Reinforcement Learning for Mapping Instructions to Actions

S.R.K. Branavan, Harr Chen, Luke S. Zettlemoyer, Regina Barzilay

CSAIL, MIT

2009

Introduction

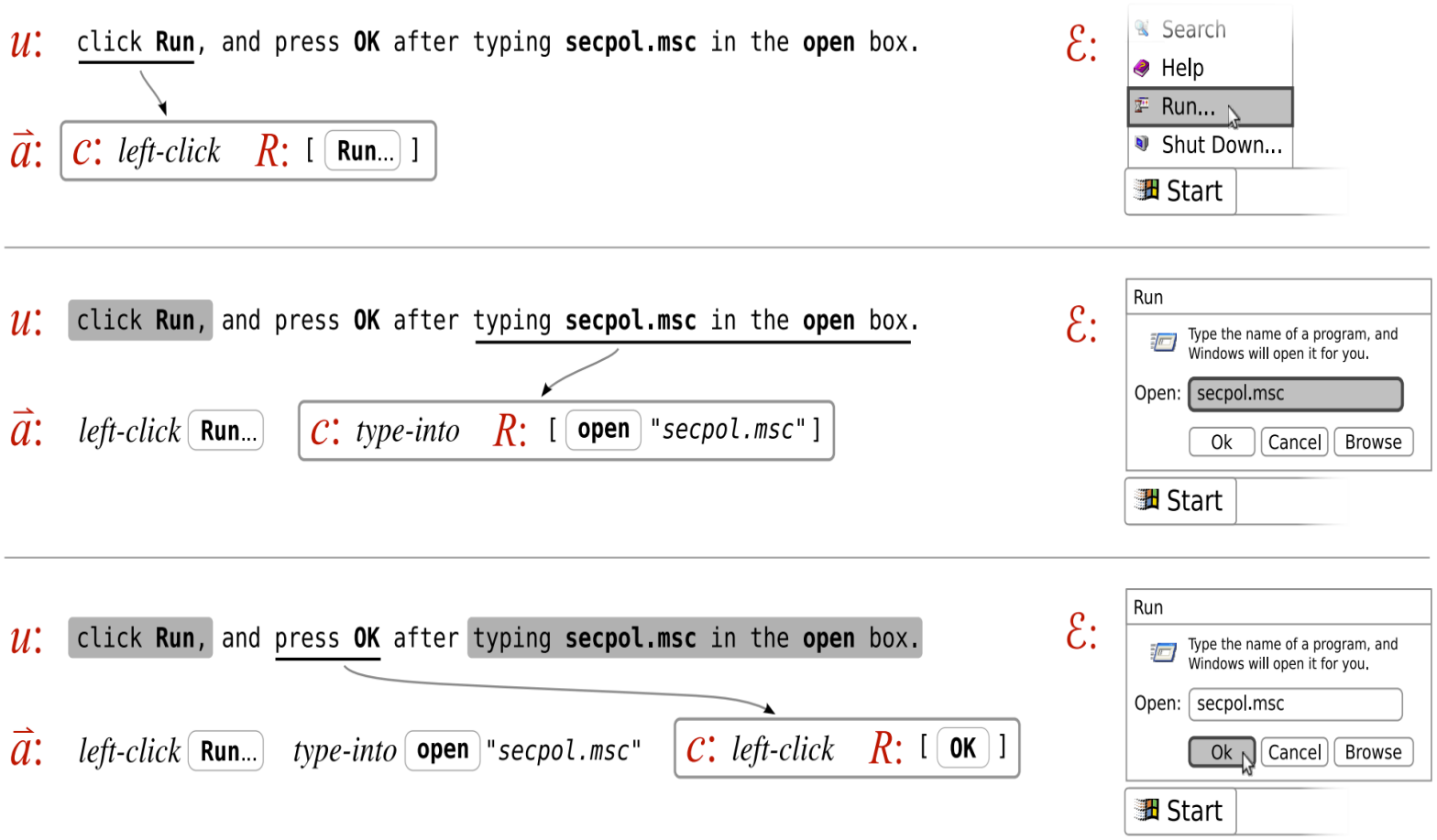
- Mapping natural language instructions to sequences of executable actions using RL
- Use policy gradient
- 2 domains :
  - Windows troubleshooting guides
  - Game tutorials

- Click start, point to search, and then click for files or folders.
- In the search results dialog box, on the tools menu, click folder options.
- In the folder options dialog box, on the view tab, under advanced settings, click *show hidden files and folders*, and then click to clear the *hide file extensions for known file types* check box.
- Click apply, and then click ok.
- In the search for files or folders named box, type msdownld.tmp.
- In the look in list, click my computer, and then click search now.
- In the search results pane, right-click msdownld.tmp and then click delete on the shortcut menu, a *confirm folder delete* message appears.
- Click yes.

