Reinforcement Learning for NLP

Advanced Machine Learning for NLP

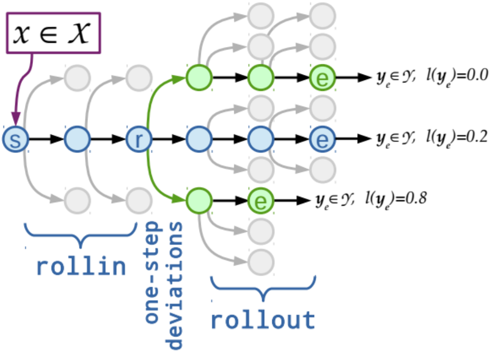
Jordan Boyd-Graber

DEEP SHIFT-REDUCE PARSERS

What Makes NLP different from RL?

- Often, best actions are **known**
- We're not just searching for high-reward
- Sometimes actions themselves are known

Roll In vs. Roll Out



- Roll In: Which states does the algorithm see
- Roll Out: What states do you use for training

Known Policy vs. Exploration

roll-out → / ↓ roll-in	**Reference**	**Mixture**	**Learned**
Reference	Inconsistent		
Learned	Not locally opt.	Good	RL

- RL only gets reward
- Roll-in with reference gives unrealistic trajectories
- How to incorporate knowledge of true actions?
- Train classifier as proxy for policy

RL for Translation

RL for Translation

Observation

1. Mit dem Zug

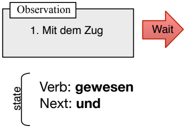
RL for Translation

Observation

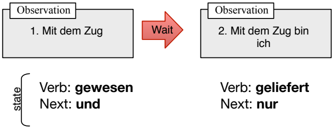
1. Mit dem Zug

state $\left\{ \begin{array}{l} \text{Verb: } \textbf{gewesen} \\ \text{Next: } \textbf{und} \end{array} \right.$

