Jason Eisner

Jiarong Jiang

He He

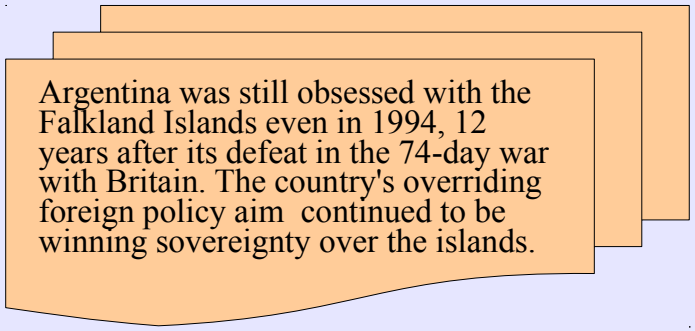
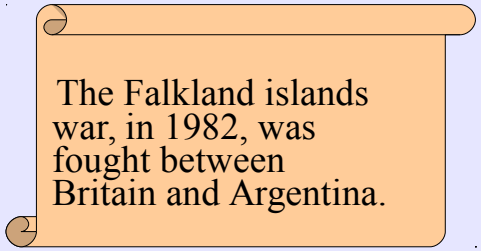
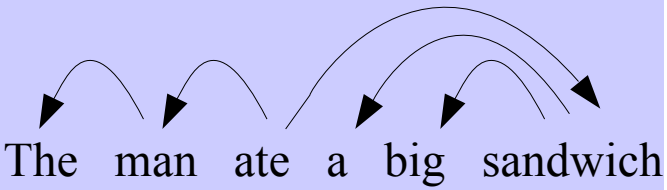
Better! Faster! Stronger*!

Learning to balance accuracy and efficiency when predicting linguistic structures

(*theorems)

Hal Daumé III | UMD CS, UMIACS, Linguistics | me@hal3.name | @haldaume3

NLP as transduction

Task	Input	Output
Machine Translation	Ces deux principes se tiennent à la croisée de la philosophie, de la politique, de l'économie, de la sociologie et du droit.	Both principles lie at the crossroads of philosophy, politics, economics, sociology, and law.
Document Summarization		
Syntactic Analysis	The man ate a big sandwich.	
...many more...		

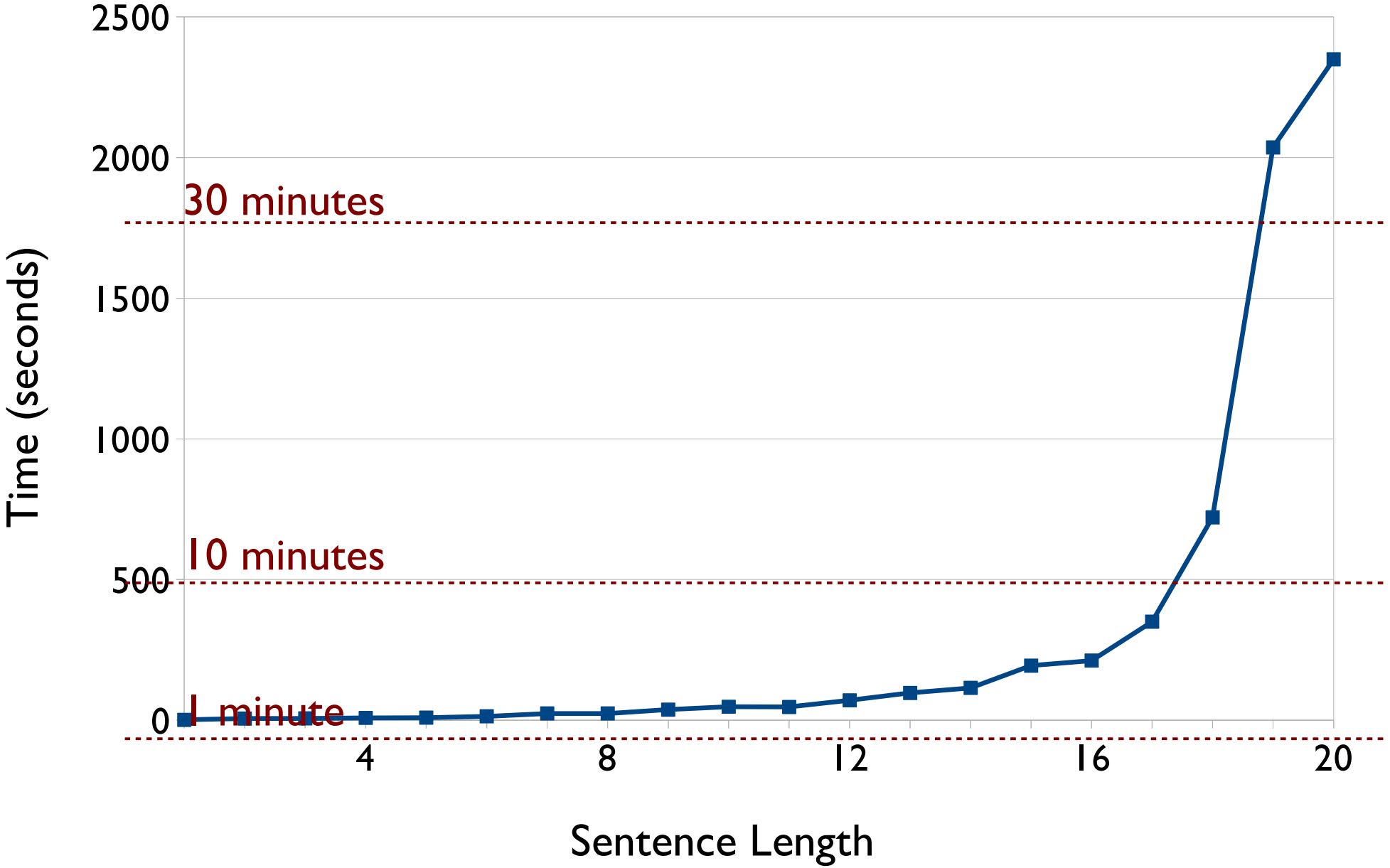
Why are complex predictions slow?

- Parsing # trees ~ $O(2^{|\text{sentence}|})$
- Translation # trans ~ $O(2^{|\text{foreign}|x|\text{english}|})$
- Summarization # sums ~ $O(2^{|\text{document}|})$

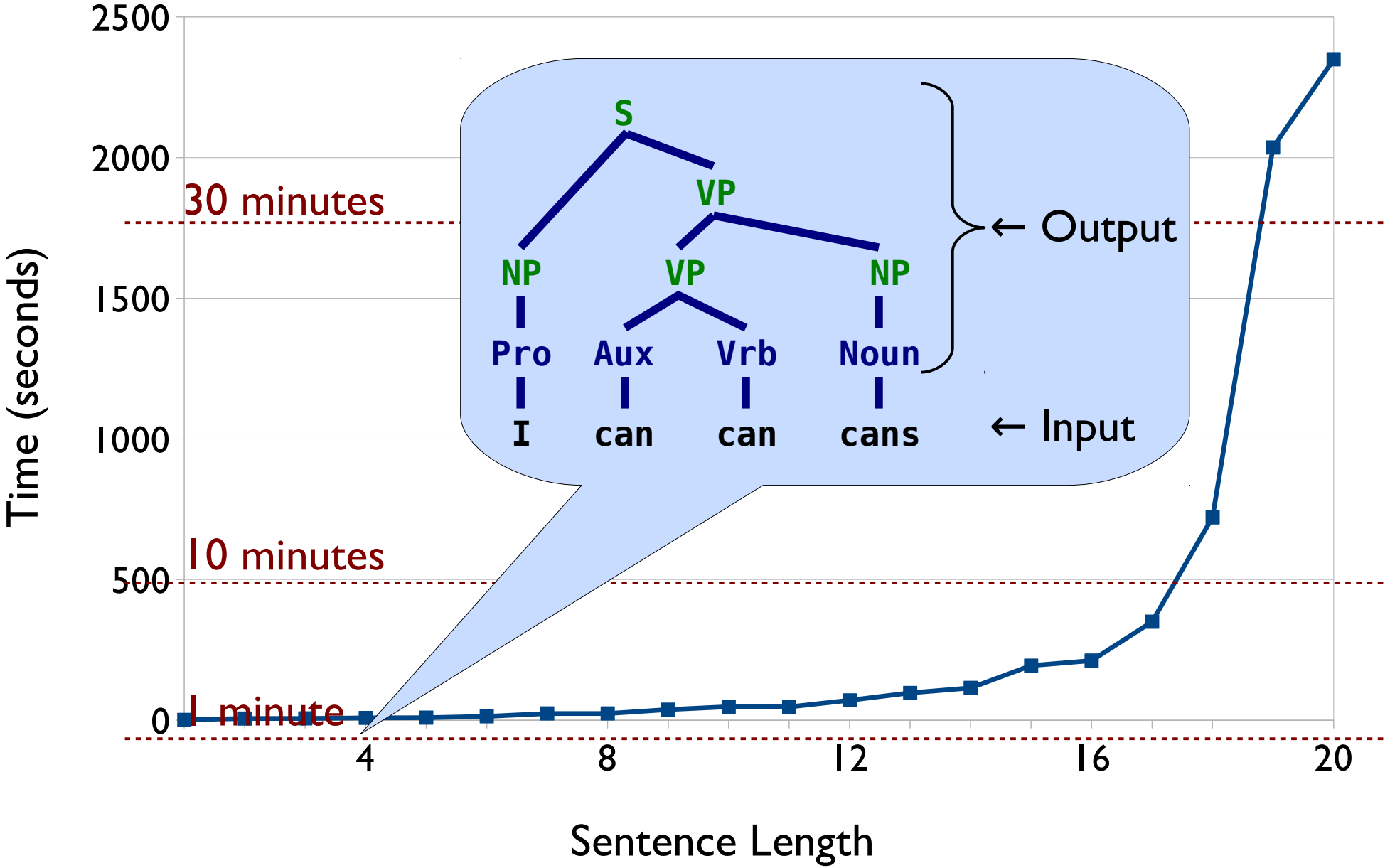
Concretely, $n^3 = 15000$
|grammar| = ½ million

- What about dynamic programming....
  - Often not possible (features are too complicated)
  - Even when possible, polynomial-time is too painful:
    Parsing is $O(|\text{grammar}| \times |\text{sentence}|^3)$
    but |grammar| is HUG

