

Using Left-corner Parsing to Encode Universal Structural Constraints in Grammar Induction

Hiroshi Noji

Nara Institute of
Science and Technology



Yusuke Miyao

National Institute of
Informatics



Mark Johnson

Macquarie University



Grammar induction is difficult

- ▶ Task: finding syntactic patterns without treebanks (supervision)
- ▶ We need a good *prior, or constraints*, to the grammars
 - Such constraints should be *universal* (language independent)
- ▶ Central question in this work:
 - Which constraint should we impose for better grammar induction across languages?

Previous work

- ▶ Many works incorporated *shorter dependency length* bias
 - Many dependency arcs are *short*



- Popular way is via initialization of EM (Klein and Manning, 2004)
 - used in most later approaches (Cohen and Smith (2009); Blunsom and Cohn (2010); Berg-kirkpatrick et al. (2010); etc)
 - Other work directly **parameterizes** length component
 - e.g., Smith and Eisner (2005); Mareček and Žabokrtský (2012)

This work

- ▶ We explore the utility of *center-embedding avoidance* in languages
- ▶ Languages tend to avoid nested, or center-embedded structures
 - because it is difficult to comprehend for human

ex:

The reporter *who the senator* *who Mary met* *attacked* *ignored the president*

- ▶ Intuition to our approach
 - Our model tries to learn grammars with less center-embedding
 - This is possible by formulating models on *left-corner parsing*

Contributions

- ▶ Learning method to avoid deeper center-embedding
 - We detect center-embedded derivations in a chart efficiently using left-corner parsing
- ▶ Application to dependency grammar induction
 - We focus on dependency grammar induction since it is the most widely studied task
- ▶ Experiments on many languages in Universal Dependencies
 - We find that our approach shows *different tendencies than the dependency length-based constraints*
 - We give an analysis of this difference to characterize our approach

Approach and Model

Approach overview

- ▶ We assume a *base* generative model for dependency trees

$$p_{base}(\overset{\curvearrowright}{a} \overset{\curvearrowright}{dog} barks) = 0.023$$

- ▶ We constraint the model by multiplying a penalty factor f

$$p(t) = p_{base}(t) \times f(t)$$

- ▶ One such f that penalizes center-embedding is:

$$f(t) = \begin{cases} 0 & \text{if } t \text{ contains degree } \geq 2 \text{ center-embedding} \\ 1 & \text{else} \end{cases}$$

- ▶ Smith and Eisner (2005) is the same approach with different f

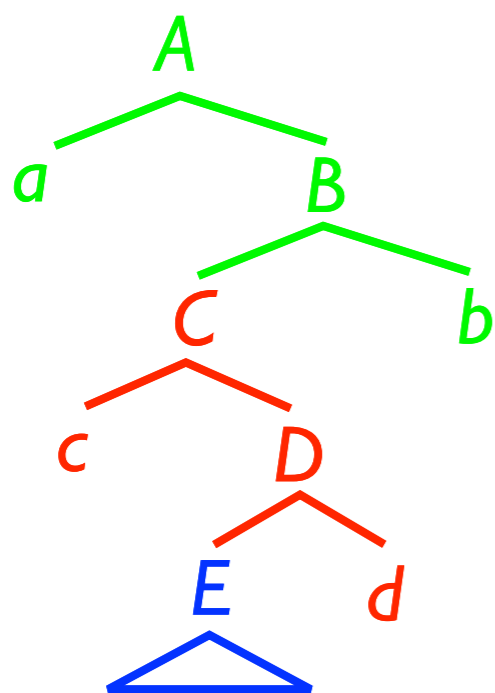
- ▶ We only add a constraint during learning (EM)

- **Challenge:** how to efficiently compute f during EM in a chart?

