Learned Prioritization for Trading Off Speed and Accuracy

Jiarong Jiang[1] Adam Teichert[2] Hal Daumé III[1]
Jason Eisner[2]

[1]University of Maryland, College Park
[2]Johns Hopkins University

ICML workshop on Inferning: Interactions between Inference and Learning

Introduction

- Fast and accurate structured prediction

Introduction

- Fast and accurate structured prediction
- Manual exploration of speed/accuracy tradeoff
  - Prioritization heuristics
    - A* [Klein and Manning, 2003]
    - Hierarchical A* [Pauls and Klein, 2010]
  - Pruning heuristics
    - Coarse-to-fine pruning [Charniak et al., 2006; Petrov and Klein, 2007]
    - Classifier-based pruning [Roark and Hollingshead, 2008]

Introduction

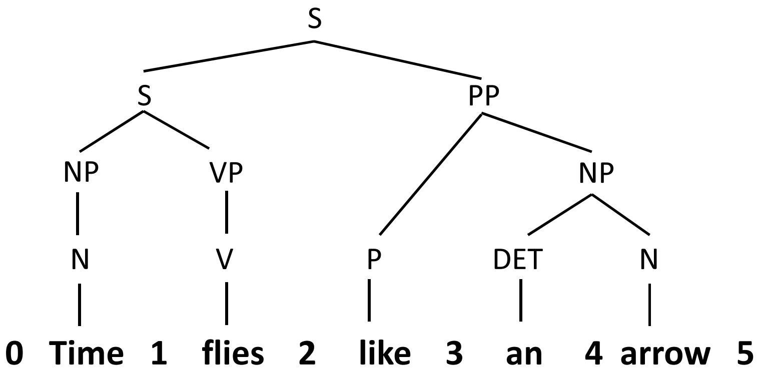
- Fast and accurate structured prediction
- Manual exploration of speed/accuracy tradeoff
  - Prioritization heuristics
    - A* [Klein and Manning, 2003]
    - Hierarchical A* [Pauls and Klein, 2010]
  - Pruning heuristics
    - Coarse-to-fine pruning [Charniak et al., 2006; Petrov and Klein, 2007]
    - Classifier-based pruning [Roark and Hollingshead, 2008]
- Goal: learn a heuristic for your input distribution, grammar, and speed/accuracy needs

Introduction

- Fast and accurate structured prediction
- Manual exploration of speed/accuracy tradeoff
  - Prioritization heuristics
    - A* [Klein and Manning, 2003]
    - Hierarchical A* [Pauls and Klein, 2010]
  - Pruning heuristics
    - Coarse-to-fine pruning [Charniak et al., 2006; Petrov and Klein, 2007]
    - Classifier-based pruning [Roark and Hollingshead, 2008]
- Goal: learn a heuristic for your input distribution, grammar, and speed/accuracy needs
- Objective measure