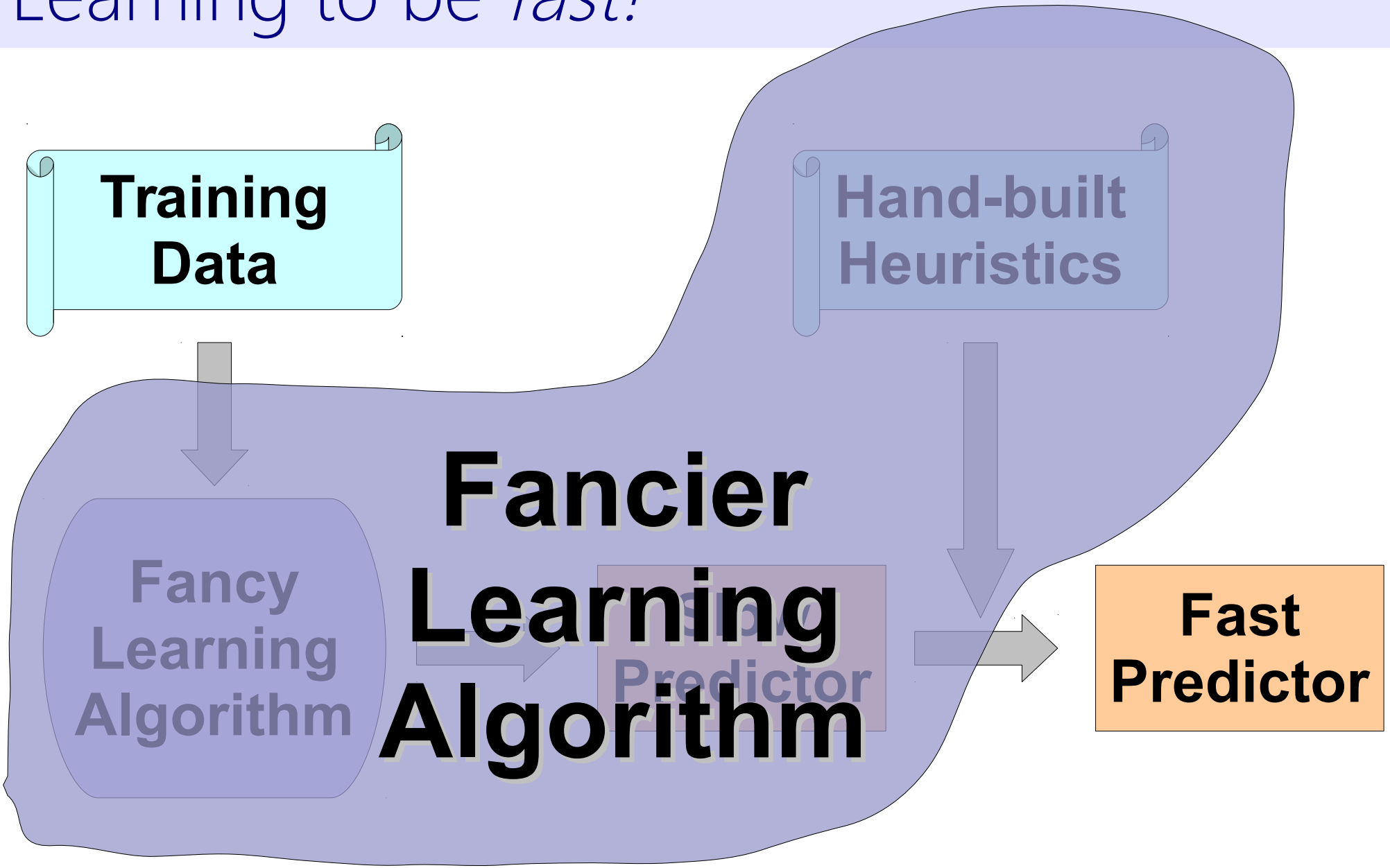
Case study: prioritized parsing



Case study: prioritized parsing

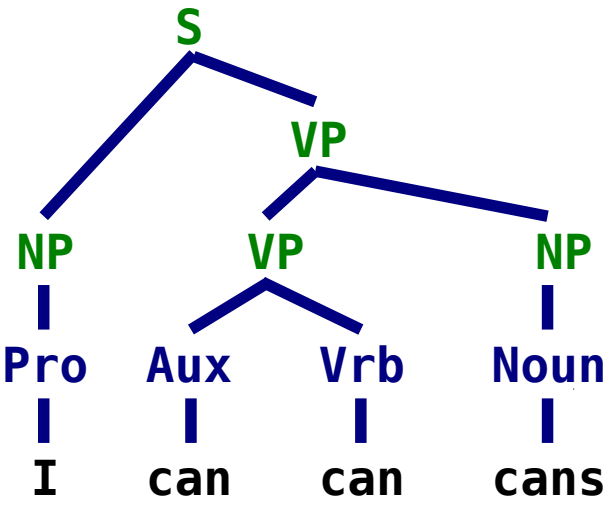


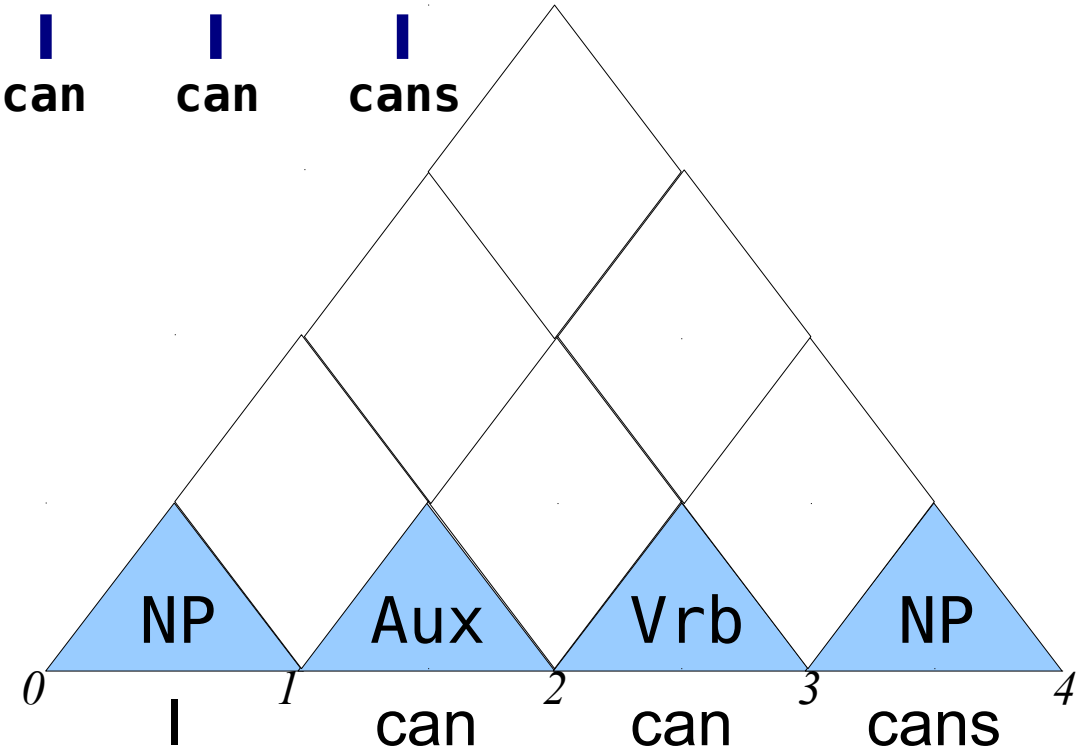
Learning to be *fast!*



Quality = tradeoff(accuracy, time)

Prioritized search (eg., parsing)

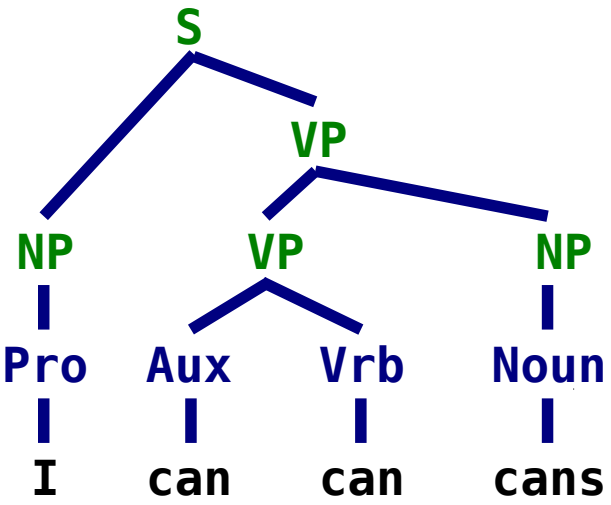


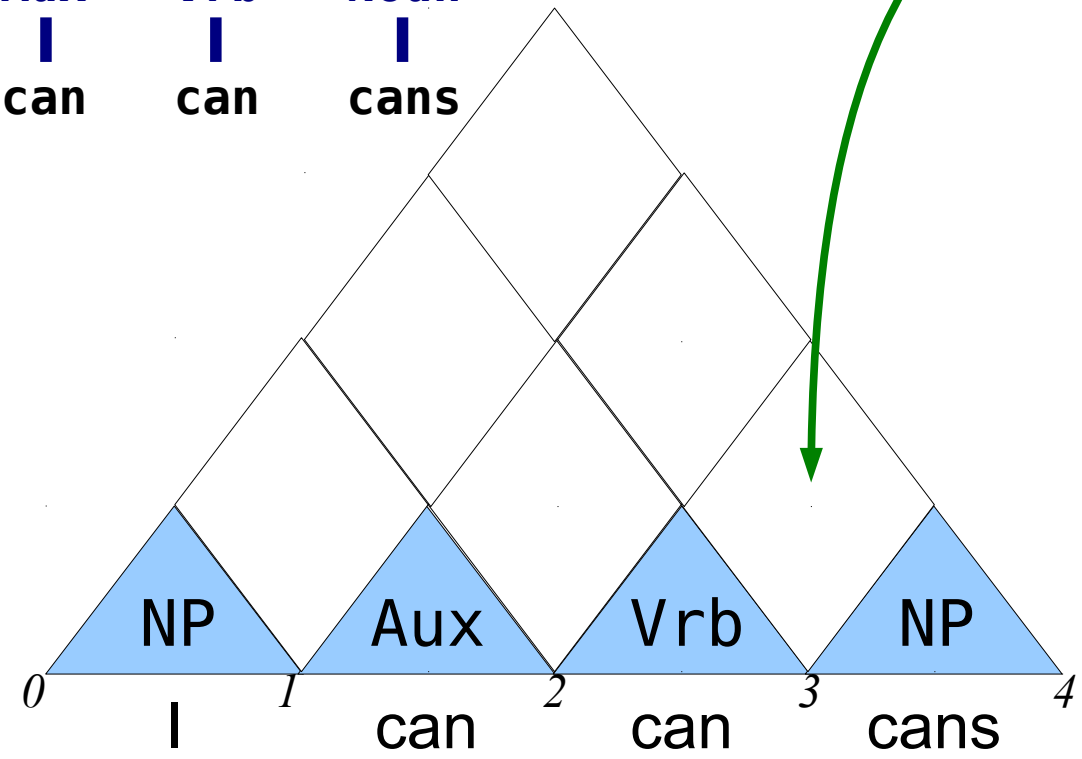


7 VP[2,4]
2 VP[1,3]
1 S[0,2]

(Jiang+Teichert+Eisner+D, NIPS 2012)

Prioritized search (eg., parsing)