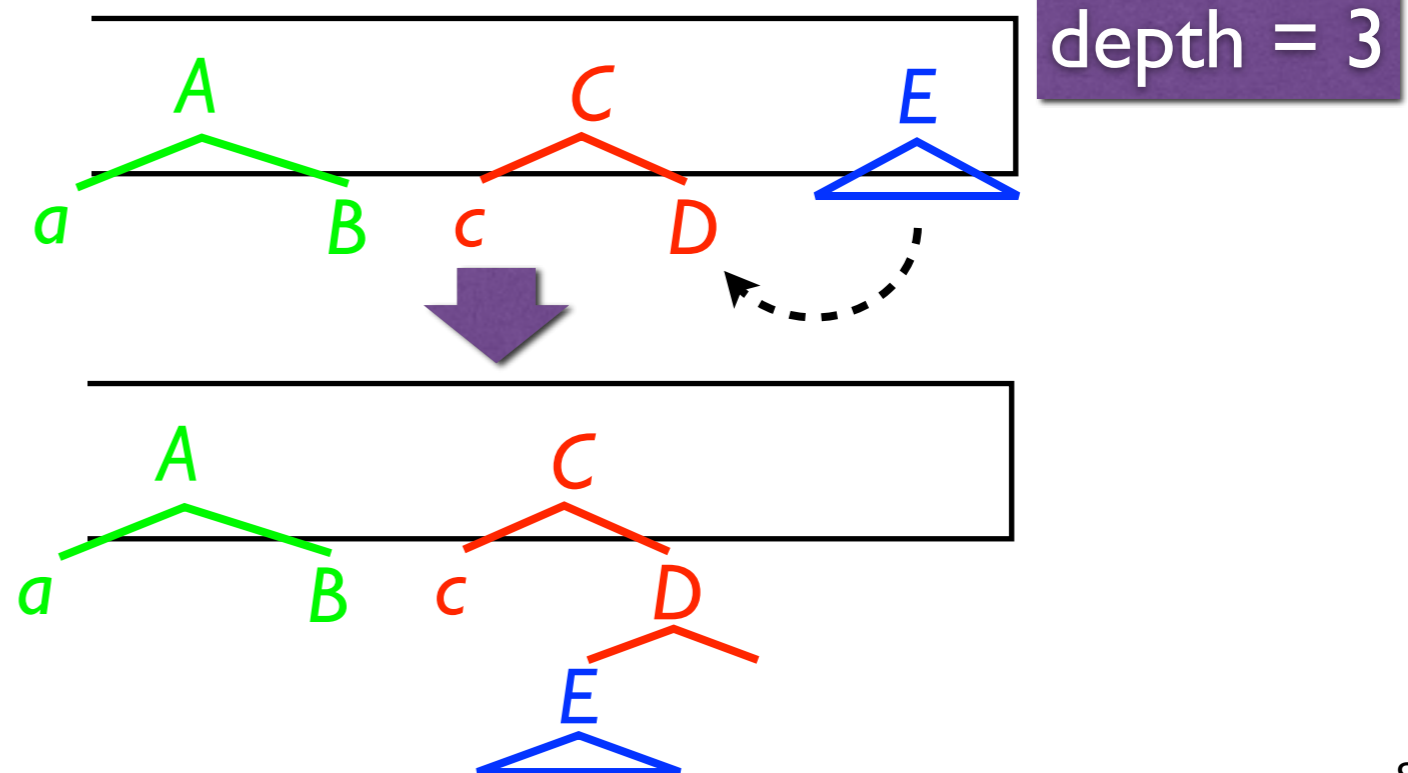
Key tool: left-corner parsing

- ▶ There are several variants in left-corner parsing
 - We use one particular method by Schuler et al. (2010)
- ▶ A parsing algorithm on a stack
 - The stack size grows only when processing center-embedding
 - **Stack depth = (degree of center-embedding) + 1**