$$\text{quality} = \text{accuracy} - \lambda \times \text{time}$$

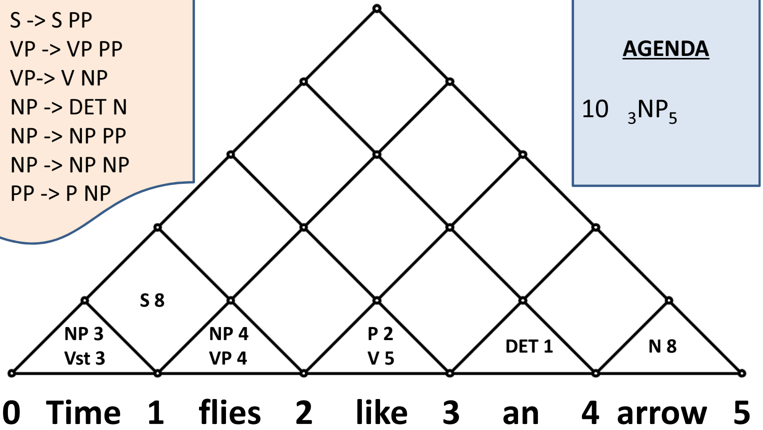
Agenda-based Parsing



Agenda-based Parsing

GRAMMAR

1 S -> NP VP
6 S -> Vst NP
2 S -> S PP
1 VP -> VP PP
2 VP-> V NP
1 NP -> DET N
2 NP -> NP PP
3 NP -> NP NP
0 PP -> P NP



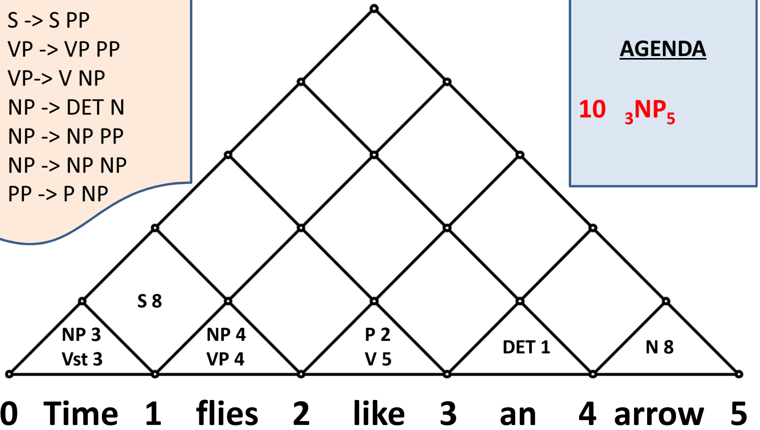
AGENDA

10 ${}_3NP_5$

Agenda-based Parsing

GRAMMAR

1 S -> NP VP
6 S -> Vst NP
2 S -> S PP
1 VP -> VP PP
2 VP-> V NP
1 NP -> DET N
2 NP -> NP PP
3 NP -> NP NP
0 PP -> P NP



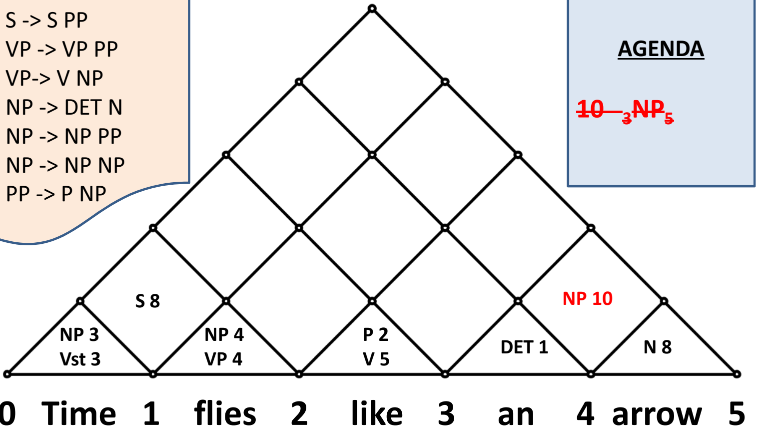
AGENDA

10 $_3NP_5$

Agenda-based Parsing

GRAMMAR

1 S -> NP VP
6 S -> Vst NP
2 S -> S PP
1 VP -> VP PP
2 VP-> V NP
1 NP -> DET N
2 NP -> NP PP
3 NP -> NP NP
0 PP -> P NP



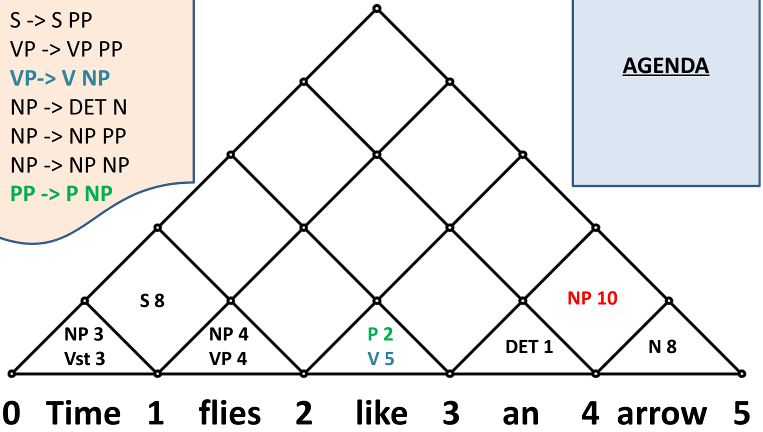
AGENDA

~~10 $_3$NP$_5$~~

Agenda-based Parsing

GRAMMAR

1 S -> NP VP
6 S -> Vst NP
2 S -> S PP
1 VP -> VP PP
2 VP-> V NP
1 NP -> DET N
2 NP -> NP PP
3 NP -> NP NP
0 PP -> P NP

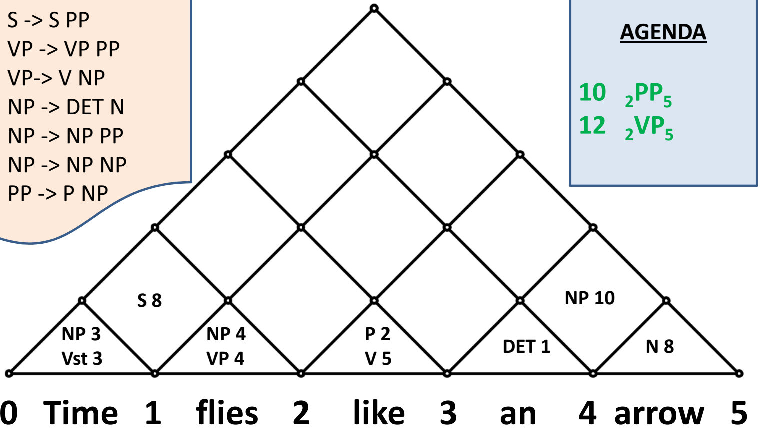


AGENDA

