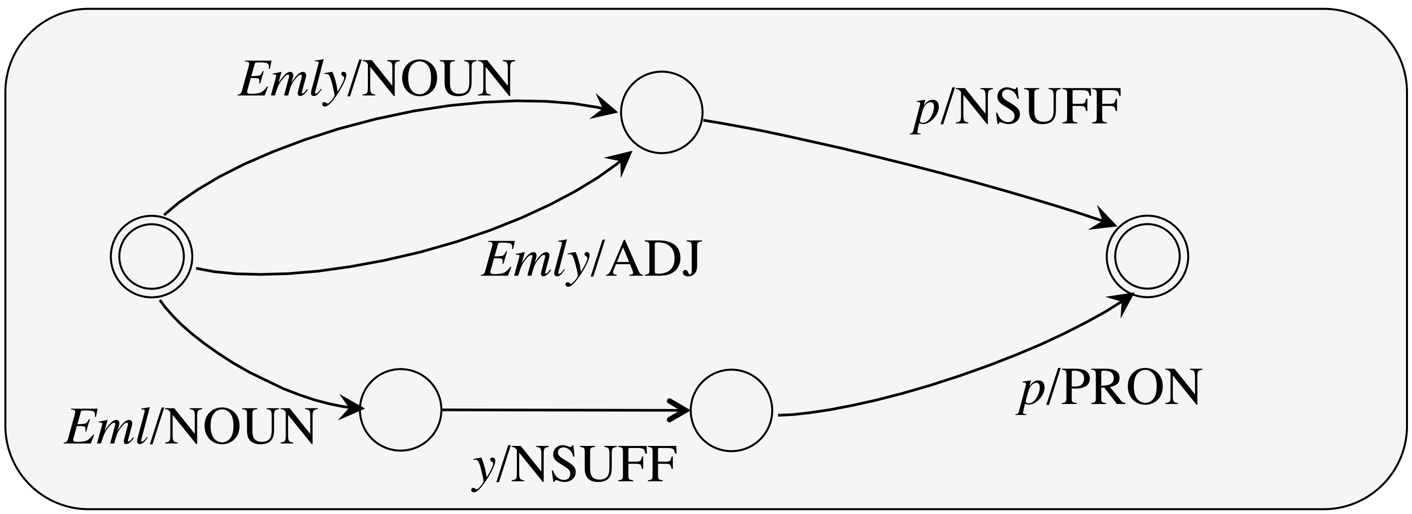
- Use MADA to generate top-*k* morphological analyses
- Convert analyses to equivalent lattice

Word *Emlyp*:

Emly/NOUN + *p*/NSUFF

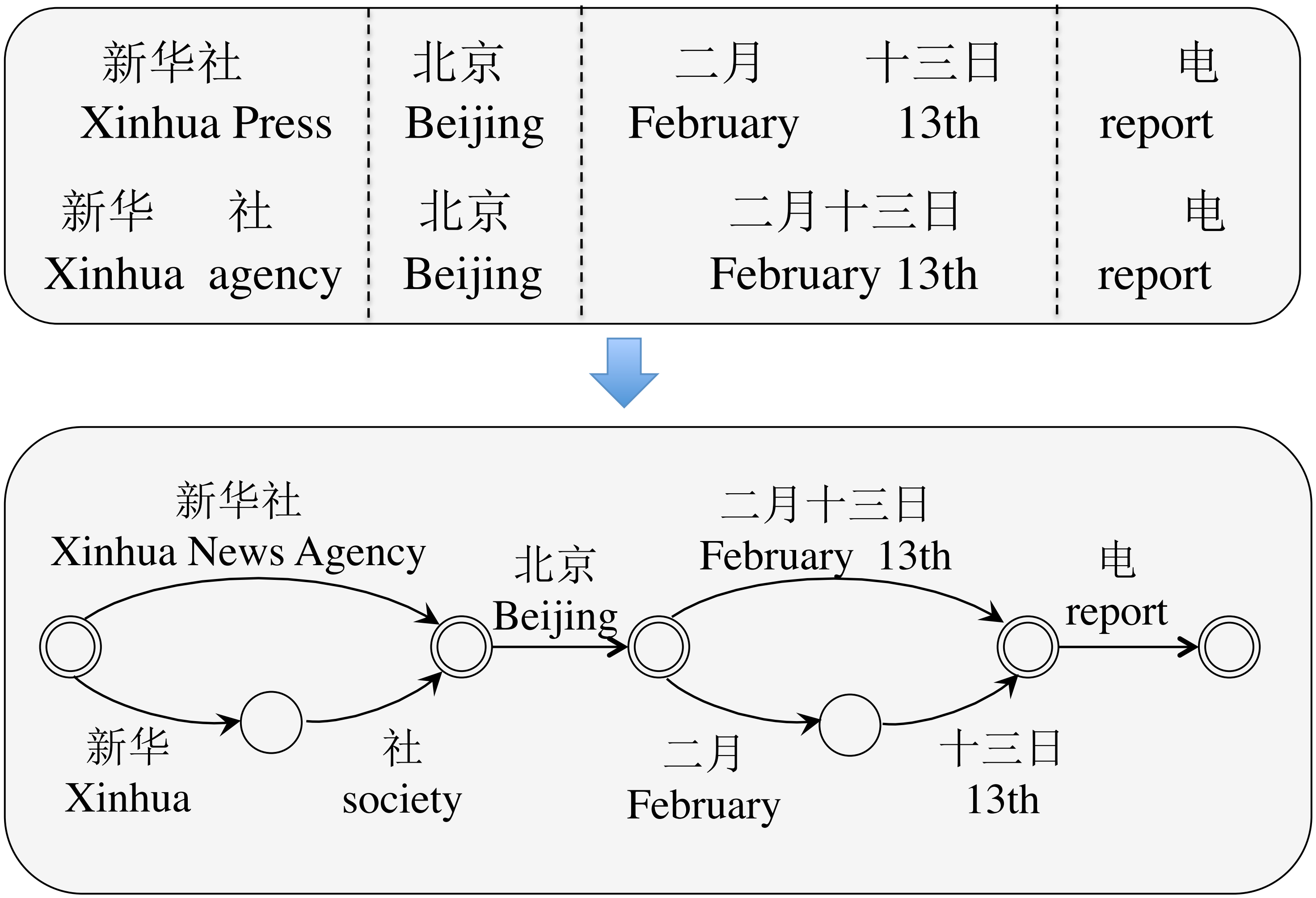
Emly/ADJ + *p*/NSUFF

Eml/NOUN + *y*/NSUFF + *p*/PRON



Generating Lattice Structure: Chinese

- Use Stanford word segmenter to generate top-*k* segmentation
- Convert segmentation to equivalent lattice



Experimental Setup

- Datasets
  - Chinese Penn Treebank 5.0 (CTB5)
  - Modern Standard Arabic (MSA): the SPMRL 2013 dataset
  - Mixed Arabic dataset
    - Training: MSA
    - Testing: Classical Arabic
    - Different vocabulary but similar grammar

- Evaluation Metric
  - F-score for segmentation, POS tagging and dependency parsing
  - TedEval (Tsarfaty et al. 2012) for the SPMRL dataset
    - A joint evaluation of segmentation and parsing quality

Baselines

- State-of-the-art
  - The SPMRL 2013 dataset: pipeline system (Björkelund et al. 2013)
  - CTB5: transition-based model (Zhang et al. 2014)