A degree-2 embedded tree

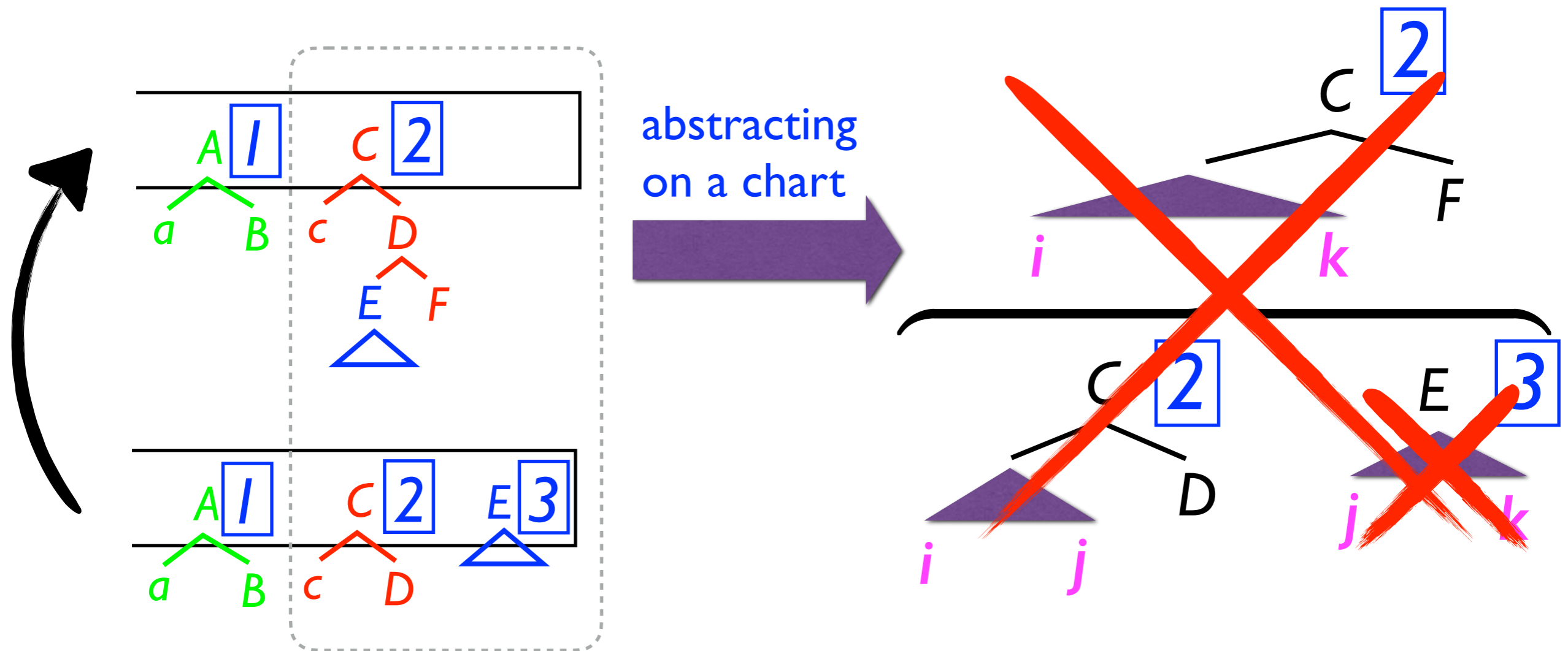


Following configuration occurs for this tree



EM on left-corner parsing

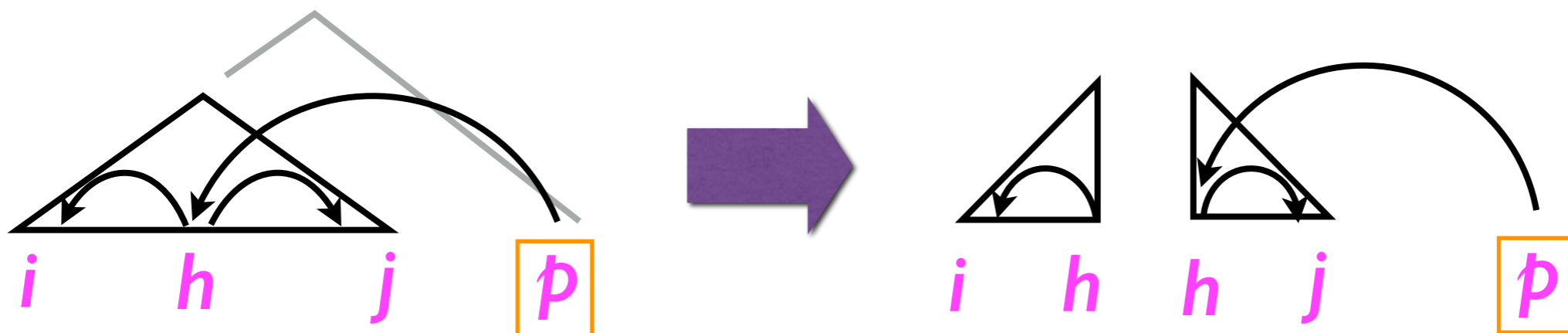
- ▶ Idea: we *keep the current stack depth* of left-corner parsing *in each chart item* in inside-outside