Agenda-based Parsing

GRAMMAR

1 S -> NP VP
6 S -> Vst NP
2 S -> S PP
1 VP -> VP PP
2 VP-> V NP
1 NP -> DET N
2 NP -> NP PP
3 NP -> NP NP
0 PP -> P NP



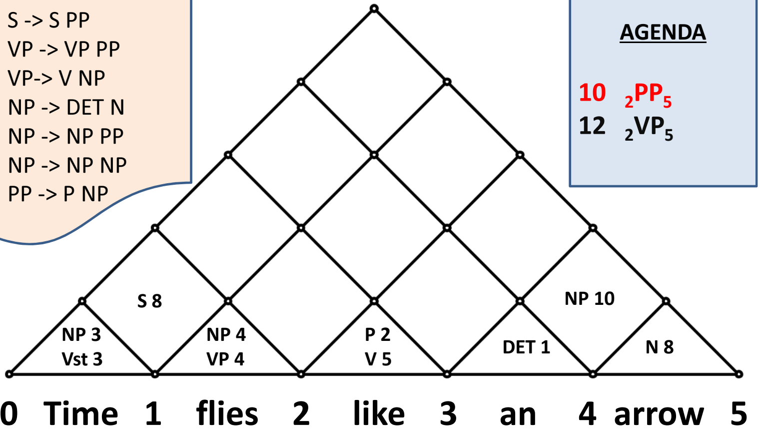
AGENDA

10 ${}_2PP_5$
12 ${}_2VP_5$

Agenda-based Parsing

GRAMMAR

1 S -> NP VP
6 S -> Vst NP
2 S -> S PP
1 VP -> VP PP
2 VP-> V NP
1 NP -> DET N
2 NP -> NP PP
3 NP -> NP NP
0 PP -> P NP



AGENDA

10 ${}_2PP_5$
12 ${}_2VP_5$

Speed Accuracy for Agenda-based Parsing

- All experiments are on Penn Treebank WSJ with sentence length $\leq$ 15.
- Preliminary results setup:
  - Berkeley latent variable PCFG trained on section 2-20
  - Training set: 100 sentences from section 21
  - Evaluated on the same 100 sentences
- Baseline 1: Exhaustive Search
  Recall: *93.3*; *Relative number of pops*: *3.0x*
- Baseline 2: Uniform Cost Search (UC)
  Recall: *93.3*; *Relative number of pops*: *1.0x*
- Baseline 3: Pruned Uniform Cost Search
  Recall: *92.0*; *Relative number of pops*: *0.33x*

