Agenda Based Parsing

- **Goal**: find most likely parse w.r.t. a grammar
- **State Space**: current chart and agenda
- **Action**: choose a partial parse from agenda
- **Transitions**: given the chosen action, deterministically updates chart and builds and pushes other partial (or full) parses to agenda
- **Reward**: accuracy – time
e.g. Accuracy = labeled span recall, Time = # of pops from agenda
- **Policy**: deterministically pops highest-priority available action: \( \pi(s) = \arg\max_\pi \cdot <a, s> \)

\[ \text{learning a policy = learning the priority function} \]

Agenda Based Parsing as a Markov Decision Process

- **State Space**: current chart and agenda
- **Action**: choose a partial parse from agenda
- **Transitions**: given the chosen action, deterministically updates chart and builds and pushes other partial (or full) parses to agenda
- **Reward**: accuracy – time
e.g. Accuracy = labeled span recall, Time = # of pops from agenda
- **Policy**: deterministically pops highest-priority available action: \( \pi(s) = \arg\max_\pi \cdot <a, s> \)

\[ \text{learning a policy = learning the priority function} \]

Policy Gradient with Reward Shaping

- **Weakness of vanilla policy gradient with Boltzmann exploration**: No attempt to determine which actions were responsible for a trajectory’s reward
- **Reward Shaping**: fast convergence

\[ \pi(s) = \rho <a, s> \quad \text{if a is a full parse tree} \]
\[ \pi(s) = 1 - \rho \quad \text{if a is in the true parse} \]
\[ \text{otherwise} \]

\( \rho <a, s> \): a negative reward for actions which received early reward for constituents that were not in the final parse.

\[ \text{Result on development data: Recall = 56.4, Relative \# of pops = 0.46x} \]

**Solution**: Oracle-Infused Policy Gradient

- **Oracle action**: action that leads to a maximum-reward tree
- **Apprenticeship learning via classification**: following oracle trajectories = training a supervised log-linear classifier

\[ \text{Result on development data: Recall = 84.2, Relative \# of pops = 0.85x} \]

**Too hard to imitate oracle with our features**

- **Oracle-infused policy**: \( \pi^*_w = \delta_{\delta} \left( \rho \pi^*_w(s) + (1 - \delta) \pi_w(s) \right) \)

\( \delta = \delta_{\text{epoch}} \)

- **Result on development data**: Recall = 91.2, Relative \# of pops = 0.46x

**Solution**: explore near oracle \[ \delta_{\text{epoch}} \text{ slow}, \delta_{\text{epoch}} \text{ near learned policy} \]

---

**Learned Prioritization for Trading Off Accuracy and Speed**

Jiarong Jiang\* Adam Teichert\* Hal Daumé III\* Jason Eisner\* 
\*University of Maryland College Park \**Johns Hopkins University

---

**Policy Gradient with Reward Shaping**

- **Weakness of vanilla policy gradient with Boltzmann exploration**: No attempt to determine which actions were responsible for a trajectory’s reward
- **Reward Shaping**: fast convergence

\[ \pi(s) = \rho <a, s> \quad \text{if a is a full parse tree} \]
\[ \pi(s) = 1 - \rho \quad \text{if a is in the true parse} \]
\[ \text{otherwise} \]

\( \rho <a, s> \): a negative reward for actions which received early reward for constituents that were not in the final parse.

\[ \text{Result on development data: Recall = 56.4, Relative \# of pops = 0.46x} \]

**Solution**: Oracle-Infused Policy Gradient

- **Oracle action**: action that leads to a maximum-reward tree
- **Apprenticeship learning via classification**: following oracle trajectories = training a supervised log-linear classifier

\[ \text{Result on development data: Recall = 84.2, Relative \# of pops = 0.85x} \]

**Too hard to imitate oracle with our features**

- **Oracle-infused policy**: \( \pi^*_w = \delta_{\delta} \left( \rho \pi^*_w(s) + (1 - \delta) \pi_w(s) \right) \)

\( \delta = \delta_{\text{epoch}} \)

- **Result on development data**: Recall = 91.2, Relative \# of pops = 0.46x

**Solution**: explore near oracle \[ \delta_{\text{epoch}} \text{ slow}, \delta_{\text{epoch}} \text{ near learned policy} \]

---

**Learned Prioritization for Trading Off Accuracy and Speed**

Jiarong Jiang\* Adam Teichert\* Hal Daumé III\* Jason Eisner\* 
\*University of Maryland College Park \**Johns Hopkins University

---

**Policy Gradient with Reward Shaping**

- **Weakness of vanilla policy gradient with Boltzmann exploration**: No attempt to determine which actions were responsible for a trajectory’s reward
- **Reward Shaping**: fast convergence

\[ \pi(s) = \rho <a, s> \quad \text{if a is a full parse tree} \]
\[ \pi(s) = 1 - \rho \quad \text{if a is in the true parse} \]
\[ \text{otherwise} \]

\( \rho <a, s> \): a negative reward for actions which received early reward for constituents that were not in the final parse.

\[ \text{Result on development data: Recall = 56.4, Relative \# of pops = 0.46x} \]

**Solution**: Oracle-Infused Policy Gradient

- **Oracle action**: action that leads to a maximum-reward tree
- **Apprenticeship learning via classification**: following oracle trajectories = training a supervised log-linear classifier

\[ \text{Result on development data: Recall = 84.2, Relative \# of pops = 0.85x} \]

**Too hard to imitate oracle with our features**

- **Oracle-infused policy**: \( \pi^*_w = \delta_{\delta} \left( \rho \pi^*_w(s) + (1 - \delta) \pi_w(s) \right) \)

\( \delta = \delta_{\text{epoch}} \)

- **Result on development data**: Recall = 91.2, Relative \# of pops = 0.46x

**Solution**: explore near oracle \[ \delta_{\text{epoch}} \text{ slow}, \delta_{\text{epoch}} \text{ near learned policy} \]