Teaching Intelligent Agents New Tricks: Natural Language Instructions plus Programming-by-Demonstration for Teaching Tasks

Brad A. Myers and Toby Jia-Jun Li
Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
http://www.cs.cmu.edu/~bam
bam, tobyli @cs.cmu.edu
Twitter: @bradamyers @TobyJLi
Conversational Agents

- Ubiquitous and growing
- But limited
- Can’t do everything that apps can
- No one has yet talked about extending what AI agents can do
Back-end Services → API call → Agent → Conversation → GUI → User
Extensibility Missing

- Only extensible with 3rd party “skills”
Need New Skills

- The “long tail” of apps and tasks are unlikely to be supported in the foreseeable future due to the engineering efforts required.

Xu et al. Identifying diverse usage behaviors of smartphone apps. IMC ’11
Need Personalization

- Users also may have **highly personalized** and **diverse** preferences on how tasks should be performed
- Even for existing skills

I like hot coffee when it’s cold outside and iced coffee when it’s hot outside. I would want to have a breakfast sandwich on days when I have early meetings
Support EUD through PBD + CUI

• Need to allow users to create their own customizations
  – End-User Development
    • Also called End-User Programming = EUP

• Our Approach: Combine:
  – Programming by Demonstration (PBD)
  – Conversational User Interface (CUI)
Other Approaches for End-User Development

Automate

HiroMacro

IFTTT

if email then
Add receipts & orders to Receipt Spreadsheet

if Google Drive then
Add all new Pocket notes to Evernote default notebook

if news feed then
Text me Bible Gateway's Verse of the Day

© 2018 – Carnegie Mellon University
Want end user development (EUD)

API call

Agent

Conversation

Back-end Services

GUI

User

No open API available

Lack of technical expertise
Back-end Services

Agent

Direct Manipulation at Runtime

Semantic Understanding

API call

GUI

Demonstration

User
GUI Demonstration + Verbal Instruction
Programming by Demonstration

- Use Programming by Demonstration (PbD) to let the end-users create their own, custom behaviors and scripts
  - Also called “Programming by Example” - PbE
- Users create programs by performing the steps by example
- Assumes user knows how to do the problem concretely with the UI
- System “watches what you do” with the interactive interface
- Records the sequence of operations, data
- Generalizes the program so that you can use an analogous procedure with new situations and data
Concrete User Interface

- Demonstrate how to perform the new tasks on an Android smartphone
- Leverages millions of existing mobile apps & user knowledge on how to operate these apps
- Use the Android accessibility API to monitor and remote-control apps
Prior Work on PbD

- Lots of prior work on PbD
- Grew out of early AI work
- Early research summarized in 2 books
  - Cypher, 1993
  - Lieberman, 2001
  - We authored many chapters
- Many PbD systems from my group over the last 30 years
  - Peridot, Lapidary, Jade, Gilt, Tourmaline, C32, Pursuit, Gold, Marquise, Katie, Turquoise, Topaz, Gamut, Agate, Lapis, Playbook, Graphite, Euklas
Only a few PbD Successes

- Formatting by example in Microsoft Word
  - E.g., paragraph numbering, bullets, indenting, Styles
- Excel value extend & “flash fill”
- Parallel edit of multiple items at once
  - E.g., of variable name refactoring in IDEs
  - From [Miller&Myers IUI'2002]
- Macros by example
  - But rarely used
Key Issues with PBD

• The Generalization Problem
  – How do I describe actions and data?
    • “Data Description problem”
  – How to find the corresponding data at run time?
  – How to infer control structures?
    • Multiple examples?

• Feedback
  – How does the system show the user what has been learned?

• Advice & Editing
  – How can the user influence or repair what the system learns or edit the program later?