Agenda-based Parsing as a Markov Decision Process

- State space: current chart and agenda
- Action: *pop* a partial parse from the agenda
- Transition: Given the chosen action, deterministically updates chart and pushes other parses to the agenda
- Policy: computes action priorities from extracted features

$$\pi_\theta(s) = \arg\max_a \theta \cdot \phi(a, s)$$

- (Delayed) Reward

$$\text{reward} = \text{accuracy} - \lambda \times \text{time}$$

  - accuracy = labeled span recall
  - time = # of pops from agenda

Agenda-based Parsing as a Markov Decision Process

- State space: current chart and agenda
- Action: *pop* a partial parse from the agenda
- Transition: Given the chosen action, deterministically updates chart and pushes other parses to the agenda
- Policy: computes action priorities from extracted features

$$\pi_\theta(s) = \arg\max_a \theta \cdot \phi(a, s)$$

- (Delayed) Reward

$$\text{reward} = \text{accuracy} - \lambda \times \text{time}$$

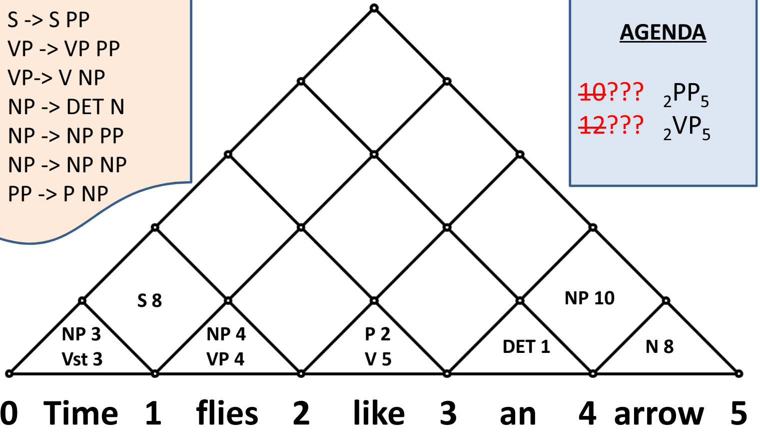
  - accuracy = labeled span recall
  - time = # of pops from agenda

Learning Policy = Learning Prioritization Function

Decoding as a Markov Decision Process (MDP)

GRAMMAR

1 S -> NP VP
6 S -> Vst NP
2 S -> S PP
1 VP -> VP PP
2 VP-> V NP
1 NP -> DET N
2 NP -> NP PP
3 NP -> NP NP
0 PP -> P NP



AGENDA

~~10~~??? $_2PP_5$
~~12~~??? $_2VP_5$

Boltzmann Exploration

- Transition at test time: deterministic
- Transition at training time: exploration with stochastic policies: $\pi_{\vec{\theta}}(a \mid s)$.
- Boltzmann exploration:

$$\pi_{\vec{\theta}}(a \mid s) = \frac{1}{Z(s)} \exp\left[\frac{1}{temp}\,\vec{\theta} \cdot \vec{\phi}(a, s)\right]$$

- Temperature $\to 0$, exploration $\to$ exploitation
- A trajectory $\tau = \langle s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_T, a_T, r_T \rangle$.
- Expected future reward:

$$R = \mathbb{E}_{\tau \sim \pi_{\vec{\theta}}}[R(\tau)] = \mathbb{E}_{\tau \sim \pi_{\vec{\theta}}}\left[\sum_{t=0}^{T} r_t\right].$$