- ▶ When we prohibit degree ≥ 2 center-embedding, the above rule is eliminated

Applying to dependency grammar induction

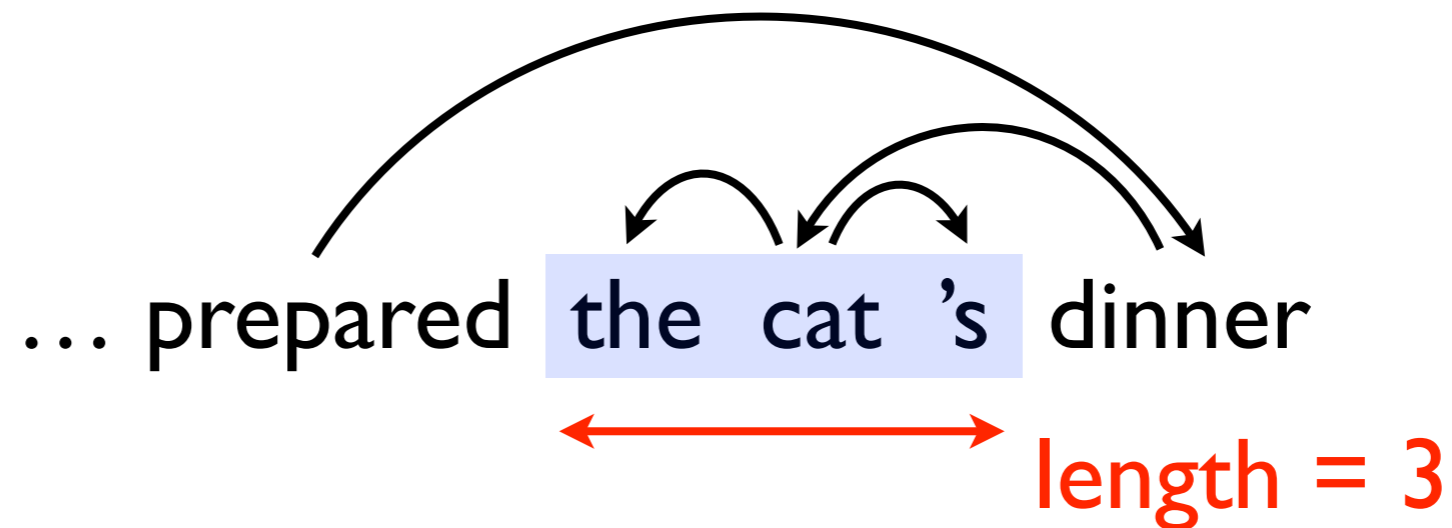
- ▶ The technique is quite general, and can be applied to *any models on PCFG*
- ▶ We apply the technique into DMV (Klein and Manning, 2004)
 - The most popular generative model for grammar induction
 - Since DMV can be formulated as a PCFG, we can apply the idea
- ▶ The time complexity of the naive implementation is $O(n^6)$ due to the need to remember *additional index*
 - We can improve it to $O(n^4)$ using head-splitting



Span-based constraints

- ▶ Motivation: many occurrences of center-embedding are due to embeddings of *small chunks*, not clauses

Example



- ▶ We will try the following constraints in experiments

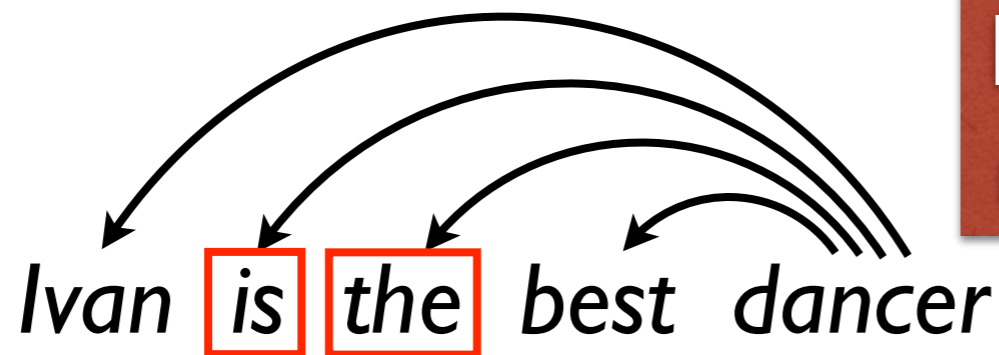
$$f(t) = \begin{cases} 0 & \text{if } t \text{ contains embedded chunk of length } > \delta \\ 1 & \text{else} \end{cases}$$

- ▶ This can be done by changing (relaxing) the condition of increasing stack depth

Experiments

Universal Dependencies (UD)

