Reading Between the Lines: Learning to Map High-level Instructions to Commands

S.R.K. Branavan

*Joint work with:* Luke Zettlemoyer, Regina Barzilay

MIT
Mapping Instructions to Commands

**Input**
- **Instructions:** text descriptions of commands
- **Target environment:** where commands need to be executed

**Output**
- **Command sequence executable in the environment**

1. Click **Start**, point to **Search**, and then click **For Files or Folders**.
2. In the **Search for** box, type "msdownld.tmp"
3. In the **Look in** list, click **My Computer**, and then click **Search Now**.
4. ...

```
LEFT_CLICK(Start)
LEFT_CLICK(Search)
...
TYPE_INTO(Search for:,"msdownld.tmp")
...
```
Segment Text into Individual Instructions

Input Document:
"double click system in the control panel, then go to the advanced tab"
Translate Instructions to Command Sequences

Instructions:

"double click system"

"in the control panel"

"go to the advanced tab"

Translated Command Sequence:

double-click System

left-click Start

left-click Settings

left-click Control Panel

left-click Advanced
Permute Command Sequence to Execution Order

Instructions:

"double click system"

"in the control panel"

"go to the advanced tab"

Output Command Sequence:

1. left-click Start
2. left-click Settings
3. left-click Control Panel
4. double-click System
5. left-click Advanced
Previous Work: Low-level Instructions

All commands are explicitly specified

Instructions

"click start"
"click settings"
"click control panel"
"double click administrative tools"
"double click computer management"
"double click device manager"
"under disk drives on the device list,"
"select the primary IDE drive"

Environment Commands

left-click Start
left-click Settings
left-click Control Panel
double-click Administrative Tools
double-click Computer Management
double-click Device Manager
left-click disk drives
left-click Primary HDD
Current Focus: High-level Instructions

Most commands are specified only implicitly

Instructions

"open device manager"
"under disk drives on the device list,"
"select the primary IDE drive"

Environment Commands

left-click Start
left-click Settings
left-click Control Panel
double-click Administrative Tools
double-click Computer Management
double-click Device Manager
left-click disk drives
left-click Primary HDD
Current Focus: High-level Instructions

Most commands are specified only implicitly

Instructions

"open device manager"
"under disk drives on the device list,"
"select the primary IDE drive"

Environment Commands

left-click Start
left-click Settings
left-click Control Panel
double-click Administrative Tools
double-click Computer Management
double-click Device Manager
left-click disk drives
left-click Primary HDD
Interpreting High-level Instructions: An Old AI Problem

Winograd, 1972

Di Eugenio, 1992; Webber, 1995

MacMahon, 2006

Block-worlds

Virtual character animation

Navigation
Interpreting High-level Instructions: An Old AI Problem

Approach: Heavy reliance on domain knowledge

Winograd, 1972

Di Eugenio, 1992; Webber, 1995

MacMahon, 2006

Block-worlds

Virtual character animation

Navigation
Interpreting High-level Instructions: An Old AI Problem

Winograd, 1972

Block-worlds

Di Eugenio, 1992; Webber, 1995

Virtual character animation

MacMahon, 2006

Navigation

Approach: Heavy reliance on domain knowledge

manually specified
Humans Interpretation of High-level Instructions

High-level instruction: “open control panel”

1. left-click Start
2. left-click Settings
3. left-click Control Panel
Environment Model as a State Transition Graph
Environment Model as a State Transition Graph

- Extremely large graph
- Need to learn a relevant portion of graph
Key Idea

Ground language interpretation in environment dynamics

1. **Automatically construct a relevant environment model**

2. **Incorporate environment model into language interpretation algorithm**
Reinforcement Learning Framework for Mapping Low-level Instructions

(ACL 2009)
Reinforcement Learning for Instruction Mapping

(State $s$)

Observed text and environment

Select run after clicking start. In the open box type "dcomcnfg".
Reinforcement Learning for Instruction Mapping

(State $s$

*Observed text and environment*

Select run after clicking start. In the open box type "dcomcnfg".

Action $a$

*word span: clicking start*

*command: LEFT_CLICK(start)*

*Policy function*

$p(a | s)$
Reinforcement Learning for Instruction Mapping

(ACL 2009)

State $s$

Observed text and environment

Select run after clicking start. In the open box type "dcomcnfg".

Action $a$

word span: clicking start
command: LEFT_CLICK( start )

Policy function

$p(a | s)$

State $s'$

Observed text and environment

Select run after clicking start. In the open box type "dcomcnfg".
Reinforcement Learning for Instruction Mapping

Repeat:
- Observe current state of text + environment
- Select action based on policy function
- Execute action
- Receive reward and update parameters of policy function $p(a|s; \theta)$

*ACL 2009*
**Policy Function Parameterization**

**Represent each action with a feature vector:**

- **action** $a$
- **state** $s$

Select run after clicking start. In the open box type "dcomconfig".