Policy Gradient

- Find parameters that maximize the expected reward with respect to the induced distribution over trajectories
- Policy gradient [Sutton et al., 2000]
  The gradient of the objective

$$\nabla_{\vec{\theta}} \mathbb{E}_{\tau}[R(\tau)] = \mathbb{E}_{\tau}\left[R(\tau) \sum_{t=0}^{T} \nabla_{\vec{\theta}} \log \pi(a_t \mid s_t)\right]$$

where

$$\nabla_{\vec{\theta}} \log \pi_{\vec{\theta}}(a \mid s) = \frac{1}{temp}\left(\vec{\phi}(a_t, s_t) - \sum_{a' \in A} \pi_{\vec{\theta}}(a' \mid s_t) \vec{\phi}(a', s_t)\right)$$

Features

1. Width of partial parse
2. Viterbi inside score
3. Touches start of sentence?
4. Touches end of sentence?
5. Ratio of width to sentence length
6. log $p(\text{label} \mid \text{prev POS})$ and log $p(\text{label} \mid \text{next POS})$ (statistics extracted from labeled trees, word POS assumed to be most frequent)
7. Case pattern of first word in partial parse and previous/next word
8. Punctuation pattern in partial parse (five most frequent)

Policy Gradient with Boltzmann Exploration

- Preliminary results:

Method	Recall	Relative # of pops
Policy Gradient w/ Boltzmann Exploration	56.4	0.46x
Uniform cost search	93.3	1.0x
Pruned uniform cost search	92.0	0.33x

Policy Gradient with Boltzmann Exploration

- Preliminary results:

Method	Recall	Relative # of pops
Policy Gradient w/ Boltzmann Exploration	56.4	0.46x
Uniform cost search	93.3	1.0x
Pruned uniform cost search	92.0	0.33x

- **Main Difficulty**:

  Which actions were “responsible” for a trajectory’s reward?

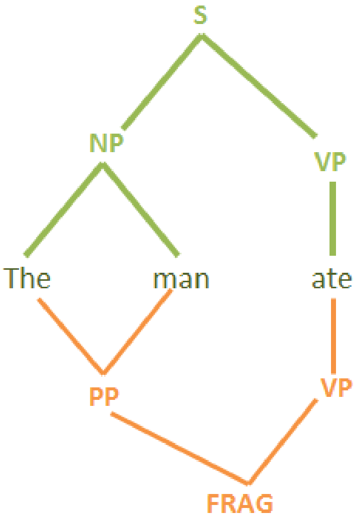
Reward Shaping

- Goal: give the agent reward *earlier* in a trajectory in order to improve its convergence rate
- Push back reward to actions

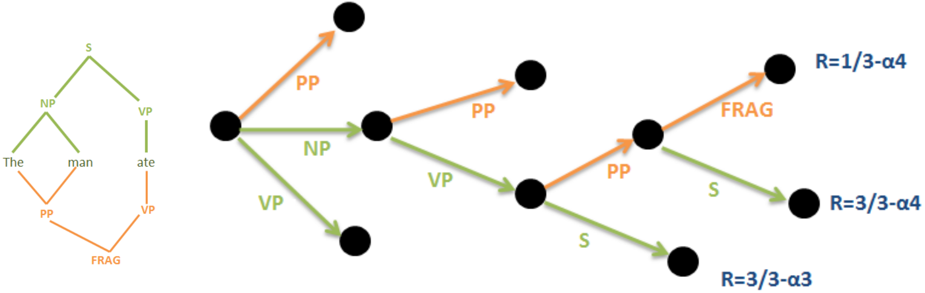
$$\tilde{r}(s,a) = \begin{cases} \xi(a)/n - \lambda & \text{if } a \text{ is a full parse tree} \\ 1/n - \lambda & \text{if } a \text{ is in the true parse} \\ -\lambda & \text{otherwise} \end{cases}$$