- ▶ We use UD in our experiments (v. 1.2)
- ▶ Characteristics:
 - all languages are annotated with the *content-head* style



In principle, function words never have a child in a tree

- ▶ Some settings:
 - 25 languages in total (remove small treebanks)
 - The inputs are universal POS tags
 - Training sentence length ≤ 15
 - Test sentence length ≤ 40

Evaluation is difficult in grammar induction

- ▶ Issue on previous grammar induction research:
 - The annotation styles of the gold treebank differ across languages (e.g., auxiliary head vs. main verb head)
 - This obscures the contribution of a constraint in each language
- ▶ Our evaluation setting to mitigate this issue:
 - We use UD to best guarantee the consistencies across languages
 - All models take the following additional constraint

$$f(t) = \begin{cases} 0 & \text{if a function word has a child on } t \\ 1 & \text{else} \end{cases}$$

- This guarantees that all outputs will follow the UD-style annotation

Models (constraints)

- ▶ All models are formulated as $p_{\text{DMV}}(t) \times f(t)$
- ▶ Only differences between models are f (at training)
 - **FUNC**: Baseline (function word constraint only)
 - **DEPTH**: In addition to FUNC, set the maximum stack depth
 - **ARCLLEN**: Equivalent to Smith and Eisner (2005), a soft bias to favor shorter dependency arcs

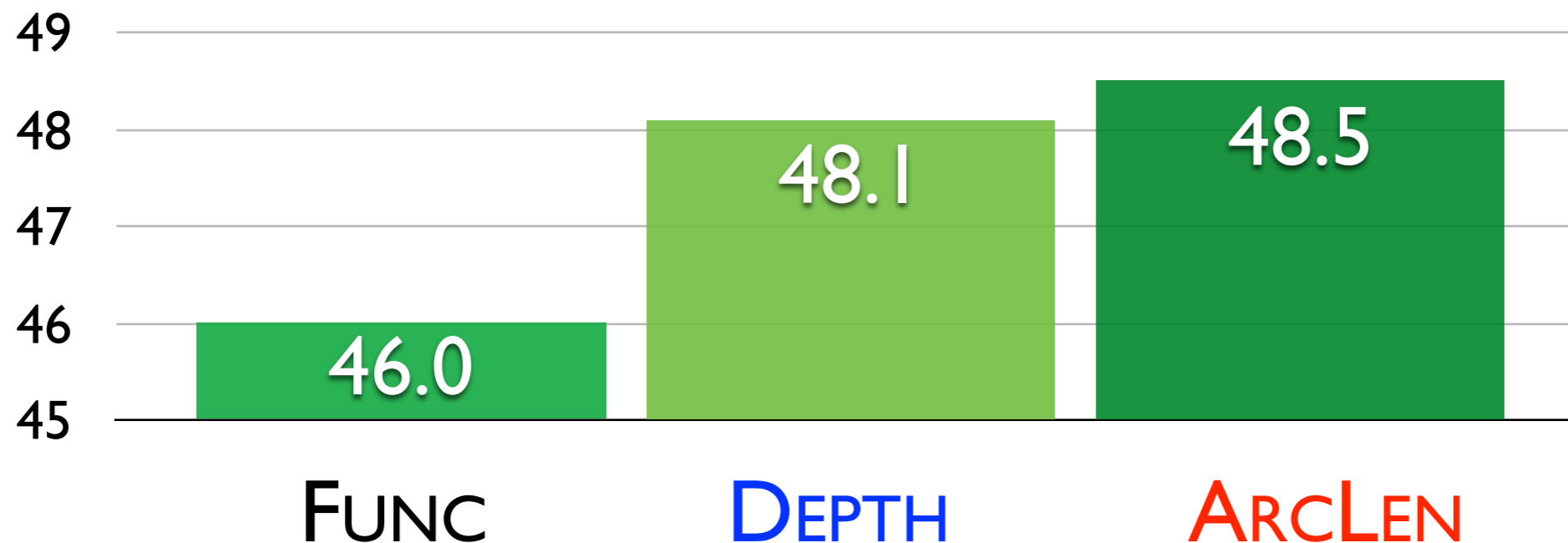


- ▶ We initialize all models uniformly
 - We found harmonic initialization does not work well