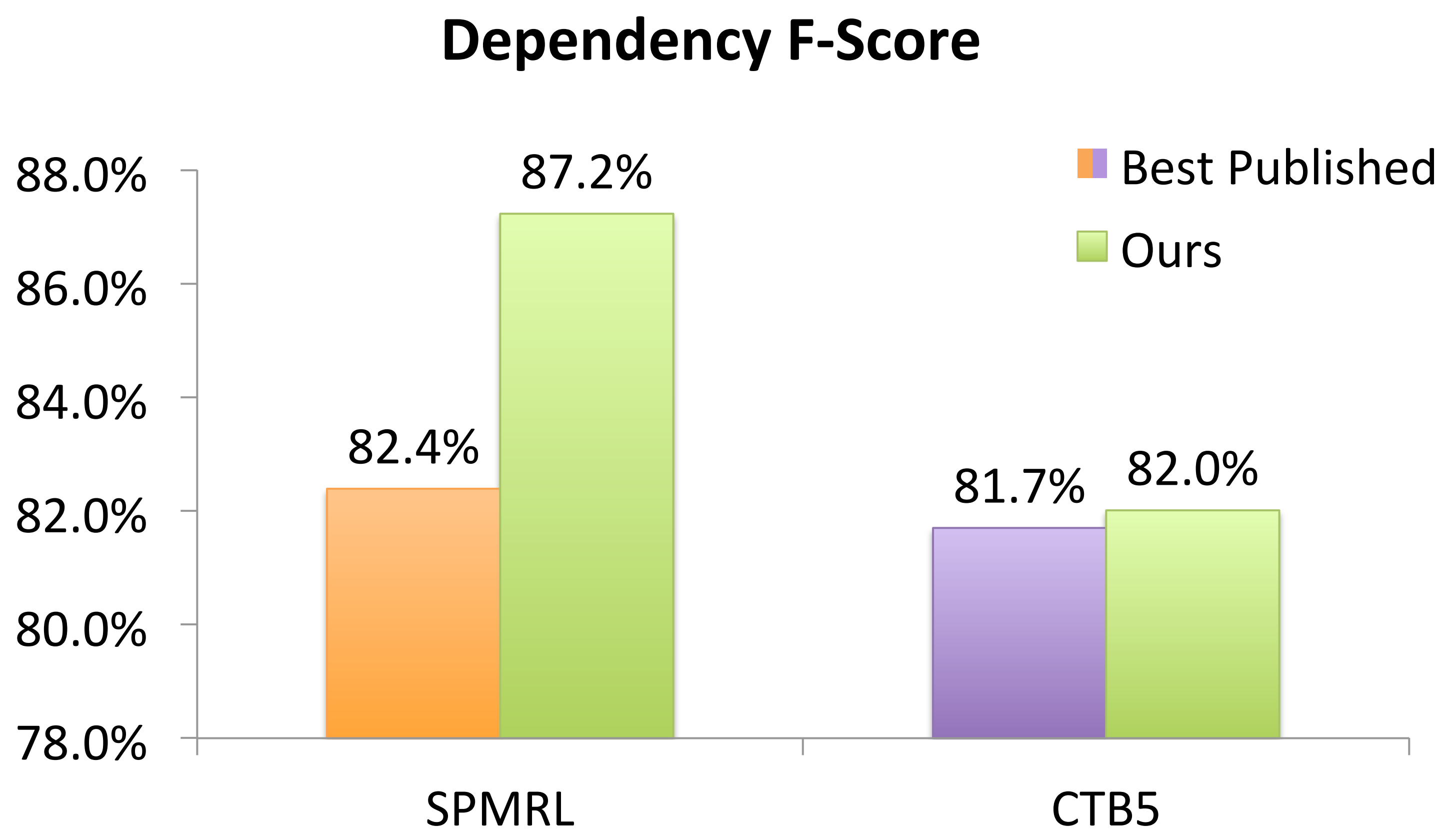
- Pipeline variants of our model
  - Predicted POS tags and segmentations by the same systems that we use to generate candidates

Features

- Segmentation
  - Morphemes/words scores, character-based features
- POS tagging
  - Up to 5-gram features, character-based features
- Dependency parsing
  - Up to 3rd-order (three arcs) features used in standard parsing