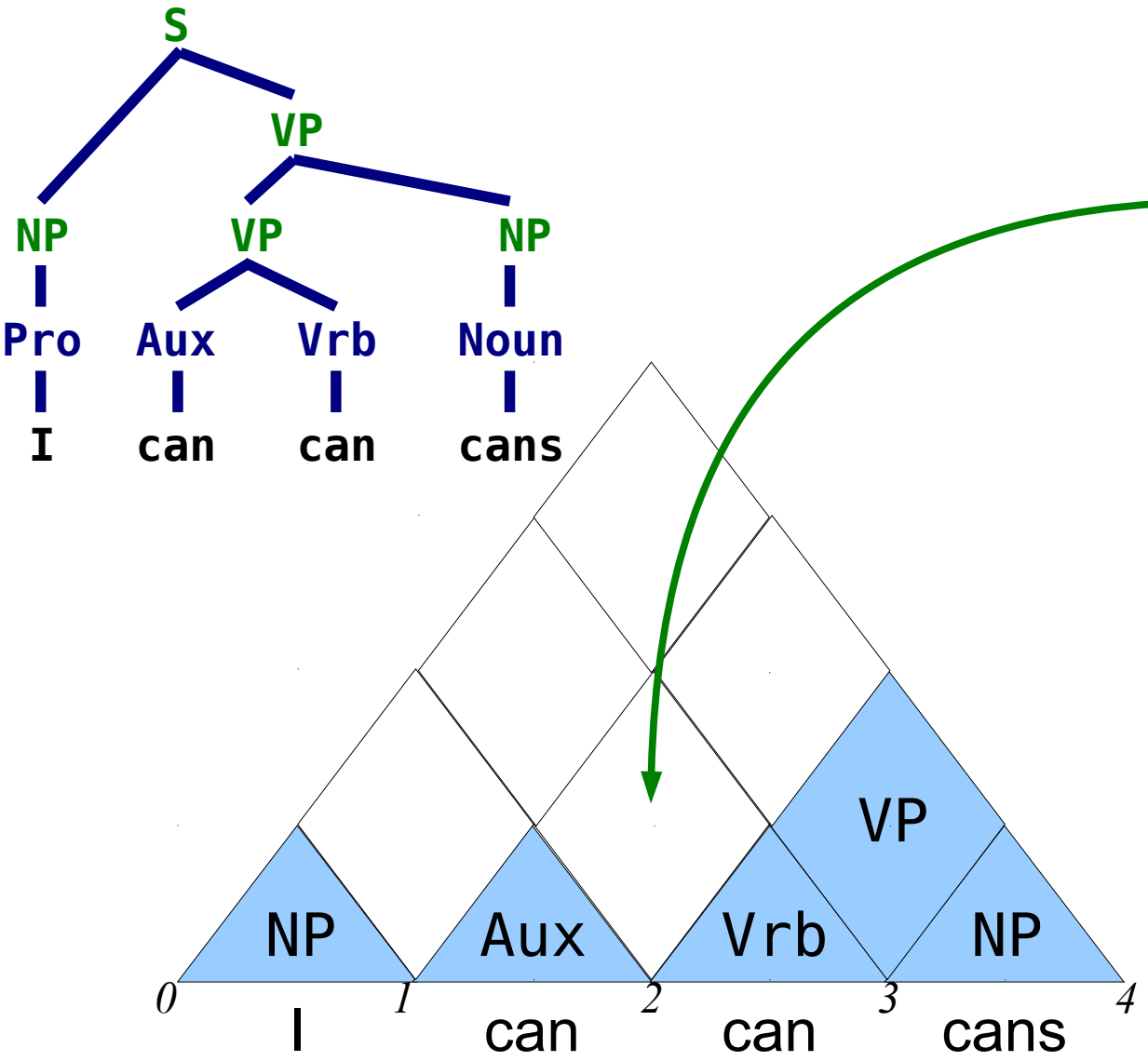




7 VP[2,4]
2 VP[1,3]
1 S[0,2]

(Jiang+Teichert+Eisner+D, NIPS 2012)

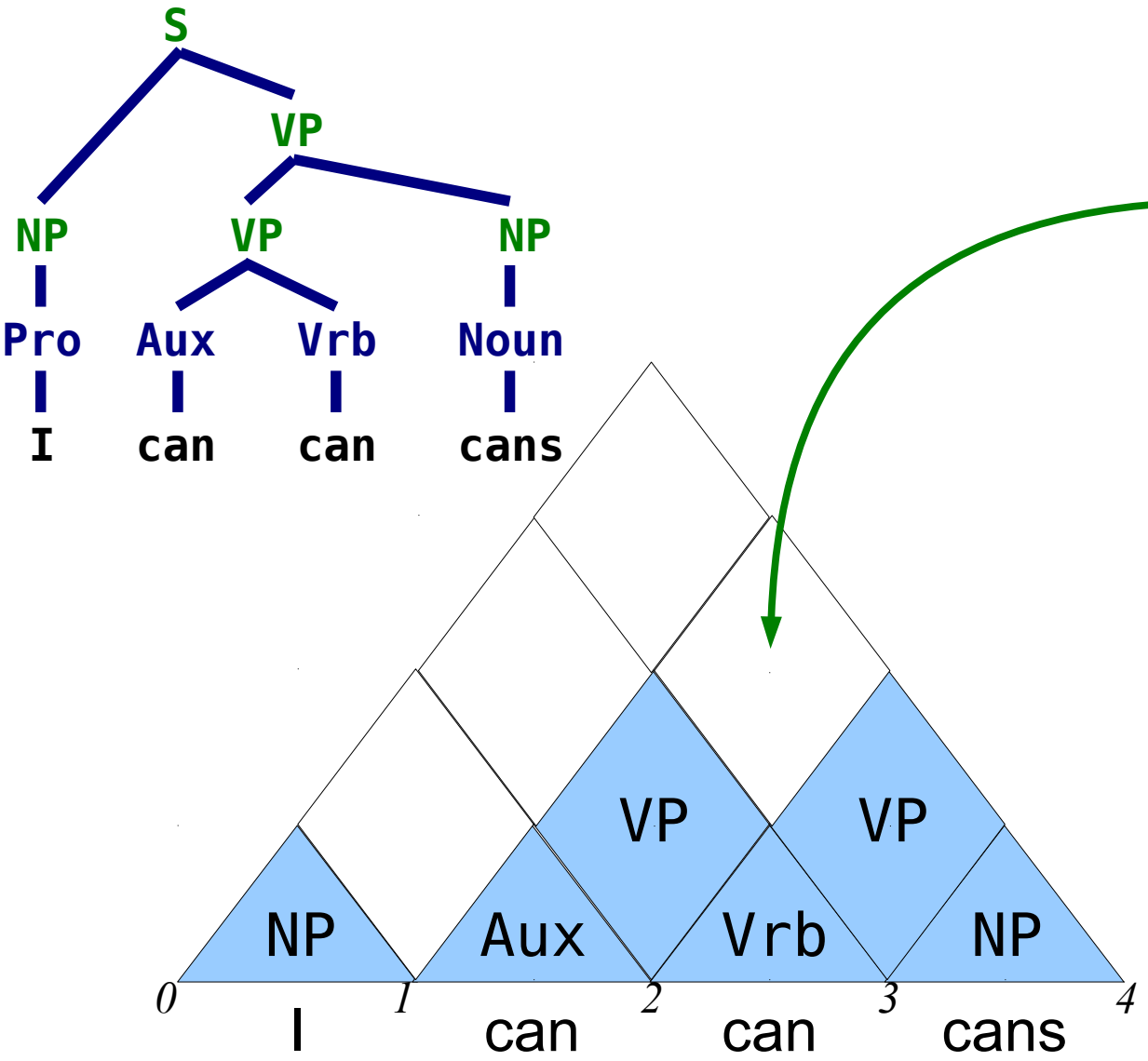
Prioritized search (eg., parsing)

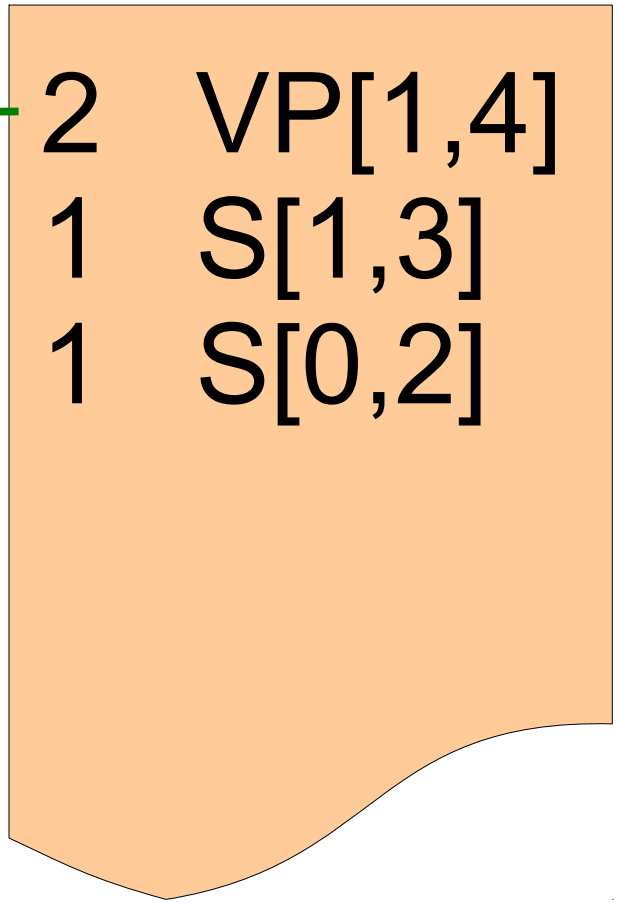


2 VP[1,3]
2 VP[1,4]
1 S[0,2]

(Jiang+Teichert+Eisner+D, NIPS 2012)