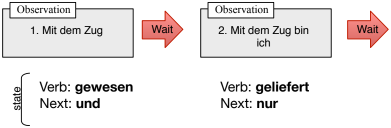
RL for Translation



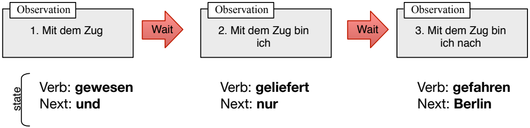
RL for Translation



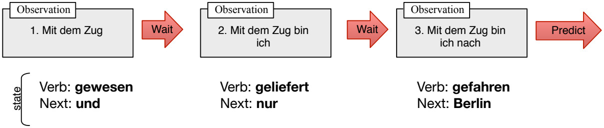
RL for Translation



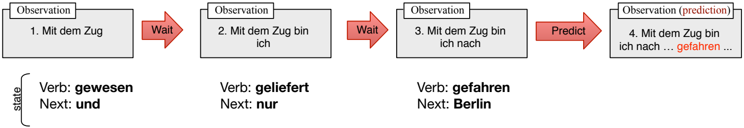
RL for Translation



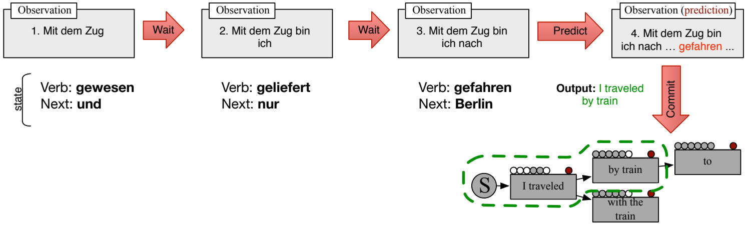
RL for Translation



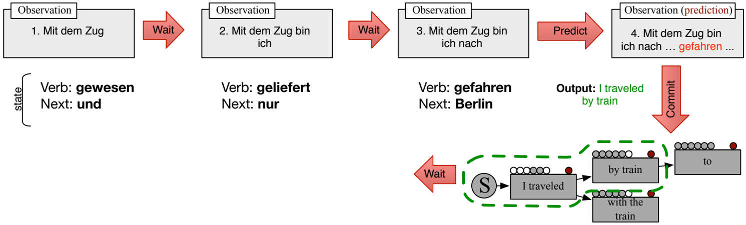
RL for Translation



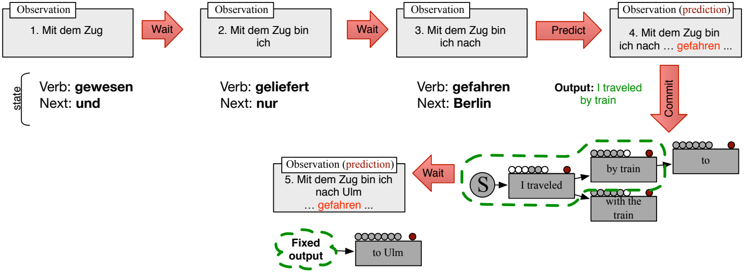
RL for Translation



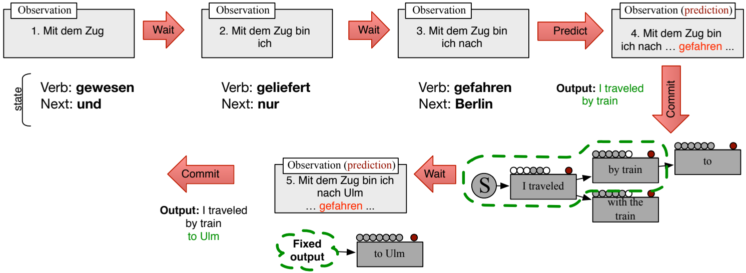
RL for Translation



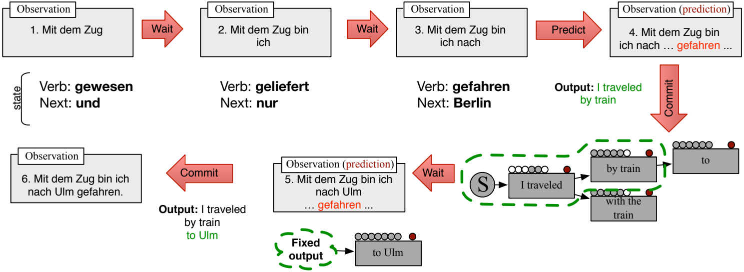
RL for Translation



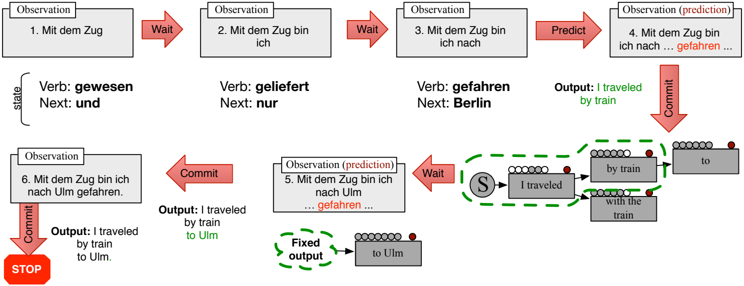
RL for Translation



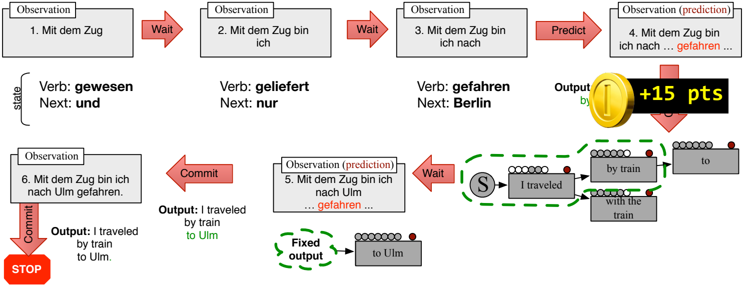
RL for Translation



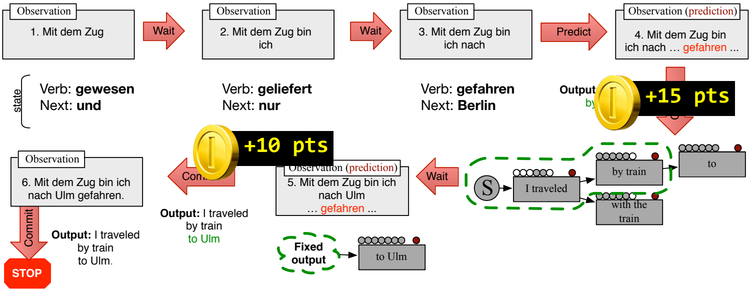
RL for Translation