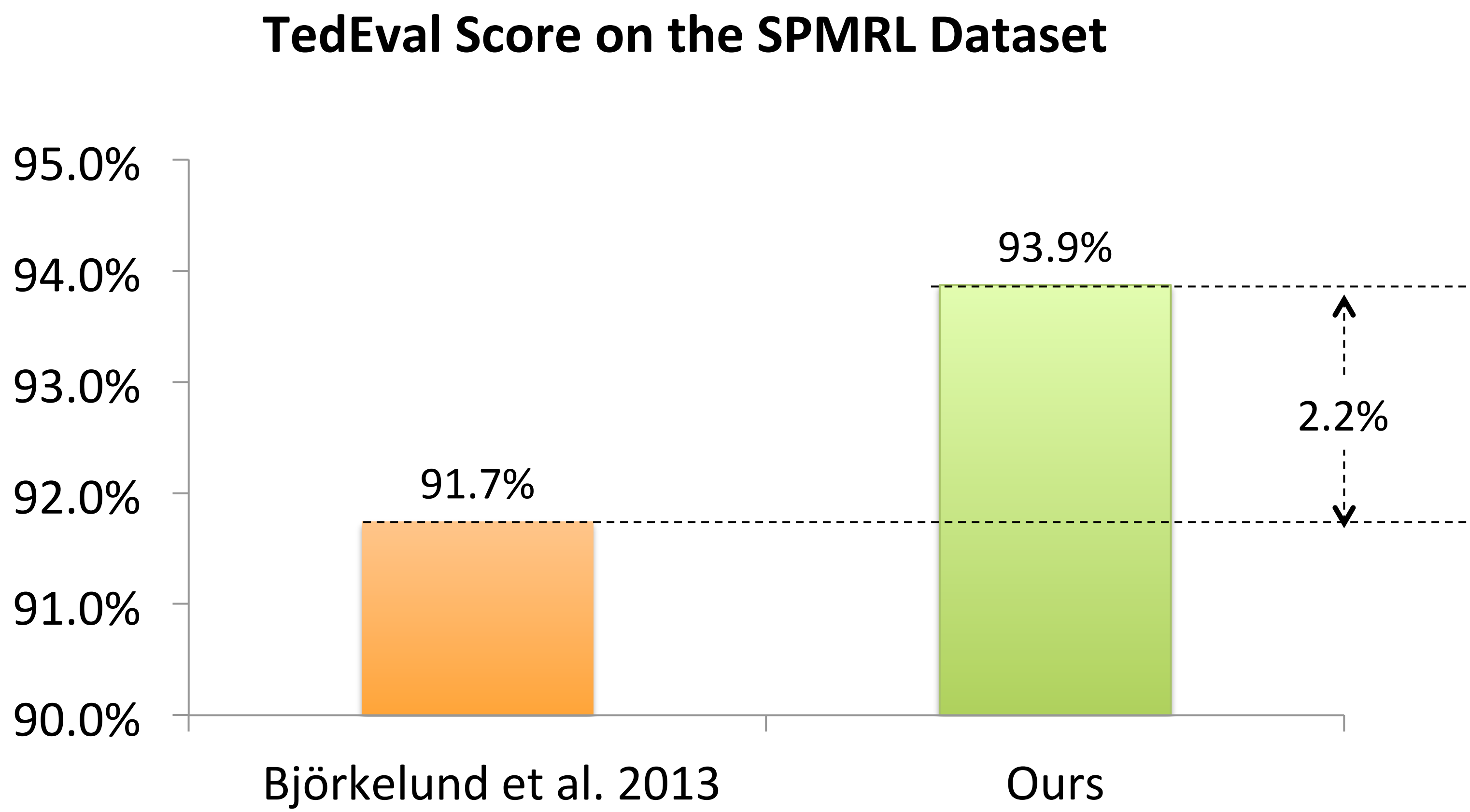
➢ Note: scoring function combines all features and capture cross-task interaction

Comparison to State-of-the-art Systems



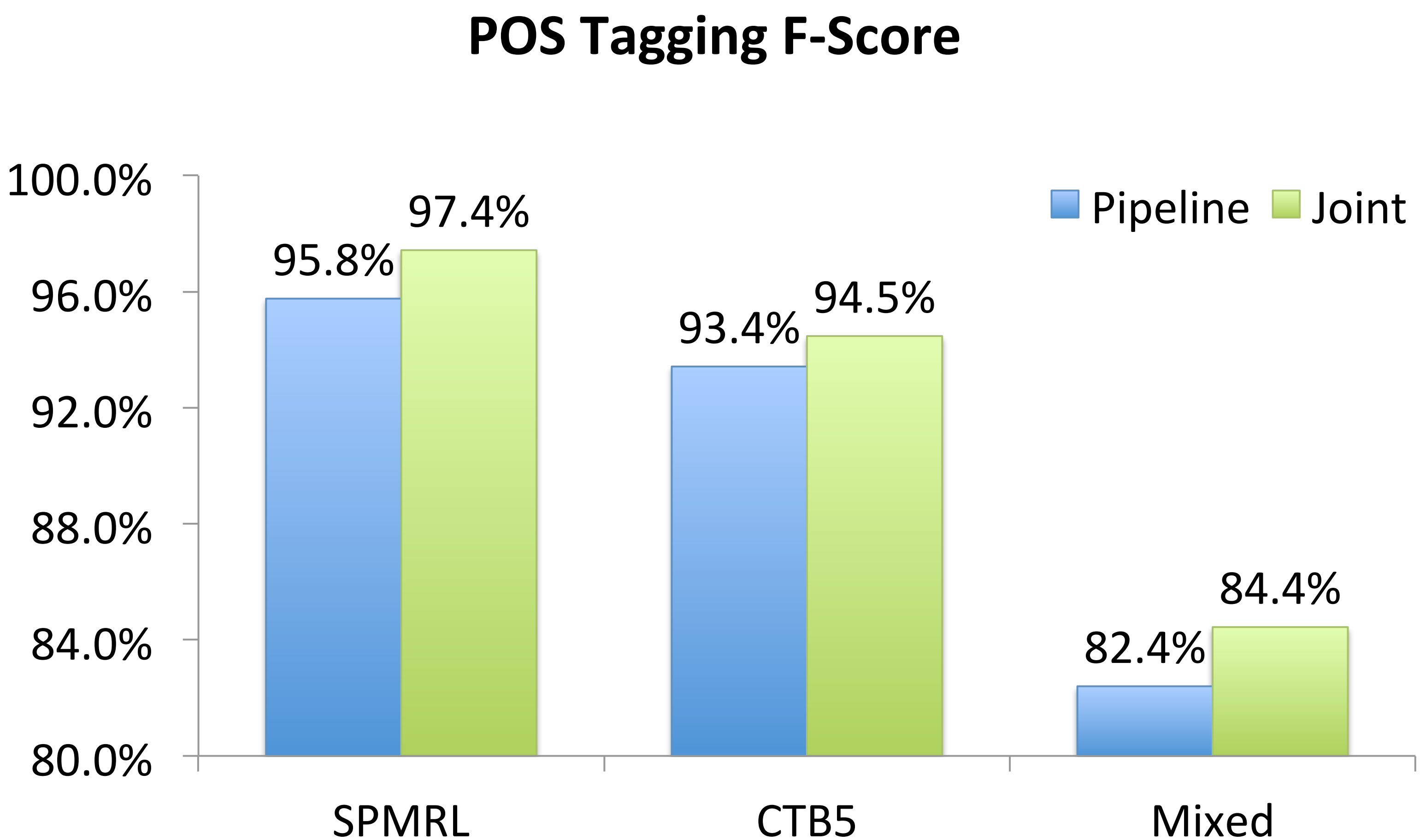
- SPMRL: pipeline model (Björkelund et al. 2013)
- CTB5: transition-based model (Zhang et al. 2014)