UD summary

- ▶ For **DEPTH**, which maximum stack depth should we use?
 - We use (UD-style) English WSJ as a development set
 - *NOTE: English data in UD is not WSJ, but Web treebank*
 - The best setting is *allowing embedded chunks of length ≤ 3*

Average scores across 25 languages (UAS)

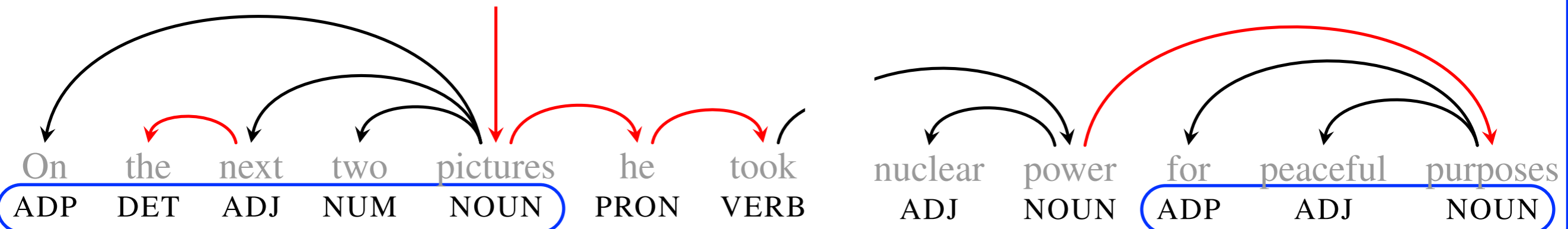


DEPTH improves scores but is slightly less effective than **ARCLLEN**

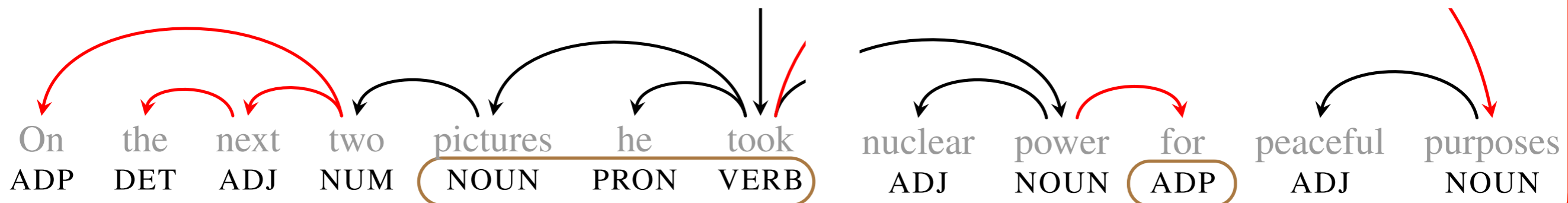
Analysis on English

- ▶ Average scores are similar, but is there any characteristics in each constraint?
- We found an interesting difference in English data (Web)