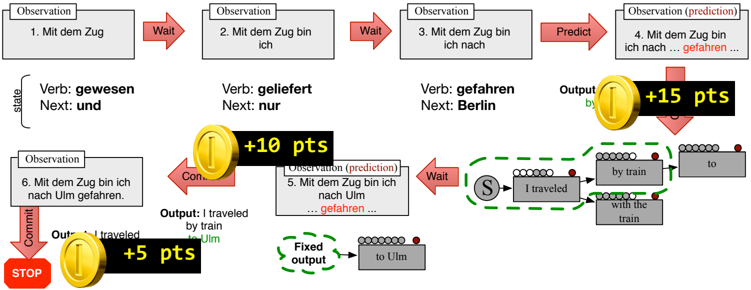
RL for Translation



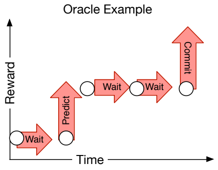
RL for Translation



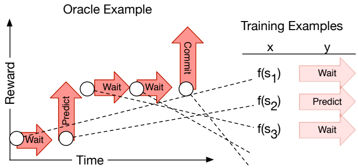
RL for Translation



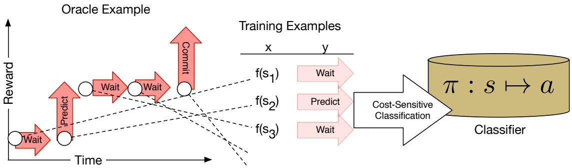
How do we find a good policy?



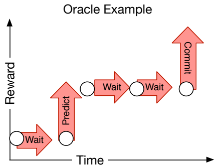
How do we find a good policy?

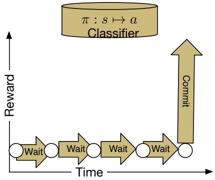


How do we find a good policy?

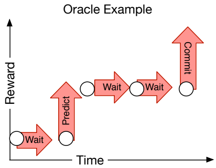


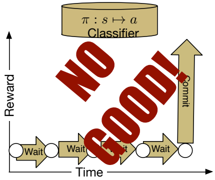
How do we find a good policy?



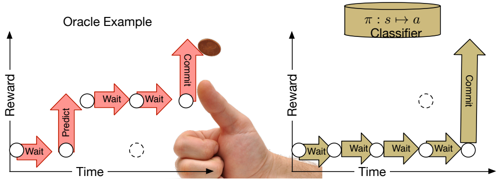


How do we find a good policy?

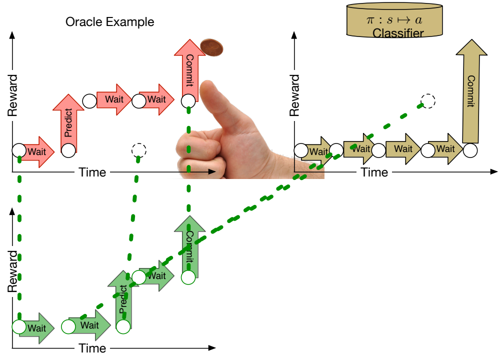




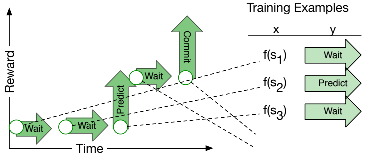
How do we find a good policy?



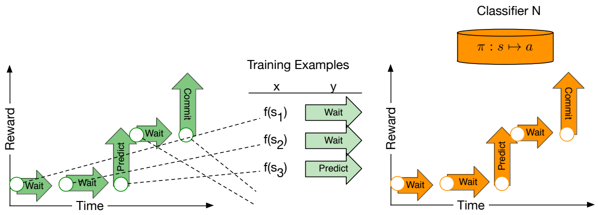
How do we find a good policy?