  $\xi(s)$: a negative reward for actions which received early reward for constituents that were not in the final parse
- Property: $R(\tau) = \sum_{t=0}^{T} \tilde{r}(s,a)$

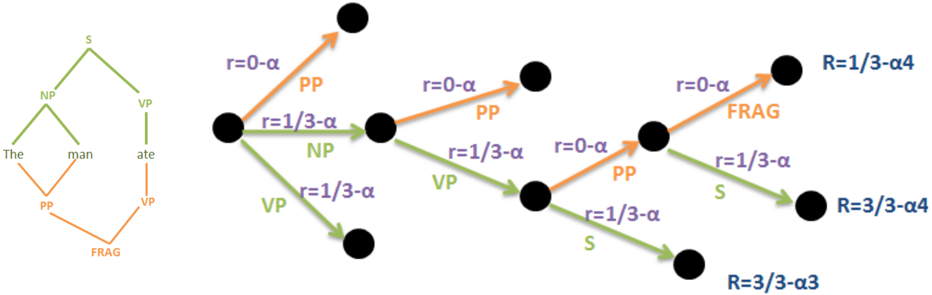
Reward Shaping



Reward Shaping



Reward Shaping



Reward Shaping

- Gradient step:

$$\nabla_\theta \mathbb{E}_\tau[R(\tau)] = \nabla_\theta \mathbb{E}_\tau[\tilde{R}(\tau)] = \mathbb{E}_\tau \left[\sum_{t=0}^{T} \left(\sum_{t'=t}^{T} \gamma^{t'-t} \tilde{r}_{t'} \right) \nabla_\theta \log \pi(a_t \mid s_t) \right]$$

Reward Shaping

- Gradient step:

$$\nabla_\theta \mathbb{E}_\tau[R(\tau)] = \nabla_\theta \mathbb{E}_\tau[\tilde{R}(\tau)] = \mathbb{E}_\tau \left[\sum_{t=0}^{T} \left(\sum_{t'=t}^{T} \gamma^{t'-t} \tilde{r}_{t'} \right) \nabla_\theta \log \pi(a_t \mid s_t) \right]$$

- Preliminary results:

Method	Recall	Relative # of pops
Policy Gradient w/ Reward Shaping	76.5	0.13x
Policy Gradient w/ Boltzmann Exploration	56.4	0.46x
Uniform cost search Pruned uniform cost search	93.3 92.0	1.0x 0.33x

Reward Shaping

- Gradient step:

$$\nabla_\theta \mathbb{E}_\tau[R(\tau)] = \nabla_\theta \mathbb{E}_\tau[\tilde{R}(\tau)] = \mathbb{E}_\tau \left[\sum_{t=0}^{T} \left(\sum_{t'=t}^{T} \gamma^{t'-t} \tilde{r}_{t'} \right) \nabla_\theta \log \pi(a_t \mid s_t) \right]$$

- Preliminary results:

Method	Recall	Relative # of pops
Policy Gradient w/ Reward Shaping	76.5	0.13x
Policy Gradient w/ Boltzmann Exploration	56.4	0.46x
Uniform cost search Pruned uniform cost search	93.3 92.0	1.0x 0.33x

- **Main difficulty**:

Only a few trajectories are reasonable!

Oracle Actions

- Focus on high-reward regions of policy space

Oracle Actions

- Focus on high-reward regions of policy space
- Oracle action: an action that leads to a maximum-reward tree, where reward is defined in terms of accuracy *and* speed

Oracle Actions

- Focus on high-reward regions of policy space
- Oracle action: an action that leads to a maximum-reward tree, where reward is defined in terms of accuracy *and* speed
- How to get oracle actions?
  - Ground truth of a sentence
  - Exact parse with the best speed-accuracy tradeoff

Oracle Actions

- Focus on high-reward regions of policy space
- Oracle action: an action that leads to a maximum-reward tree, where reward is defined in terms of accuracy *and* speed
- How to get oracle actions?
  - Ground truth of a sentence
  - Exact parse with the best speed-accuracy tradeoff
- Apprenticeship learning via classification
  1. Generate classification examples (s_t, a_t) labeled according to oracle actions
  2. Train a maximum entropy classifier
  3. Classifier objective: maximize number of times policy matches oracle action

Apprenticeship Learning via Classification

- Preliminary results:

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

Apprenticeship Learning via Classification

- Preliminary results:

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

- **Main difficulty**:

Too hard to imitate oracle with our features!