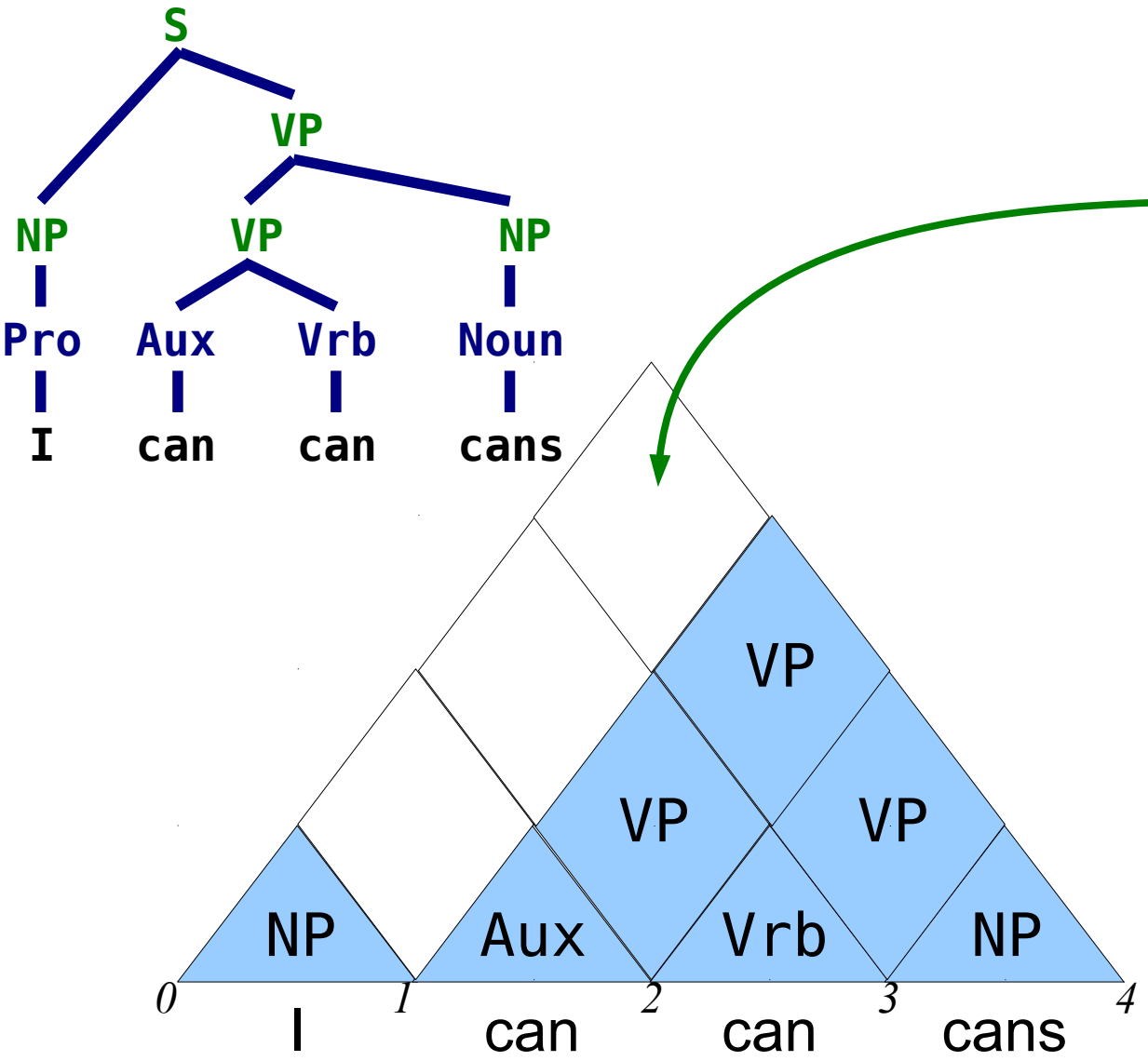
Prioritized search (eg., parsing)

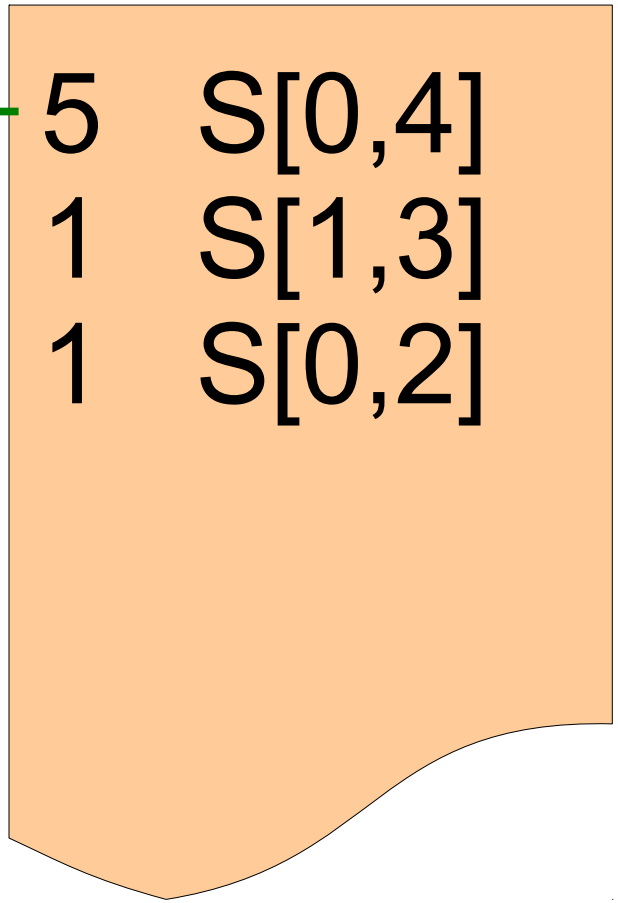




(Jiang+Teichert+Eisner+D, NIPS 2012)

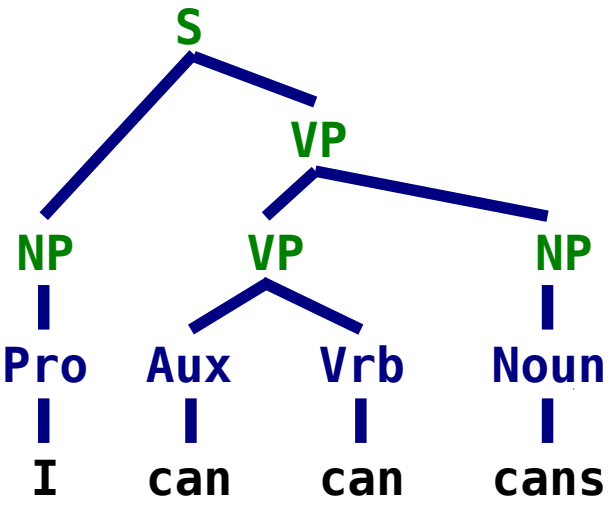
Prioritized search (eg., parsing)

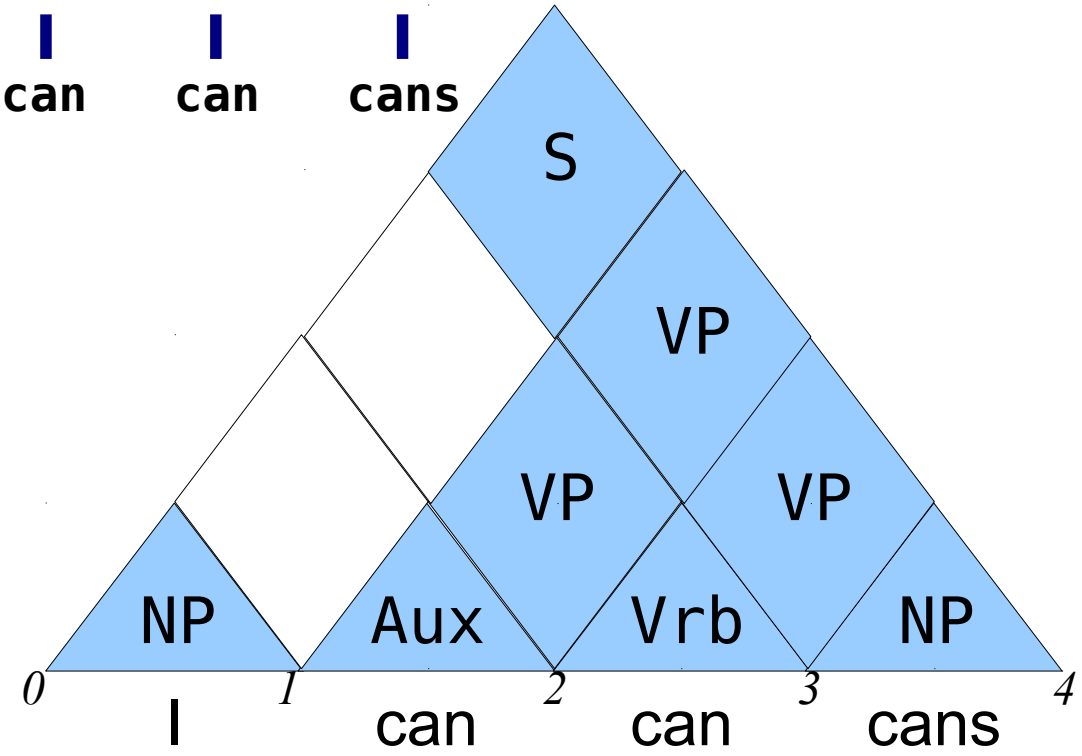




(Jiang+Teichert+Eisner+D, NIPS 2012)

Prioritized search (eg., parsing)





9 stop
1 S[1,3]
1 S[0,2]

(Jiang+Teichert+Eisner+D, NIPS 2012)

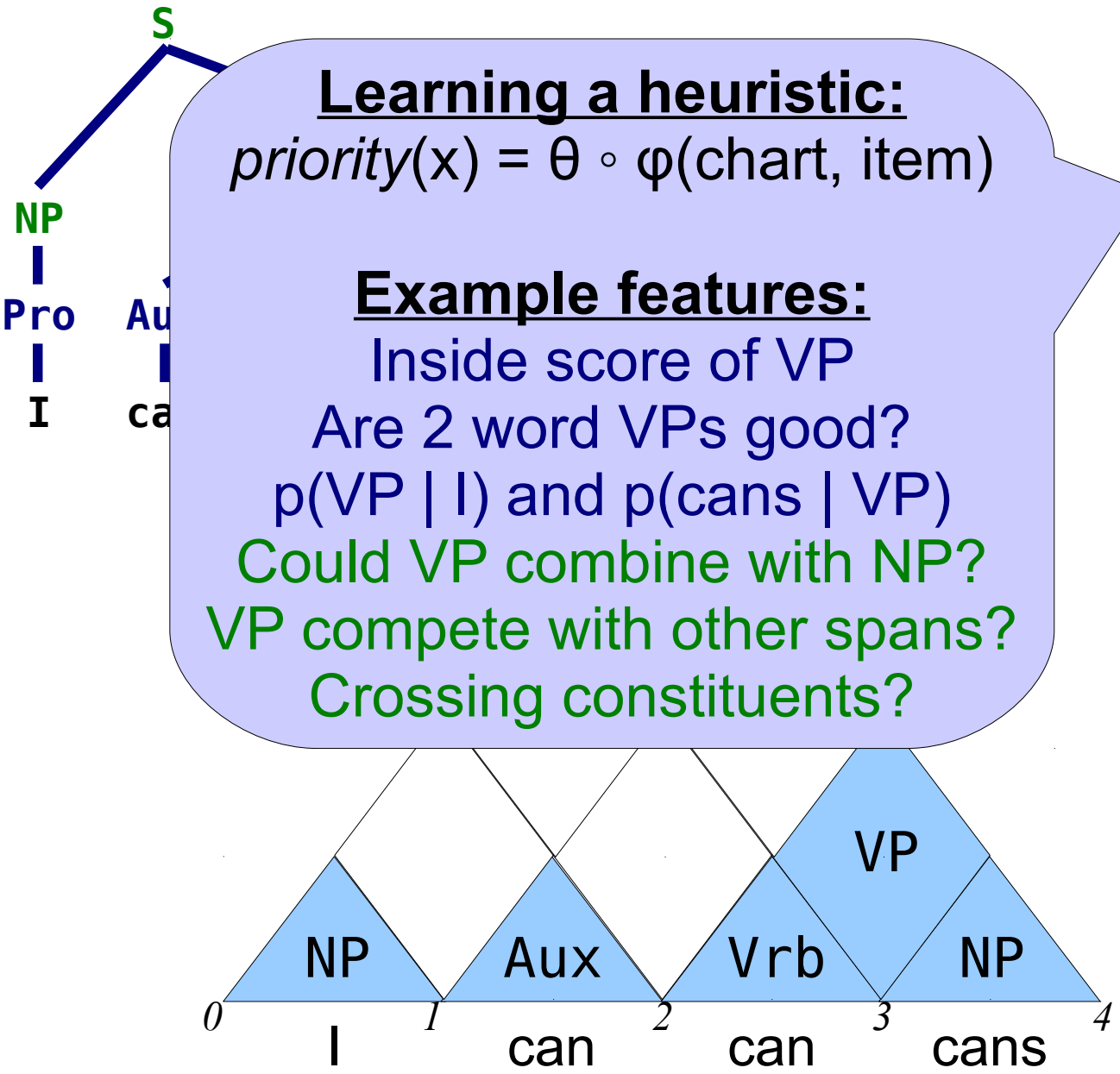
What do better priorities buy us?