Comparison to State-of-the-art Models



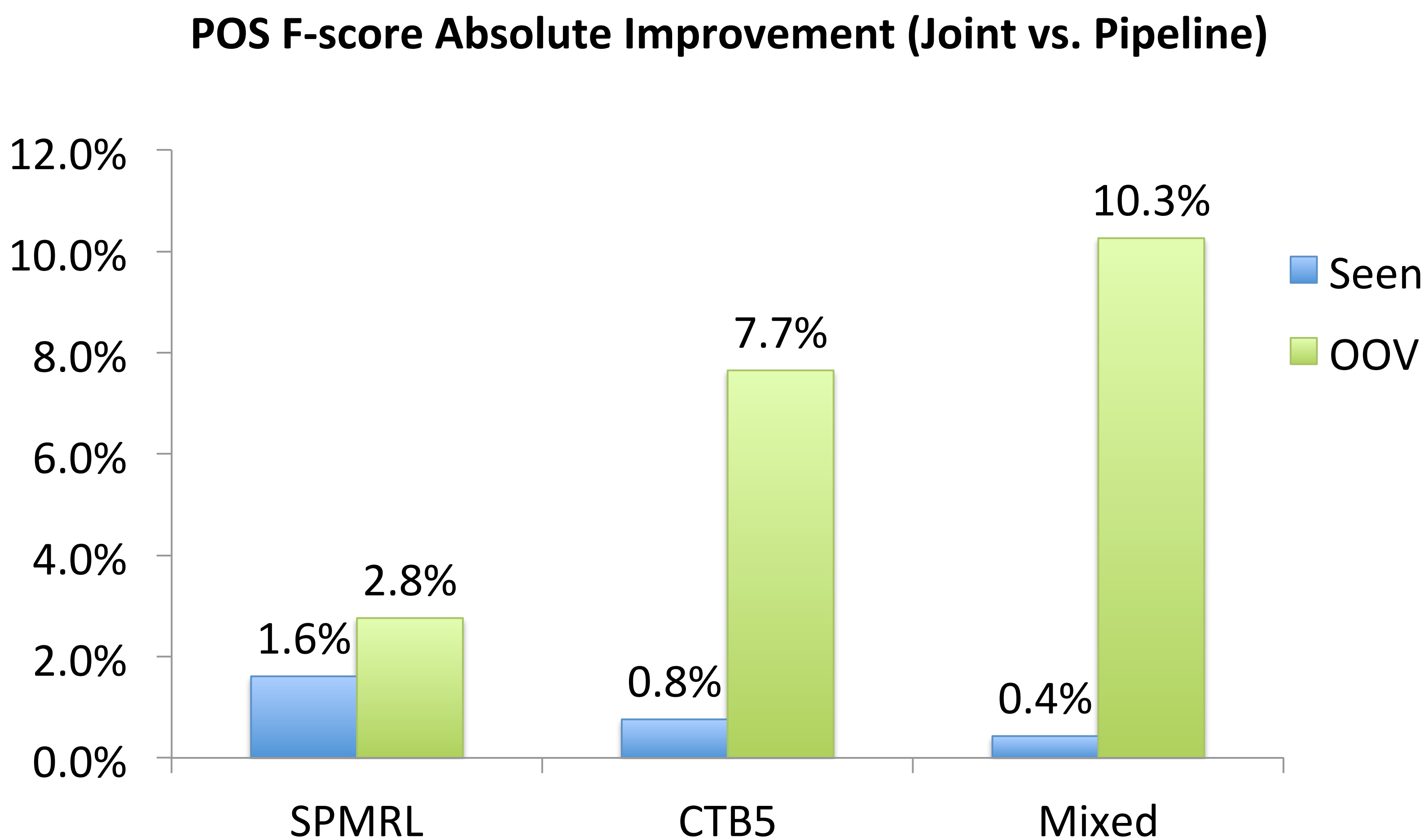
- 27% error reduction on the TedEval score