How do we find a good policy?



How do we find a good policy?



LOLS

Algorithm 1 Locally Optimal Learning to Search (LOLS)

Require: Dataset $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$ drawn from $\mathcal{D}$ and $\beta \geq 0$: a mixture parameter for roll-out.

Initialize a policy π_0.
for all $i \in \{1, 2, \ldots, N\}$ (loop over each instance) **do**
    Generate a reference policy π^{ref} based on $\boldsymbol{y}_i$.
    Initialize $\Gamma = \emptyset$.
    for all $t \in \{0, 1, 2, \ldots, T-1\}$ **do**
        Roll-in by executing $\pi_i^{\text{in}} = \hat{\pi}_i$ for t rounds and reach s_t.
        for all $a \in A(s_t)$ **do**
            Let $\pi_i^{\text{out}} = \pi^{\text{ref}}$ with probability β, otherwise $\hat{\pi}_i$.
            Evaluate cost $c_{i,t}(a)$ by rolling-out with π_i^{out} for $T - t - 1$ steps.
        end for
        Generate a feature vector $\Phi(\boldsymbol{x}_i, s_t)$.
        Set $\Gamma = \Gamma \cup \{\langle c_{i,t}, \Phi(\boldsymbol{x}_i, s_t)\rangle\}$.
    end for
    $\hat{\pi}_{i+1} \leftarrow \text{Train}(\hat{\pi}_i, \Gamma)$ (Update).
end for
Return the average policy across $\hat{\pi}_0, \hat{\pi}_1, \ldots \hat{\pi}_N$.

LOLS on Dependency Parsing

Policy is learning actions for shift-reduce parser

roll-out → ↓ roll-in	**Reference**	**Mixture**	**Learned**
Reference is optimal			
Reference	87.2	89.7	88.2
Learned	**90.7**	90.5	86.9
Reference is suboptimal			
Reference	83.3	87.2	81.6
Learned	87.1	**90.2**	86.8
Reference is bad			
Reference	68.7	65.4	66.7
Learned	75.8	**89.4**	87.5

But what structure is best?

- RecNN not much better than DAN
- But syntax may not be optimal
- Can we learn structure?

But what structure is best?

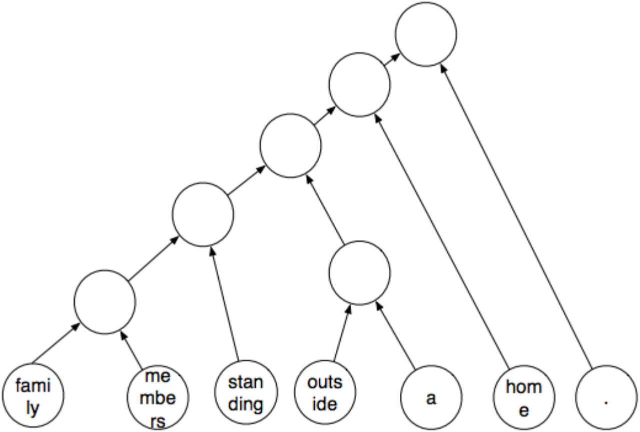
- RecNN not much better than DAN
- But syntax may not be optimal
- Can we learn structure?
  - Policy learns shift-reduce parser
  - TreeLSTM with **learned structure**
  - Reward is performance on downstream task

Performance

Table 4: Classification accuracy on SNLI dataset.

Model	Acc.	# params.
100D-Right to left	79.1	2.3m
100D-Left to right	80.2	2.3m
100D-Bidirectional	80.2	2.6m
100D-Supervised syntax	78.5	2.3m
100D-Semi-supervised syntax	80.2	2.3m
100D-Latent syntax	**80.5**	2.3m

What structures?



Other places of NLP + RL

- Question answering
- Language games
- Dialog systems
- Human learning

Wrapup

- RL allows for algorithms to think about long-term rewards
- And to guide actions of a system
- Important for systems that interact with world
- Discrete action spaces often more difficult