- Ideally, run until queue is empty
- Ordering only matters because you want to:
  - Only pop items that have a real impact on result
  - Avoid premature pops
  - Stop early
- Goal: learn a priority function that optimizes accuracy/speed trade-off
- Typical solution: hand-built heuristics
- Our solution: learning to optimize *accuracy* - $\lambda \times$ *time*

User parameter

(Jiang+Teichert+Eisner+D, NIPS 2012)

Prioritized search (eg., parsing)



(Jiang+Teichert+Eisner+D, NIPS 2012)

Learning priorities

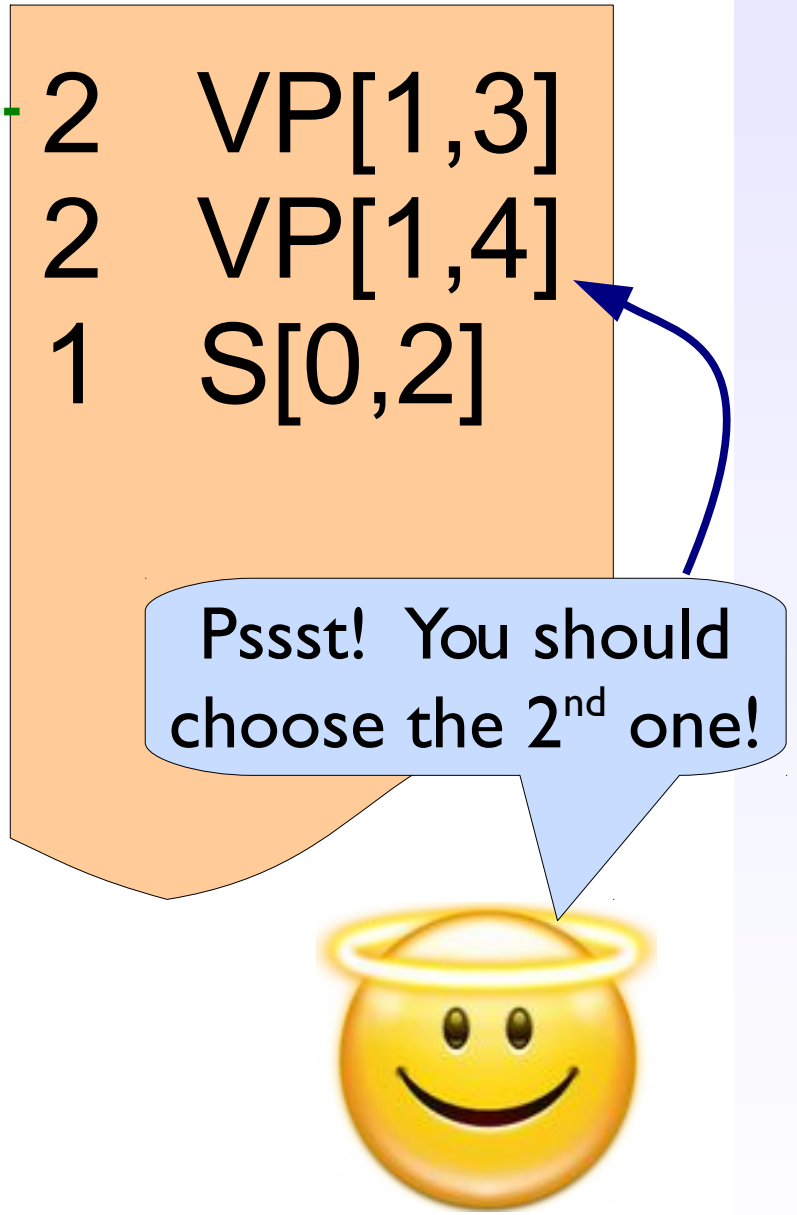
Challenges:

The world is non-deterministic

State space is huge
(25 words → 5×10^{14} states)

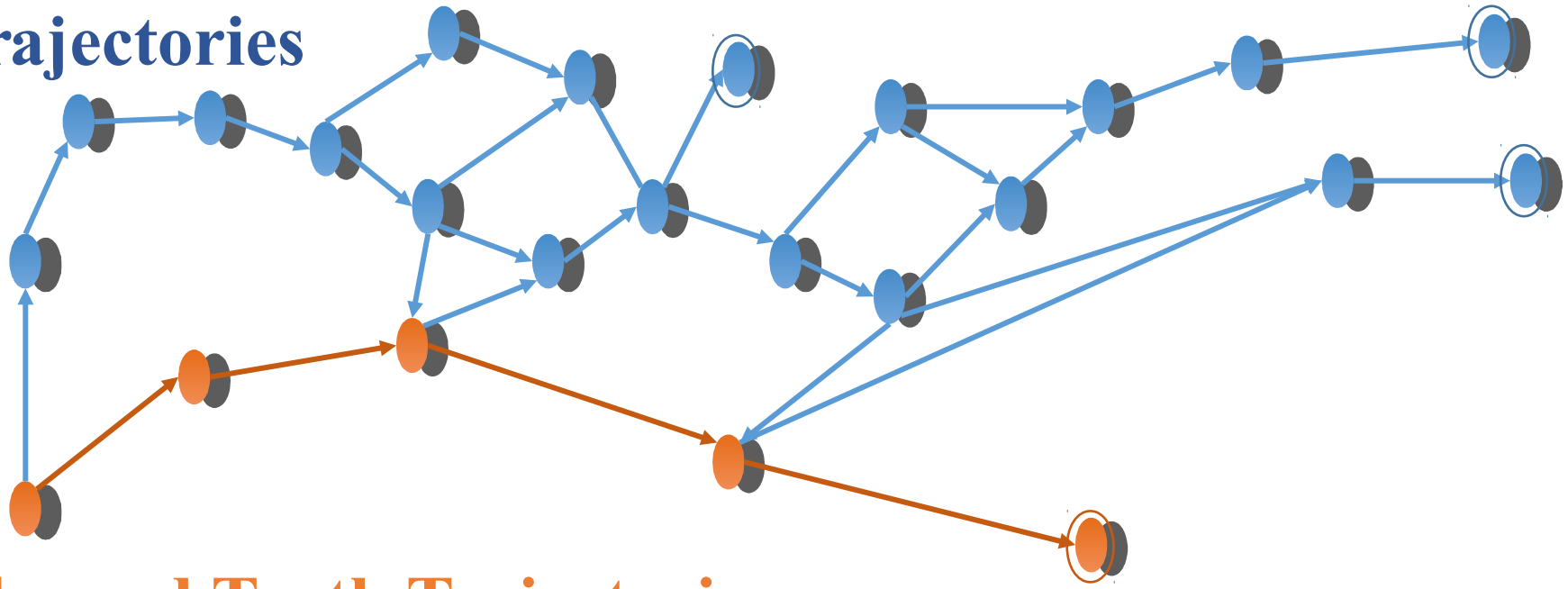
The oracle is way too good
(25 actions vs 25k actions)

The oracle does not experience a trade-off between speed and accuracy



Parsing Trajectories (I)

Policy Gradient Trajectories



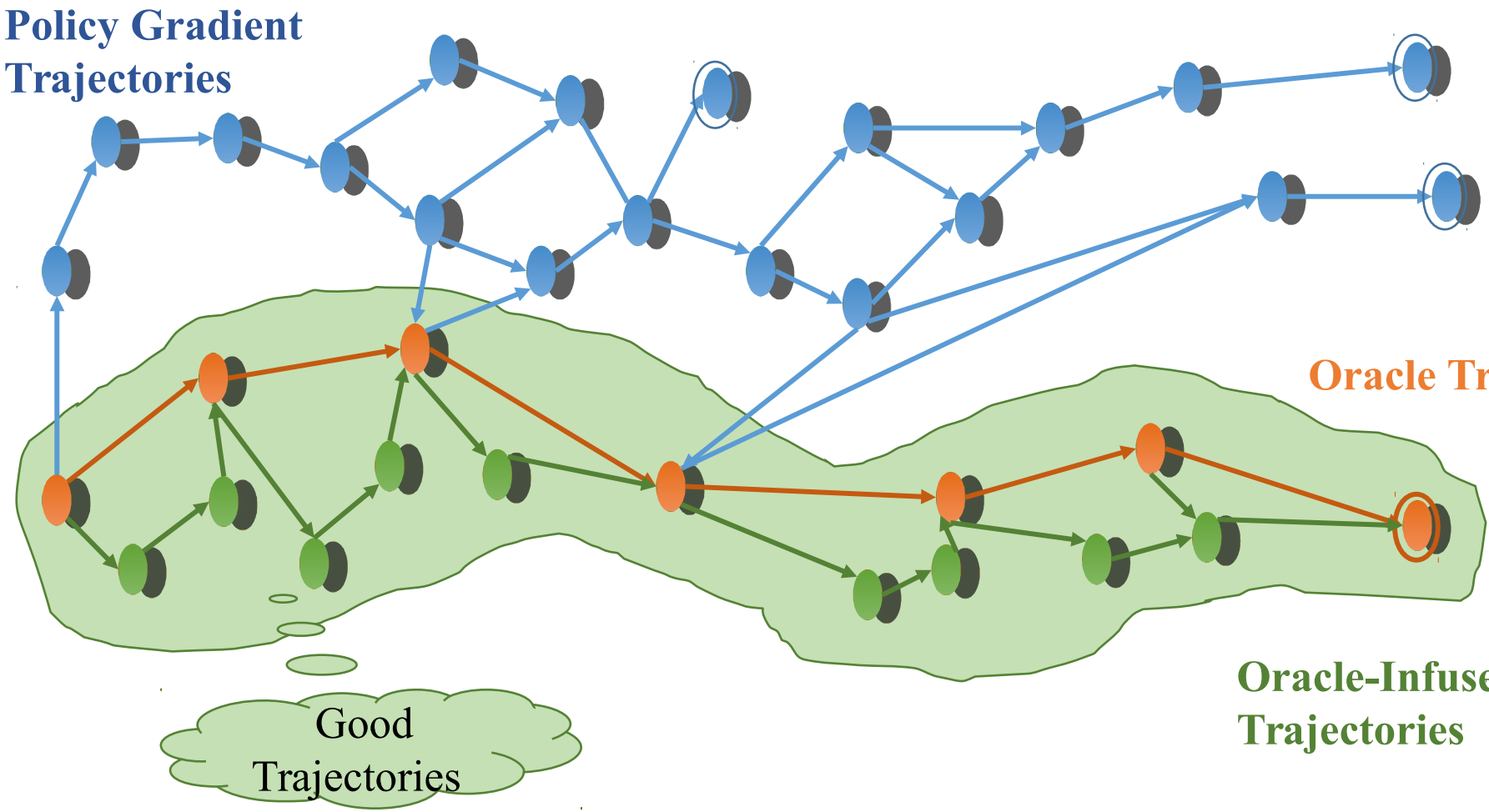
Ground Truth Trajectories (Oracle Trajectories)

Preliminary Result

Method	Recall	Relative # of pops
Apprenticeship Learning	**84.2**	**0.85x**
with Reward Shaping	76.5	0.13x
Policy Gradient with Boltzmann Exploration	56.4	0.46x
Uniform cost search	93.3	1.0x
Pruned Uniform cost search	92.0	0.33x

Failure Causes:
Too hard to imitate the oracle with our features!