Problem Formulation

- Mapping text to actions:
  - Document d
  - Sequence of sentences $(u_1, \ldots, u_\ell)$
  - Map d to actions $\vec{a} = (a_0, \ldots, a_{n-1})$
  - Action $a = (c, R, W')$ command c, command parameters R, words W' in sentence specifying c
  - Environment $\mathcal{E}$: set of objects + their properties
  - Mapping state s : $(\mathcal{E}, d, j, W)$ j: current sentence index, W: words already mapped by previous actions in j

3 step mapping from sentence to actions



$d = (u_1, \ldots, u_\ell)$	Document
u	Sentence
$\vec{a} = (a_0, \ldots, a_{n-1})$	Sequence of actions
$a = (c, R, W')$	Action
c	Command
R	Command parameters
W'	Words mapped to action
W	Words mapped to previous actions
$\mathcal{E}$	Environment state

W' underlined, W in grey

Training

- D documents
- Reward function r(h) where histc $h = (s_0, a_0, \ldots, s_{n-1}, a_{n-1}, s_n)$
- Estimate paramete θ of policy $p(a|s, \theta)$
- Log linear model (i.e. linear softmax policy) :

$$p(a|s;\theta) = \frac{e^{\theta \cdot \phi(s,a)}}{\sum_{a'} e^{\theta \cdot \phi(s,a')}}$$

- n-dimensional feature representati $\phi(s, a) \in \mathbb{R}^n$

Reinforcement Learning: Policy gradients

- We maximize $V_\theta(s) = E_{p(h|\theta)}\left[r(h)\right]$
- $p(h|\theta)$: probability of history h starting from state s acting according to policy with parameter θ

$$p(h|\theta) = \prod_{t=0}^{n-1} p(a_t|s_t;\theta)p(s_{t+1}|s_t,a_t)$$

Policy gradient algorithm

$$\frac{\partial}{\partial\theta}V_\theta(s) = E_{p(h|\theta)}\left[r(h)\sum_t \frac{\partial}{\partial\theta}\log p(a_t|s_t;\theta)\right]$$

$$\frac{\partial}{\partial\theta}\log p(a|s;\theta) = \phi(s,a) - \sum_{a'}\phi(s,a')p(a'|s;\theta)$$

Policy gradient algorithm

- Exact derivate oV_θ intractable, expectations requires summation over all histories
- Use stochastic gradient ascent, noisy estimate of expectation
- Draw samples from$p(h|\theta)$ by acting in environment, use to estimate expectation

Input: A document set D,
Feature representation ϕ,
Reward function $r(h)$,
Number of iterations T

Initialization: Set θ to small random values.

for $i = 1 \ldots T$ **do**
  foreach $d \in D$ **do**
    Sample history $h \sim p(h|\theta)$ where
      $h = (s_0, a_0, \ldots, a_{n-1}, s_n)$ as follows:
    for $t = 0 \ldots n-1$ **do**
      Sample action $a_t \sim p(a|s_t; \theta)$
      Execute a_t on state s_t: $s_{t+1} \sim p(s|s_t, a_t)$
    end
    $\Delta \leftarrow \sum_t \left(\phi(s_t, a_t) - \sum_{a'} \phi(s_t, a')p(a'|s_t; \theta)\right)$
    $\theta \leftarrow \theta + r(h)\Delta$
  end
end

Output: Estimate of parameters θ

Algorithm 1: A policy gradient algorithm.

Reward functions

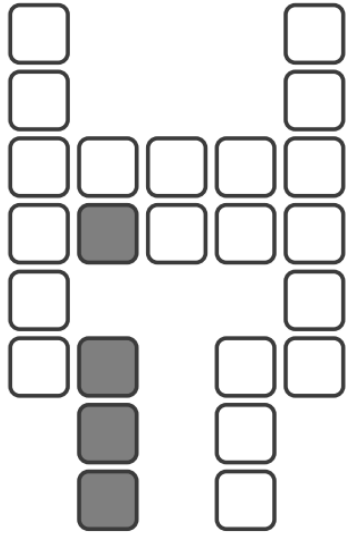
- Case: Every d in D annotated with correct action sequence
  - MLE reward, 1 when h matches annotation, 0 otherwise
  - Policy gradient same as SGD with MLE objective
- Case: No annotations available
  - Use feedback that correlates with action sequence quality

Experiments

- Microsoft Windows Help and Support:
  - Reward function: Environment feedback
  - Noisy reward: check whether execution can proceed from one sentence to next, i.e., at least one word in sentence corresponds to an object in the environment
  - When satisfied, positive reward linearly increases with percentage of words assigned to non null commands

Experiments

- Crossblock puzzle game
  - Reward: easy to directly verify completion, -1 if reach a state where game is unsolvable



Clear the bottom four from the second column from the left, the row of four, the second column from the right, the row of four, then the two columns.