Oracle-Infused Policy Gradient

- Goal: "interleaving" oracle actions with policy actions both feasible and sensible
- Let π be an arbitrary policy and let $\delta \in [0, 1]$. The oracle infused policy π^{+}_{δ} is defined as follows:

$$\pi^{+}_{\delta}(a \mid s) = \delta\pi^{*}(a \mid s) + (1 - \delta)\pi(a \mid s)$$

  - $\delta = 1$: the classifier-based approach
  - $\delta = 0$: policy gradient
  - $\delta = 0.8^{\text{epoch}}$

Oracle-Infused Policy Gradient

- Preliminary results:

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

Pareto Frontier

- **Final Results** Setup:
  - Berkeley latent variable PCFG trained on sections 2-21
  - RL (if any) trained on section 22
  - evaluated on section 23
- Baselines:
  - **(HA*)** a Hierarchical A* parser [3] with same pruning threshold at each hierarchy level
  - **(UC)** uniform cost search
  - **(UC$_p$)** pruned uniform cost search
  - **(A$^*_p$)** an A* variant, on which we decrease the pruning threshold if no tree is returned
  - **(CTF)** an agenda-based coarse-to-fine parser [4].

Pareto Frontier

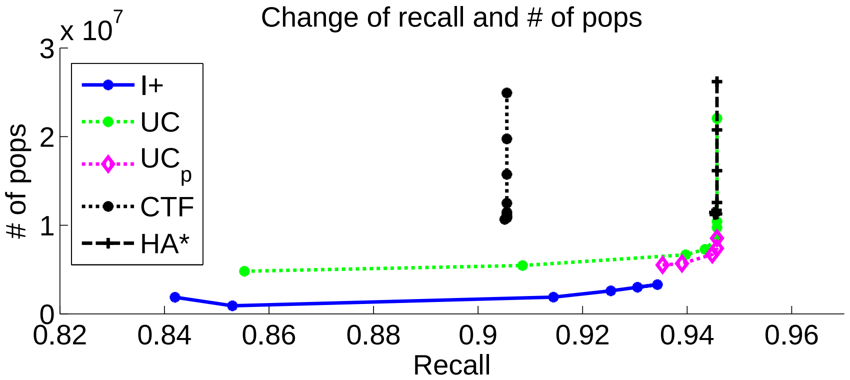


Figure: Pareto frontiers: Our **I+** parser at different values of λ, against the baselines at different pruning levels. *Lower and further right* is better.

Discussion and Conclusion

- A novel oracle-infused variant of the policy gradient algorithm for reinforcement learning
- Learn a fast and accurate parser with only a simple set of features
- Limitation of the model:
  - Feature effectiveness v.s. cost
  - Stop criteria

1. H. Daumé III, J. Langford, and D. Marcu. 2009. Search-based structured prediction. Machine Learning, 75(3):297—C325.
2. V. Gullapalli and A. G. Barto. 1992. Shaping as a method for accelerating reinforcement learning. In Proceedings of the IEEE International Symposium on Intelligent Control.
3. A. Y. Ng, D. Harada, and S. Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping In Proceedings of the Sixteenth International Conference on Machine Learning.
4. A. Pauls and D. Klein. 2009. Hierarchical search for parsing. In NAACL/HLT.
5. S. Petrov and D. Klein. 2007. Improved inference for unlexicalized parsing. In NAACL/HLT.
6. S. Ross, G. J. Gordon, and J. A. Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In AI-Stats.