Parsing Trajectories (II)



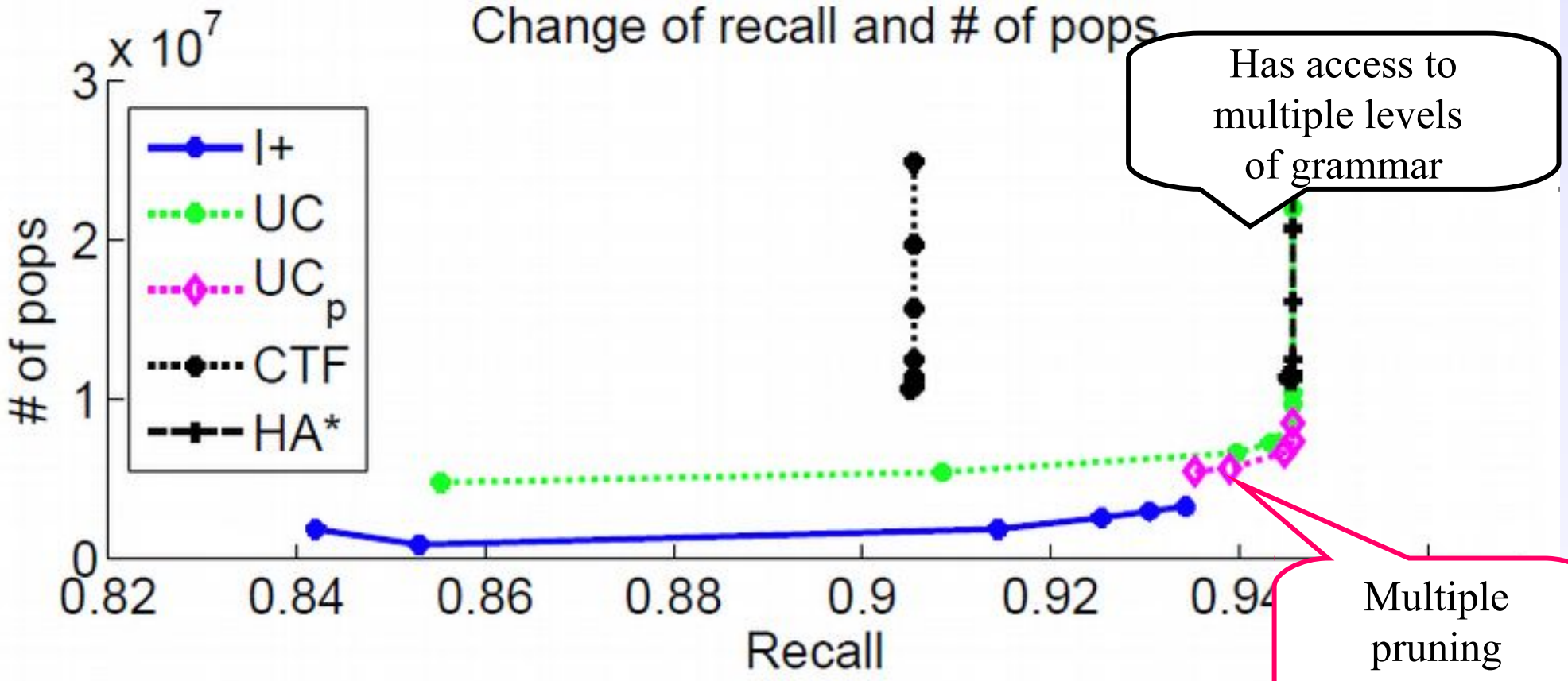
Oracle-Infused Policy Gradient

- Goal: "interleaving" oracle actions with policy actions both feasible and sensible
- Let $\delta \in [0,1]$. The oracle-infused policy

$$\pi_\delta^+(a|s) = \delta\pi^*(a|s) + (1-\delta)\pi_t(a|s)$$

Method	Recall	Relative # of pops
Oracle-Infused Policy Gradient	**91.2**	**0.46x**
Apprenticeship Learning	84.2	0.85x
with Reward Shaping	76.5	0.13x
Policy Gradient with Boltzmann Exploration	56.4	0.46x
Uniform cost search	93.3	1.0x
Pruned Uniform cost search	92.0	0.33x

Pareto Frontier



Pareto frontiers: Our **l+** parser at different values of λ, against the baselines at different pruning levels. Lower and further right is better.

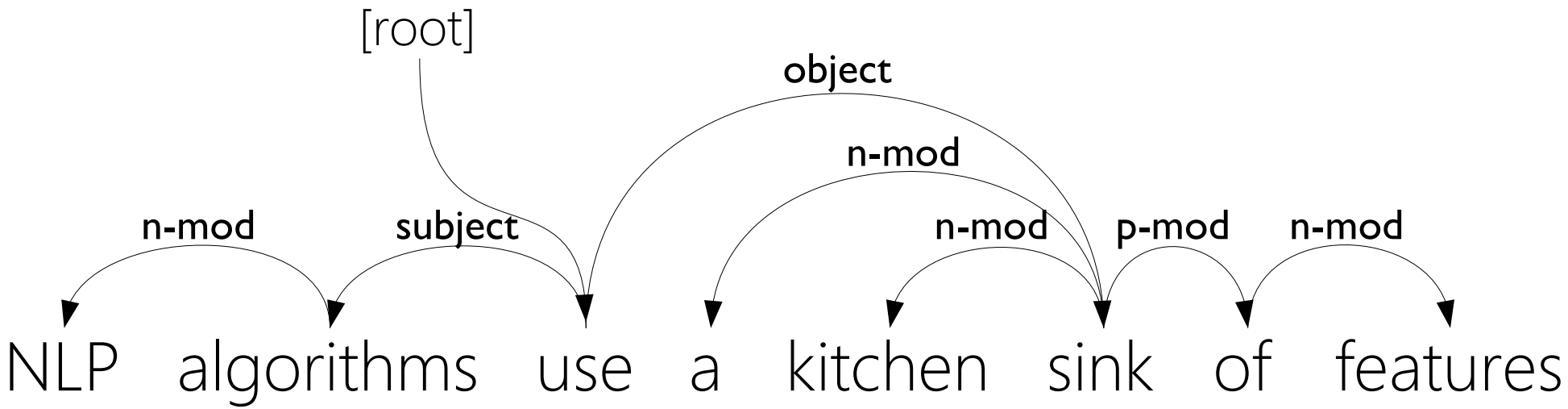
take away messages...

Many AI tasks can be cast as prioritized search (parsing, planning, inference...)

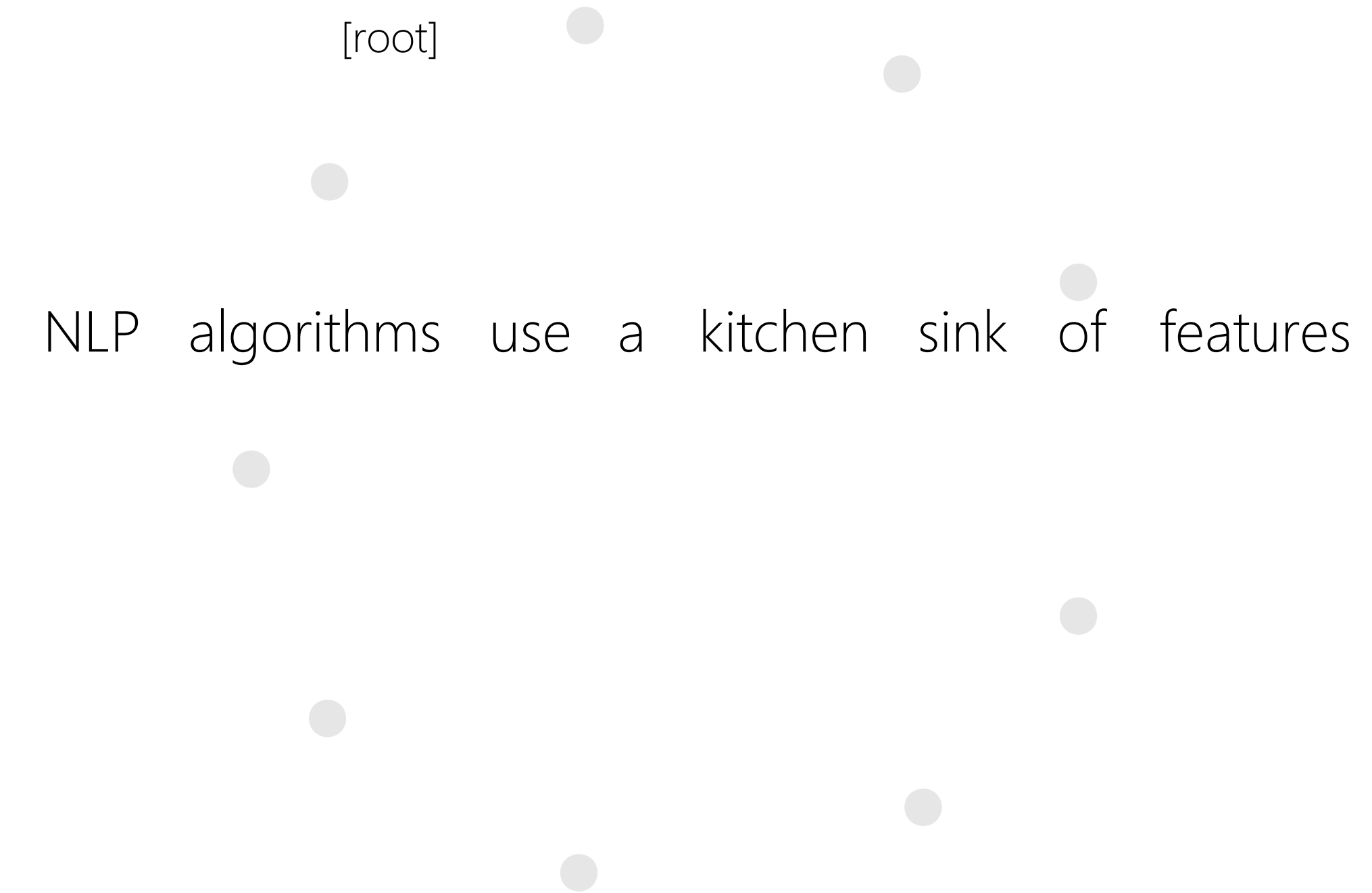
Non-determinism is a unique property of such sequential decision making processes