- e.g., Binary features on [word, command] pairs

**Define policy function as a log-linear distribution:**

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

$\theta$ - parameters of model

$\phi(s, a) \in \mathbb{R}^n$ - real valued feature function on state $s$ and action $a$
Limitations of Framework

Every command is explicitly specified in the text

→ *Cannot handle high-level instructions*

Command selection depends only on current state

→ *Cannot condition on future states*
Reinforcement Learning Framework for Mapping High-level Instructions

*(current work)*
Highlights of Solution

One-to-many mappings:

1. Allow mapping of empty text to commands

   text: "open the control panel"

   Word spans

   "" --- > left-click Start

   "" --- > left-click Settings

   "open the control panel" --- > left-click Control Panel

   • Uncertainty increases dramatically
   • Search space gets much larger
Highlights of Solution

One-to-many mappings:

2. Conditioning on future states & unmapped text

- Cannot learn a complete environment model,
  Need to learn a relevant partial model

Text: "open the control panel"

Diagram:

- Starting environment state
- Conditions: c₁, c₂, c₃, c₄
- Environment state containing label "control panel"
Acquiring a Partial Environment Model

Input documents

- previously unseen state
- previously seen state
- newly seen state
- current trace
Acquiring a Partial Environment Model

Interpretation 1

start state

- previously unseen state
- previously seen state
- newly seen state
- current trace
Acquiring a Partial Environment Model

Interpretation 2

- previously unseen state
- previously seen state
- newly seen state
- current trace
Acquiring a Partial Environment Model
Acquiring a Partial Environment Model

start state

○ previously unseen state
● previously seen state
★ newly seen state
→ current trace
Acquiring a Partial Environment Model

Partial environment model

- previously unseen state
- previously seen state
- newly seen state
- current trace
Acquiring a Partial Environment Model
Acquiring a Partial Environment Model
Exploiting The Partial Environment Model

*Select commands using information about future states*

Encode future state attributes as features, e.g.:
- Average reward
- Length of path to potential goals

\[
\phi(a, s) \quad \rightarrow \quad \phi(a, s, q)
\]

\[
p(a \mid s; \theta) \quad \rightarrow \quad p(a \mid s; q, \theta)
\]

\[q\] - partial environment model
Policy Function

Policy for selecting actions:

\[ p(a | s; q, \theta) = p(\vec{w} | s; \theta_w) \times p(c | \vec{w}, s; q, \theta_c) \]

- \( s \) - state
- \( a \) - action
- \( \theta \) - model parameters
- \( c \) - environment command
- \( \vec{w} \) - Instruction word span
- \( q \) - partial environment model
Features from Environment Model

1. Highest reward achievable from current state by any action sequence

2. Length of highest-reward action sequence

3. Average reward received at environment state for any document

text: "open the control panel"

![Diagram showing an environment model with nodes and actions labeled c1, c2, c3, c4, and connections to different states, with one state labeled "control panel" and another labeled "state with low average reward".]
Example Local Features

**Features on words and environment command & object**
- Binary feature on each (word, command) pair
- Binary feature on each (word, object type) pair

**Features on environment objects**
- Object is in foreground
- Object was previously interacted with
- Object became visible after last action

**Features on words**
- Word type
- Distance from last used word

*Total number of features: 4438*
Windows Configuration Application

Windows 2000 help documents from support.microsoft.com

Total # of documents | 188
# documents with high-level instructions | 60
Total # of words | 7448
Vocabulary size | 739
Avg. commands per document | 10

Evaluation Metric: Command prediction accuracy on heldout documents
1. How important is the environment model for interpreting high-level instruction?

*Condition only on current state*

*Condition on current and future states*

Evaluate against method without environment model
Evaluation

2. How useful is document information in constructing the environment model?

*Compare against method with uniformly sampled environment model*
Results: High-level Instructions

Command accuracy

- No environment model: 2%
- Uniformly sampled environment model: 28%
- Our approach: 62%

High-level instructions
Automatic Paraphrasing of High-level Instructions

*High-level instruction*
- open **device manager**

*Low-level instruction paraphrase*
- double click **my computer**
- double click **control panel**
- double click **administrative tools**
- double click **computer management**
- double click **device manager**
Conclusions

1. Domain knowledge is essential for high-level instructions

2. Relevant domain knowledge can be acquired automatically

Code and data available at:
[groups.csail.mit.edu/rbg/code/](http://groups.csail.mit.edu/rbg/code/)