Joint vs. Pipeline Model



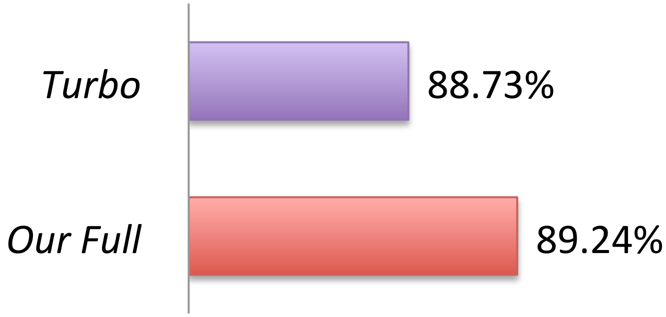
- 38% error reduction on the SPMRL dataset

Impact on Seen and OOV Words



Randomized Greedy in Dependency Parsing

- Key idea: greedy hill-climbing with random restarts
- Highly effective inference procedure



Finding Global Optimum on English

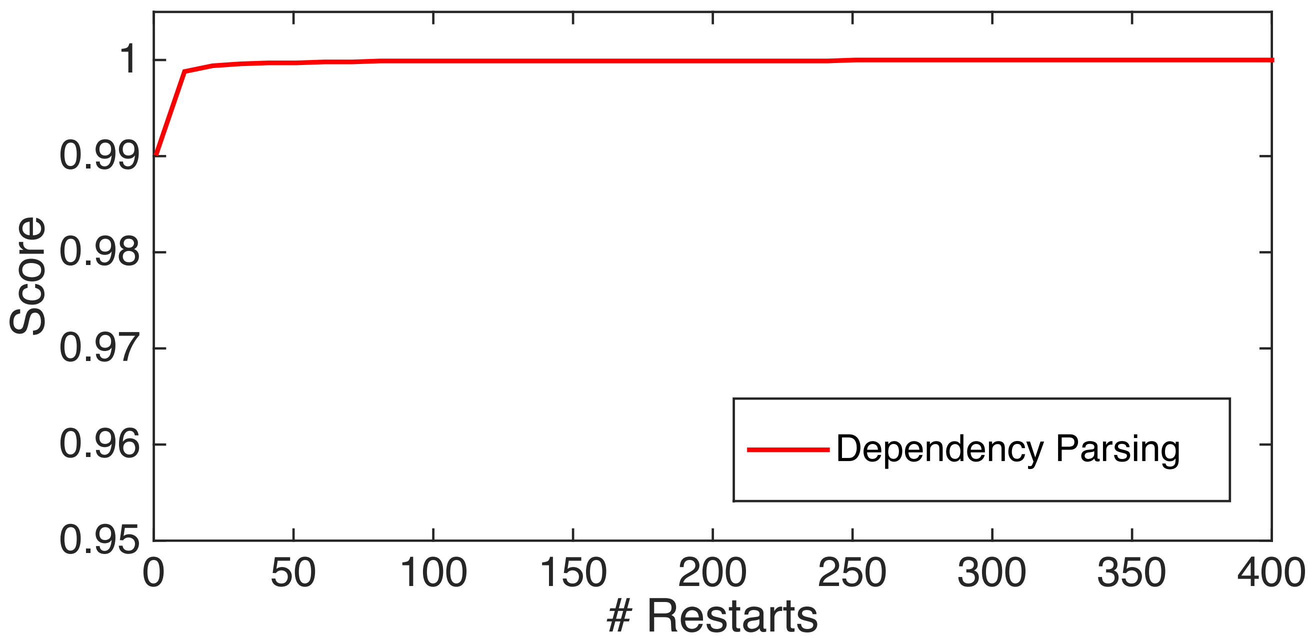
Len. ≤ 15	Len. > 15
100%	99.3%

- Analysis: parsing is easy on average

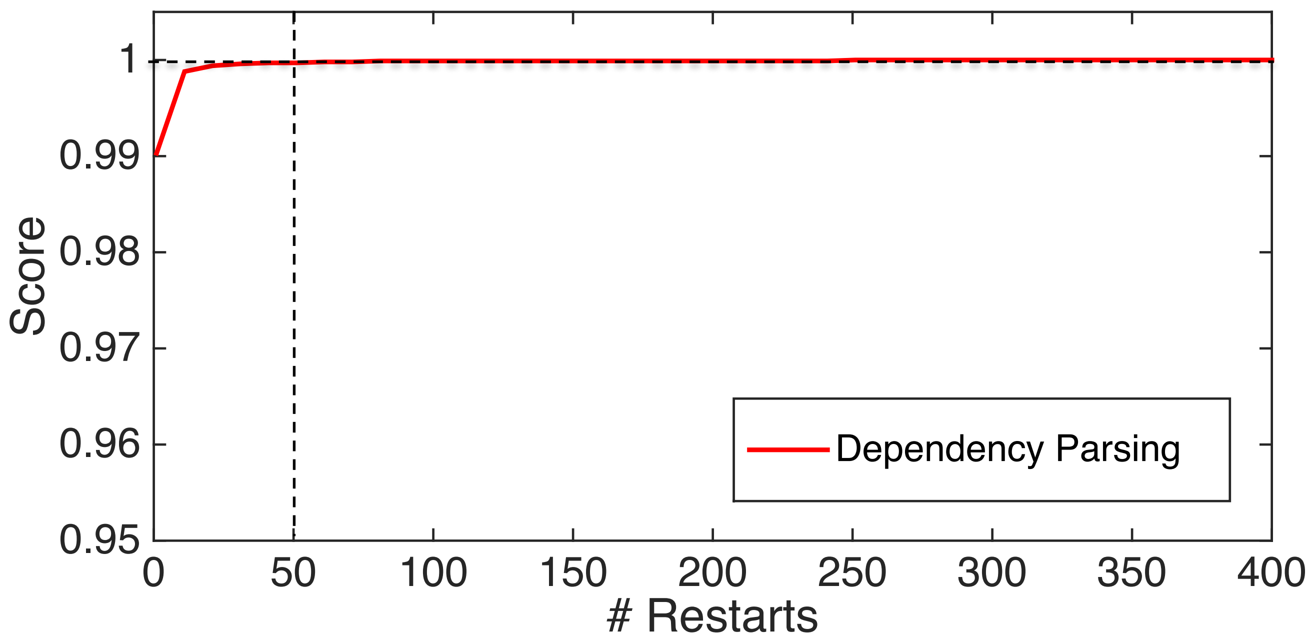
 # Optima on English Dataset

Scalable for more complex joint inference?

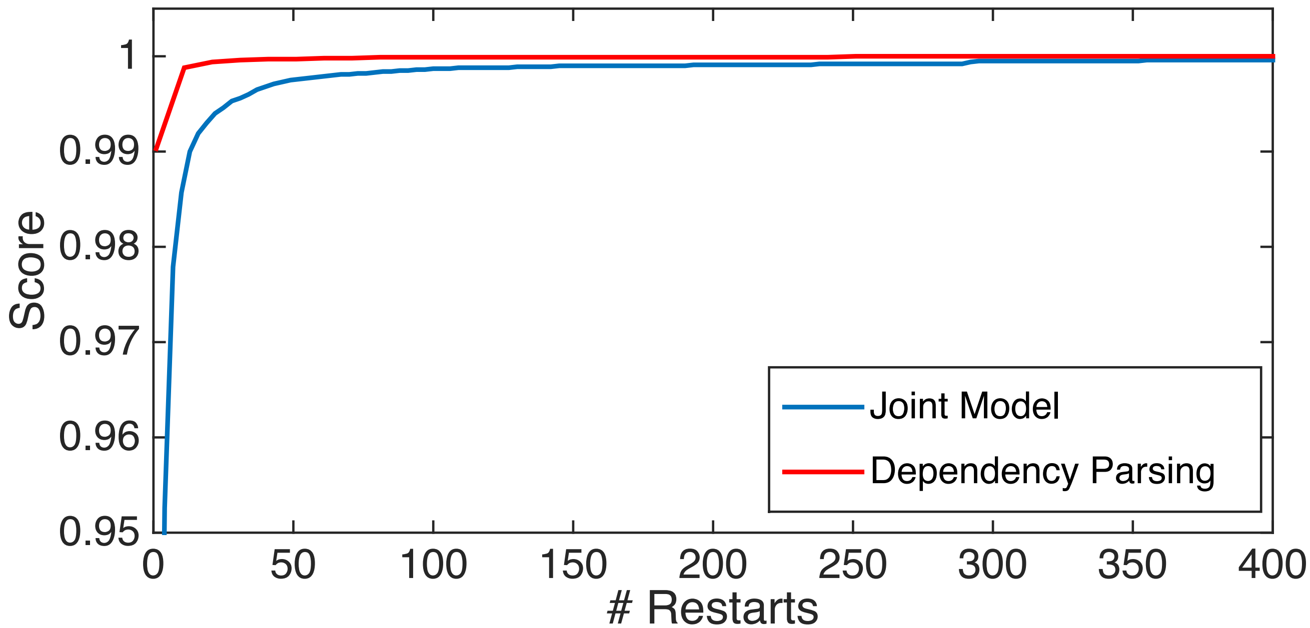
Convergence Properties: Dependency Parsing