Allows us to reason about trajectories, and learn to trade speed for accuracy

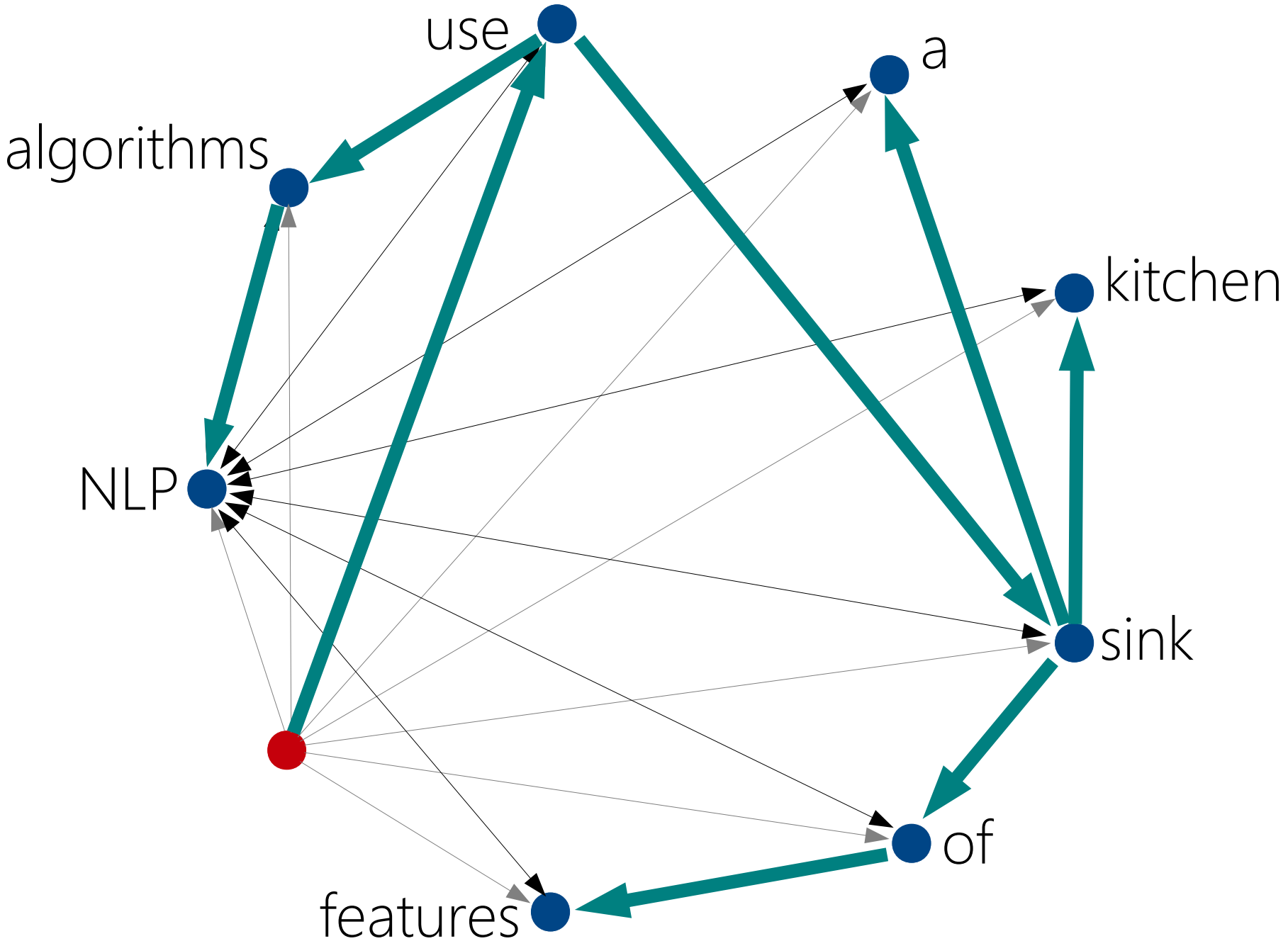
Dependency parsing



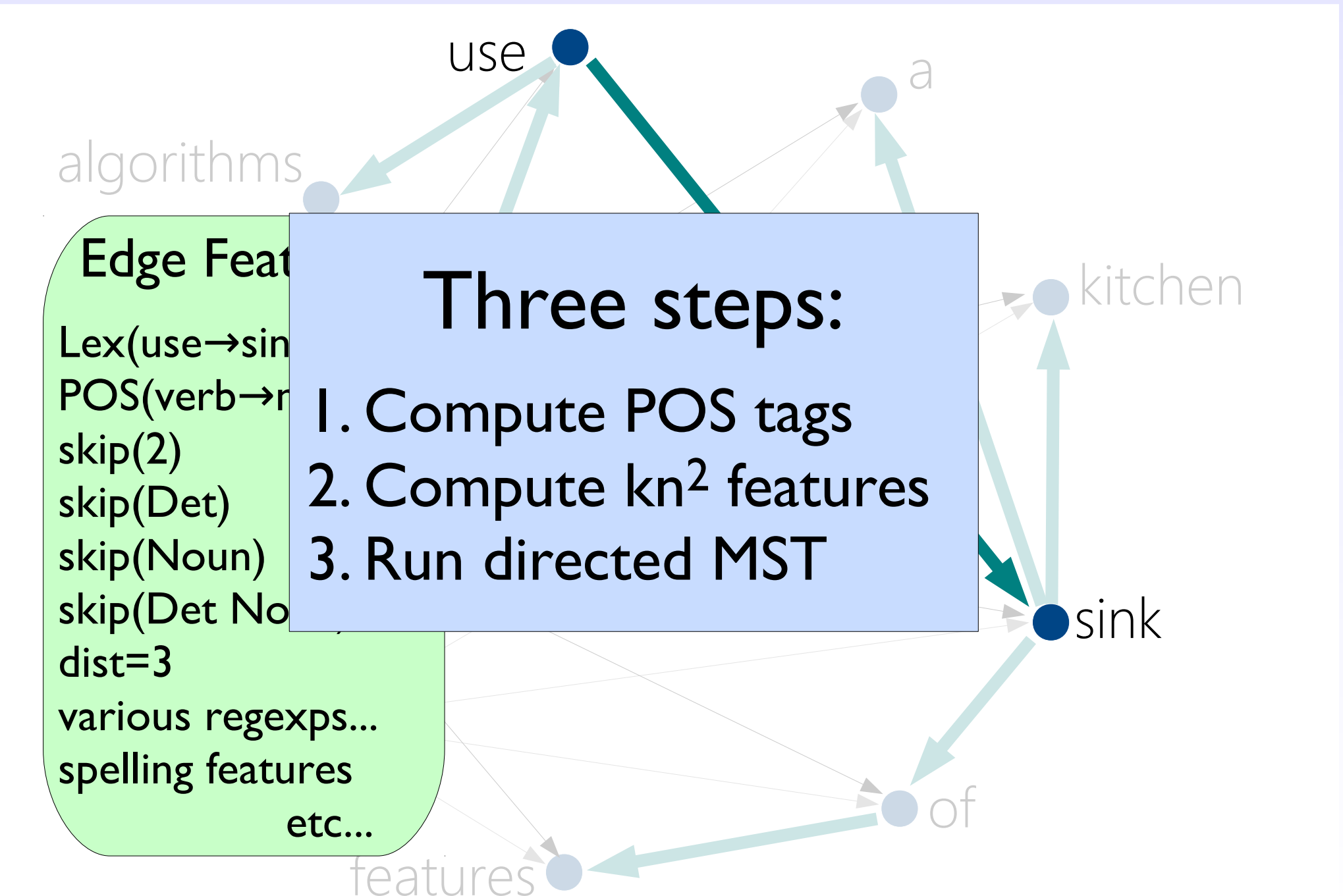
Dependency parsing



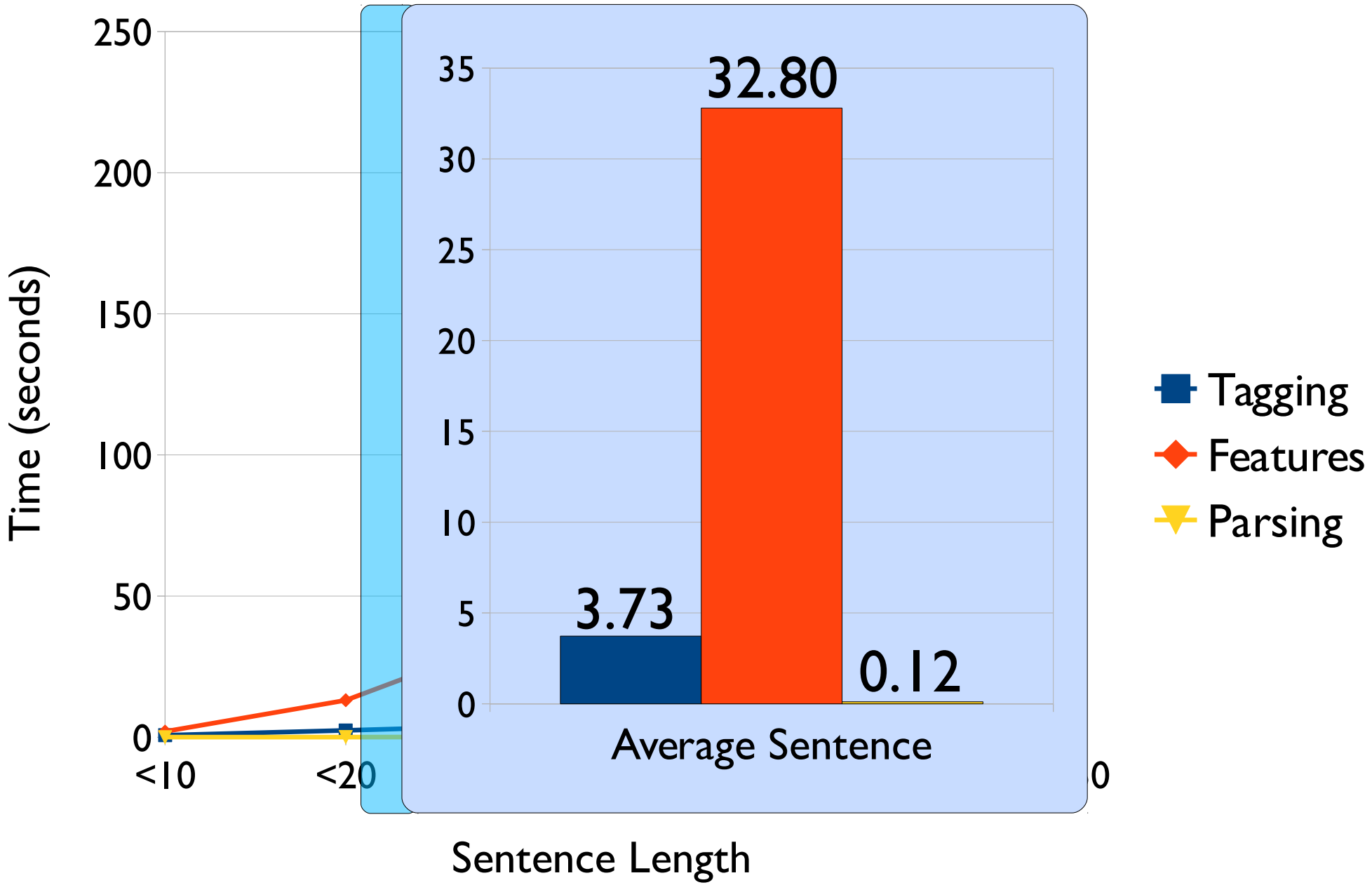
Dependency parsing



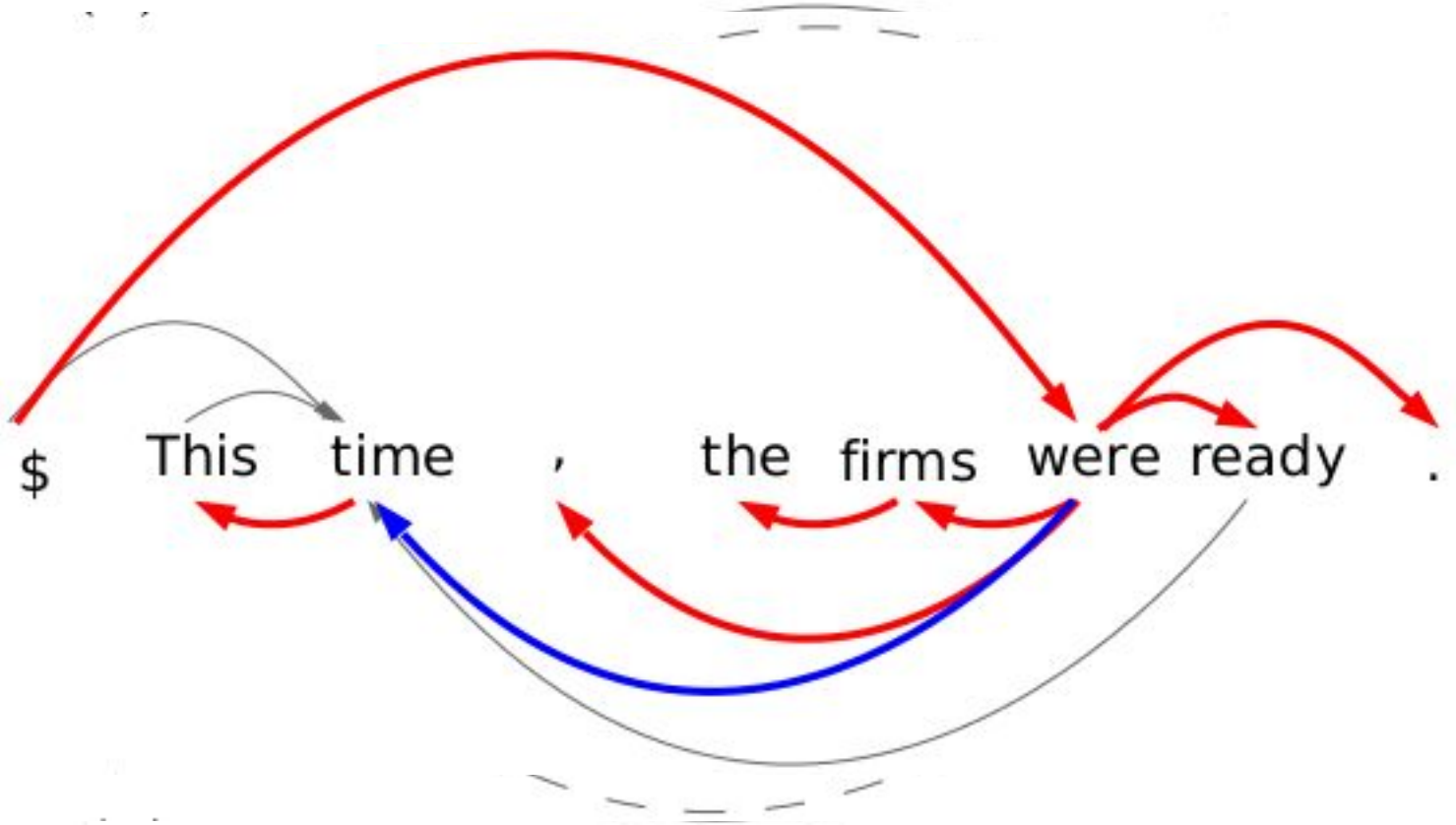
Dependency parsing



Case study: dependency parsing



Dynamic feature selection



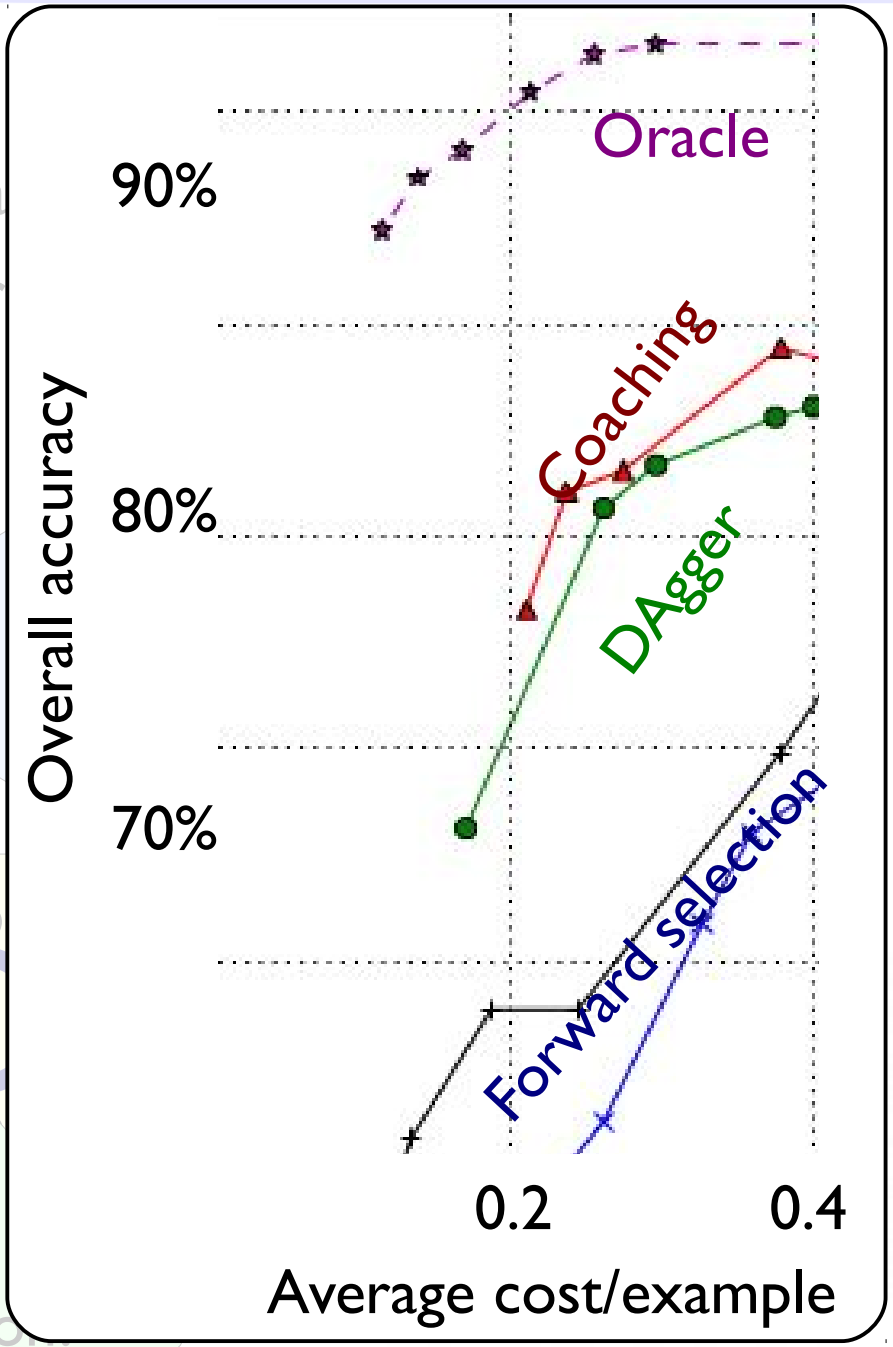
The oracle too good!

- The oracle *knows the label*
  - Picks feature with highest y*va
  - Ends after selecting one featur
- Coach says how to improve,

Psst! You should choose
argmin **E**[l(a)]
a

If N = T lo
$L(\pi_n) < T\epsilon_N$
for son

Provably smaller
DAgger's epsilon



(He+Eisner+D, NIPS 2012)

Results across languages

Language	Method	First-order				Second-order			
		Speed	Cost	UAS(D)	UAS(F)	Speed	Cost	UAS(D)	UAS(F)
Bulgarian	DYNFS	3.11	45.4	91.2	91.3	5.20	44.3	91.9	92.3
	VINEP	3.25	-	90.5	90.7	7.91	-	91.6	92.0