Figure 3: Crossblock puzzle with tutorial. For this level, four squares in a row or column must be removed at once. The first move specified by the tutorial is greyed in the puzzle.

Datasets

	Windows	Puzzle
Total # of documents	128	50
Total # of words	5562	994
Vocabulary size	610	46
Avg. words per sentence	9.93	19.88
Avg. sentences per document	4.38	1.00
Avg. actions per document	10.37	5.86

Results

	Windows				Puzzle		
	Action	Sent.	Doc.	Word	Action	Doc.	Word
Random baseline	0.128	0.101	0.000	—	0.081	0.111	—
Majority baseline	0.287	0.197	0.100	—	—	—	—
Environment reward	$*$ 0.647	$*$ 0.590	$*$ 0.375	0.819	$*$ 0.428	$*$ 0.453	0.686
Partial supervision	$\diamond$ 0.723	$*$ 0.702	0.475	0.989	0.575	$*$ 0.523	0.850
Full supervision	$\diamond$ 0.756	0.714	0.525	0.991	0.632	0.630	0.869

Table 2: Performance on the test set with different reward signals and baselines. Our evaluation measures the proportion of correct actions, sentences, and documents. We also report the percentage of correct word alignments for the successfully completed documents. Note the puzzle domain has only single-sentence documents, so its sentence and document scores are identical. The partial supervision line refers to 20 out of 70 annotated training documents for Windows, and 10 out of 40 for the puzzle. Each result marked with $*$ or $\diamond$ is a statistically significant improvement over the result immediately above it; $*$ indicates $p < 0.01$ and $\diamond$ indicates $p < 0.05$.

Results

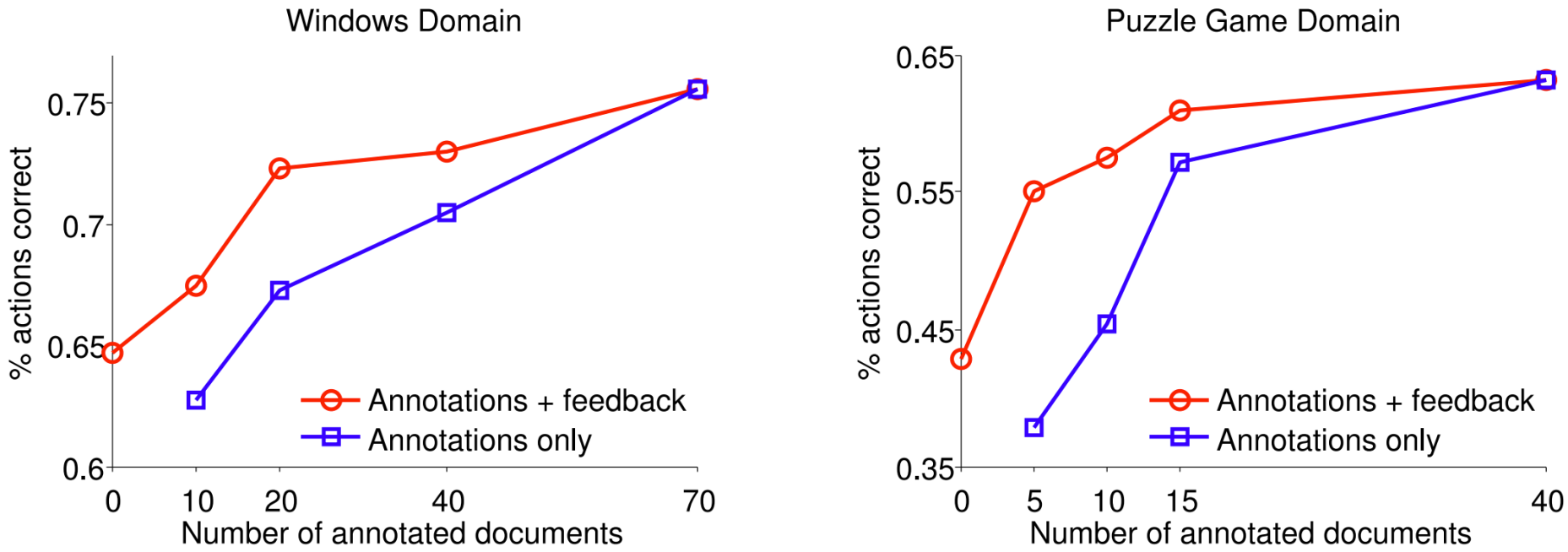


Figure 5: Comparison of two training scenarios where training is done using a subset of annotated documents, with and without environment reward for the remaining unannotated documents.