DEPTH: good at detecting *constituent boundaries*



ARCLLEN: good at detecting **VERB** → **NOUNs**, but bad at constituents

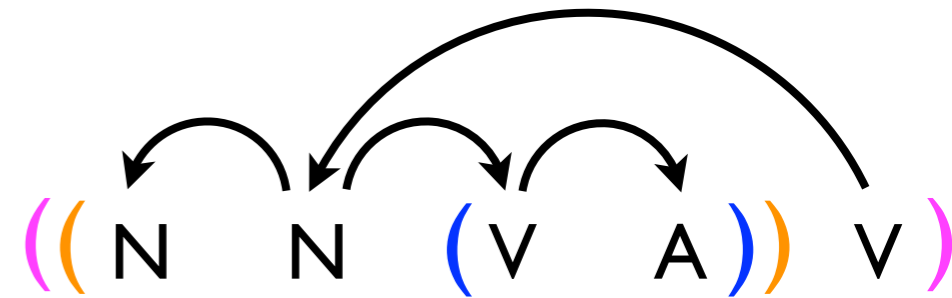


Bracket scores

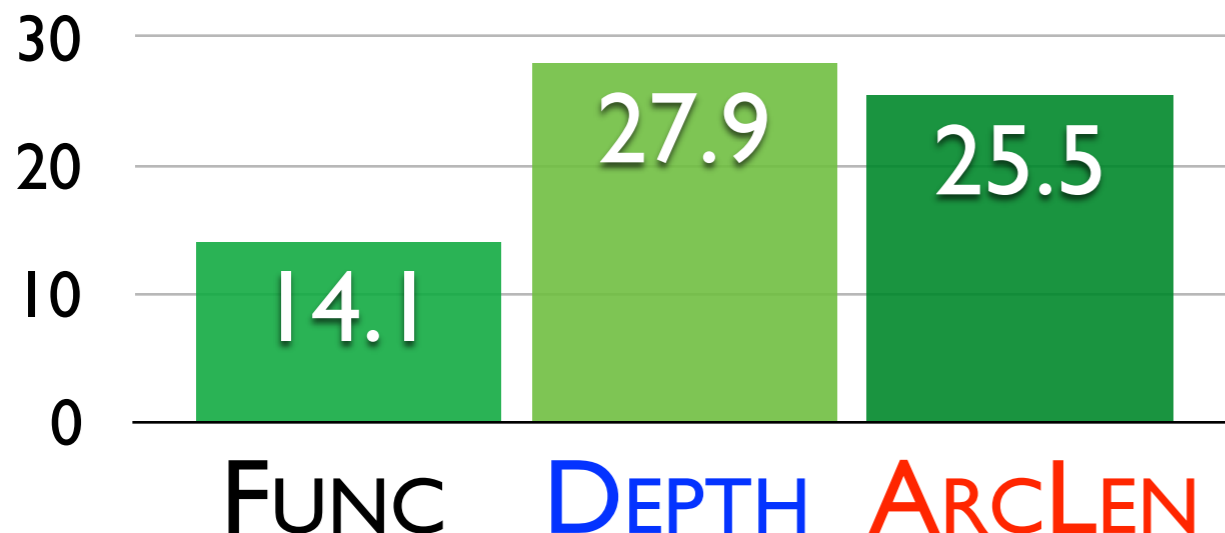
- ▶ Hypothesis: **DEPTH** is better at finding *correct constituent boundaries* in language than **ARCLLEN**
 - ... possibly because avoiding center-embedding is essentially a constraint to constituents (?)

▶ Quantitative study:

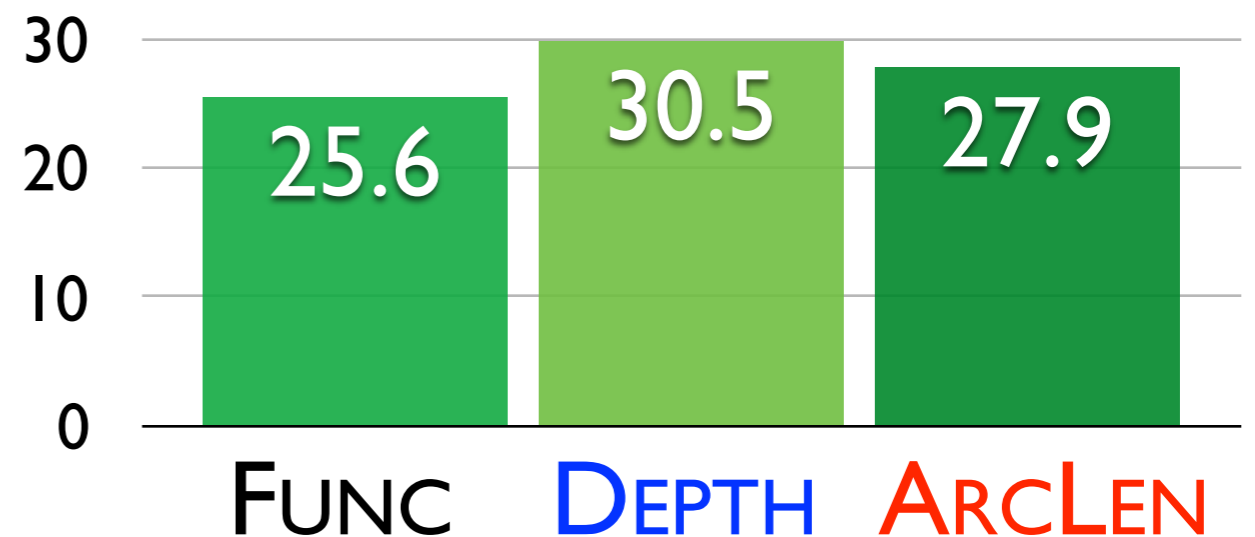
- We extract *unlabelled* brackets from gold and output trees and calculate F1 score



English:



Average:

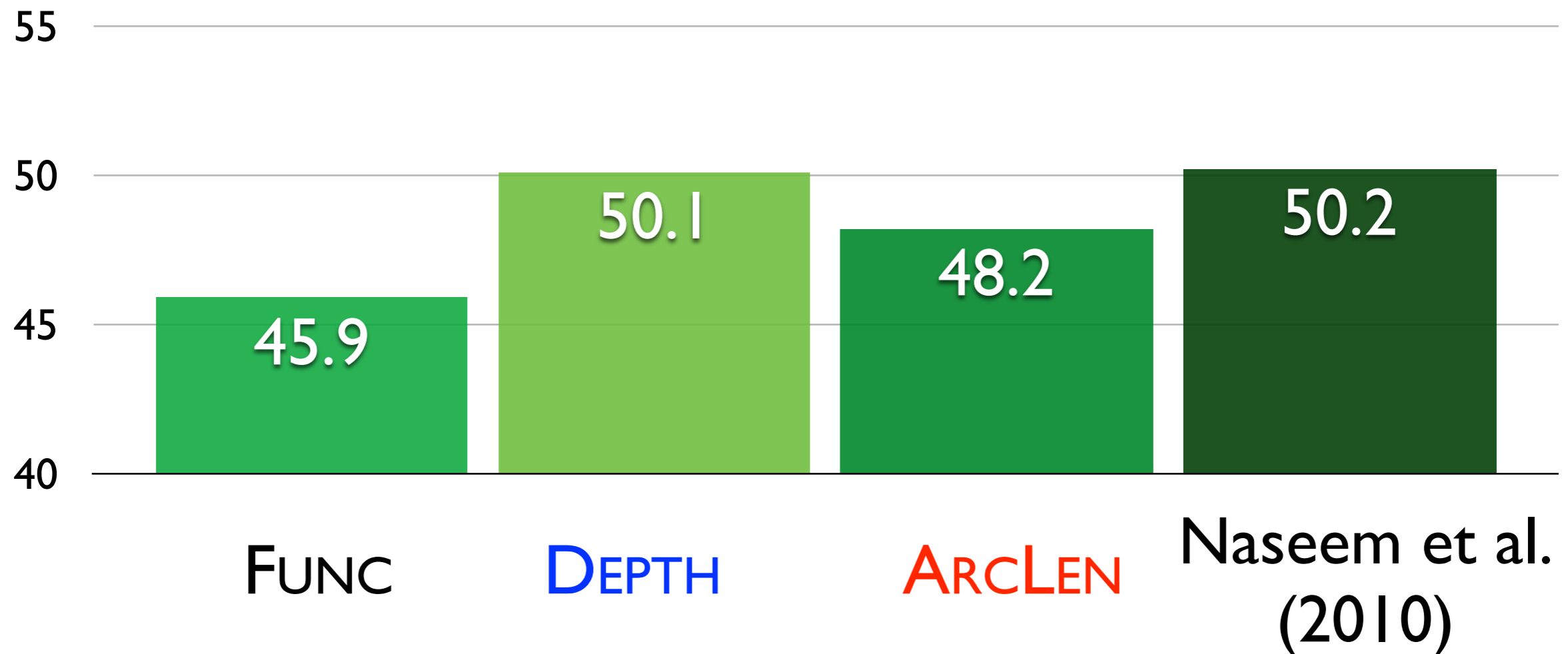


Adding constraints to the sentence root

- ▶ Results so far suggest **DEPTH** itself cannot resolve some core dependency arcs, e.g., VERB → NOUNs
- ▶ Recent state-of-the-art systems rely on additional constraints, e.g., on root candidates (Bisk and Hockenmaier, 2013; Naseem et al, 2010)
- ▶ We follow this, and add the following constraint in all models
 - **The sentence root must be a VERB or a NOUN**

Results with the root constraint

Average UAS



- **DEPTH** works the best when the root constraint is added
- Competitive with Naseem et al. (2010), which utilizes much richer prior linguistic knowledge on POS tags

Conclusion

- ▶ Main result: avoiding center-embedding is a good constraint in grammar induction
 - In particular, it helps to find linguistically correct constituent structures, probably because it is the constraint on constituents
- ▶ Future work:
 - Grammar induction beyond dependency grammars
 - including traditional constituent structure induction, which has been failed due to the lack of good syntactic cues
 - Weakly-supervised grammar induction, e.g., Garrette et al. (2015)

Thank you!