Convergence Properties: Dependency Parsing

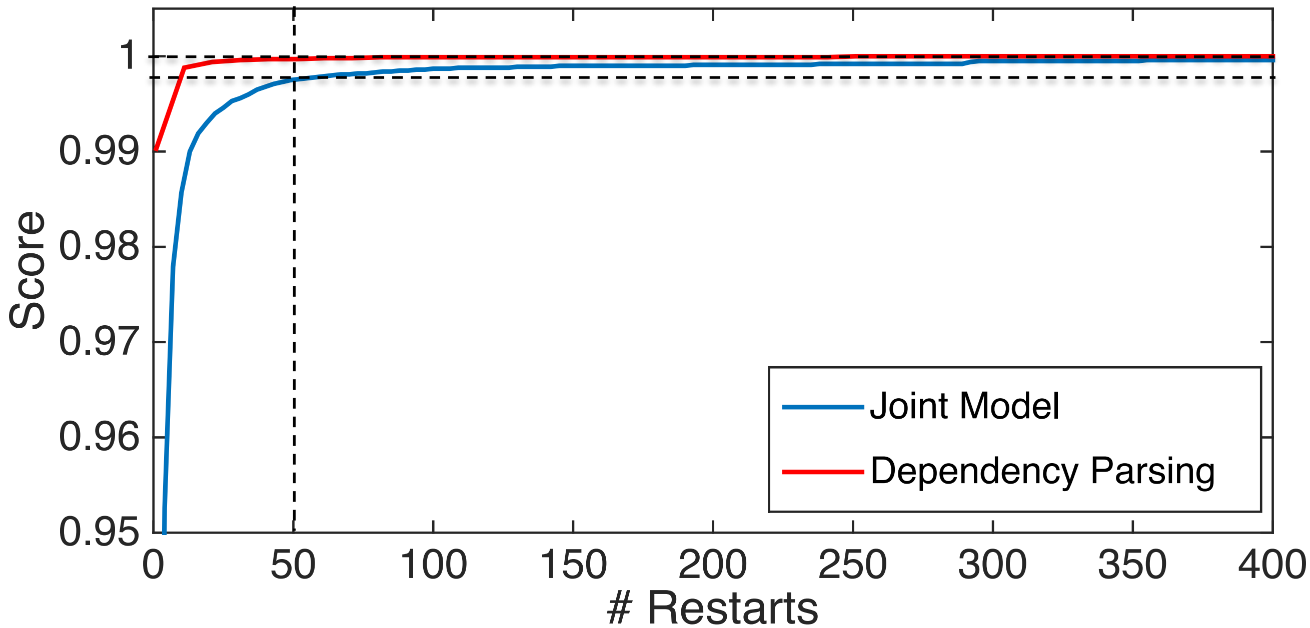


Joint Model vs. Dependency Parsing



- Both tasks exhibit similar convergence

Joint Model vs. Dependency Parsing



- Both tasks exhibit similar convergence

Conclusion

- Randomized greedy algorithm scales up for joint prediction tasks

- Our model outperforms the state-of-the-art systems and its pipeline variant on both Arabic and Chinese

Source code available at:

https://github.com/yuanzh/SegParser

Thank You!