Chinese	DYNFS	2.09	43.5	91.0	91.3	2.72	49.4	91.4	91.7
	VINEP	1.02	-	89.3	89.5	2.03	-	90.3	90.5
English	DYNFS	5.58	24.8	91.7	91.9	5.27	49.1	92.5	92.7
	VINEP	5.23	-	91.0	91.2	11.88	-	92.2	92.4
German	DYNFS	4.71	21.0	89.2	89.3	6.02	36.6	90.4	90.2
	VINEP	3.37	-	89.0	89.2	7.38	-	90.1	90.3
Japaneses	DYNFS	5.04	16.2	93.7	93.6	8.49	19.9	93.9	93.9
	VINEP	4.60	-	91.7	92.0	14.90	-	92.1	92.0
Portuguese	DYNFS	4.36	32.9	87.3	87.1	6.84	40.4	88.0	88.3
	VINEP	4.47	-	90.0	90.1	12.32	-	90.9	91.2
Swedish	DYNFS	3.60	40.4	88.8	89.0	5.04	47.1	89.0	89.3
	VINEP	4.64	-	88.3	88.5	13.89	-	89.4	89.7

Jason Eisner

Jiarong Jiang

Adam

I'm going to be on the job market soon!

- We *can* build systems that learn to trade-off speed vs accuracy (it's hard)
- Requires new algorithms
- Important to model the problem "right"
- Imitation/reinforcement learning applied to non-deterministic search
- Orthogonal improvements to most "speedup" type papers

Thanks! Questions?