• Giving Good Examples is Hard
  – People are not good at giving examples that cover a space of possibilities

• Scope & Utility
  – What can be programmed?
  – Does PBD support programming at the right level of concepts?
Further issues with EUD

- For end-user development in general, further barriers
- Alan Blackwell’s “Attention Investment Model” [HCC’2002]
  - Effort involved in the programming vs. just doing the task by hand
  - Consider the risk that the effort will be wasted or data will be damaged because the program doesn’t work
PBD & AI

• PBD is “shallow learning”
  – Opposite of “deep learning”

• Machine Learning (ML) from one example and little context
  – Narrow domains (long tail)
  – Little opportunity to leverage big data

• Explicit instructions
  – Most ML assumes *implicit* instructions
  – But people instruct each other all the time
  – Research needed on how users can directly *instruct* agents
Sugilite

- Smartphone Users Generating Intelligent Likeable Interfaces Through Examples
  - PhD work of Toby Jia-Jun Li

- Goals:
  - **Applicability**: works on any 3rd party apps
  - **Generalizability**: creates generalized scripts with parameters
  - **Usability**: usable for end users with no prior programming knowledge
  - **Robustness**: reliably perform the demonstrated task in different conditions, and allow end users to “fix” the script when new situation arises
Multi-Modal Interface

- Collaborating with Tom Mitchell and others in Machine Learning Department (MLD) at CMU
  - Part of the InMind project
- Google’s Speech API
- MLD’s “Learning by Instruction Agent (LIA)”
  - Teach agent by explicit verbal instructions

- Trigger the scripts from speech (*ISEUD’2017 & CHI’2017*)
  - Detects parameters
- Control data descriptions with speech (*submitted to VL/HCC’2018*)
Conversation

• Assume user is available so do not have to solve everything up front
• User only gives one example
  – User gives other examples only as needed
• If a script fails, then the user is there to repair
  – So no need for negative examples in advance
    • Research has shown that people are bad at negative examples
• If an example is ambiguous, can use multiple rounds to clarify
  – Sugilite asks more specific questions
SUGILITE: Creating Multimodal Smartphone Automation by Demonstration

1Toby Jia-Jun Li, 2Amos Azaria, 1Brad A. Myers
1Human-Computer Interaction Institute, Carnegie Mellon University
2Computer Science Department, Ariel University
	{tobyli, bam}@cs.cmu.edu, amos.azaria@ariel.ac.il

Available on YouTube
Script Generalization

“Order a Spicy Italian Pizza”
Handling Ambiguity

Recording

Ambiguous!
Robustness

“Request an Uber”

Script Execution Exception
Can't find the target UI element in the current screen for executing the next operation: Click on the ImageView object "Home" in Uber

Create Fork
Do you want to replace the corresponding part in the original script or to create a fork?

CREATE FORK  REPLACE
Old Script Representation

- **Action, target type, target identifier, location**
- “Try-Catch” conditionals
**IoT integration**

- Smart Home / IoT Devices
- Trigger scripts from IoT
  - Notifications
  - IFTTT integration

---


© 2018 – Carnegie Mellon University
Current Work: Ambiguity

Click on?

- Charlie Palmer Steak
- The least expensive steakhouse near me
- Closest one in Midtown East
- The one with 1,000 bonus points
- A promoted restaurant
- The second restaurant in the list

Idea: verbalize the intent
mTurk study

• Are end users able to provide useful and generalizable explanations of intentions for ambiguous demonstrated actions?

<table>
<thead>
<tr>
<th></th>
<th>With Instructions</th>
<th>Without Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Yes</strong></td>
<td>100%</td>
<td>88%</td>
</tr>
<tr>
<td><strong>No</strong></td>
<td>0%</td>
<td>12%</td>
</tr>
</tbody>
</table>

• $n = 45$

© 2018 – Carnegie Mellon University
New Version: Demo

- New user interface
- Asks user to explain *intent* for each selection
- Still asks for confirmation of every step
Order a cappuccino
Book a hotel in London
Check the score for Pittsburgh Penguins
Turn on the living room light
Order a large pepperoni pizza
a. User demonstrates the action directly on unmodified GUIs of third party apps

b. APPINITE asks the user to describe intentions for actions

c. Multi-turn conversations help users refine ambiguous descriptions
Hypothesis: GUI elements will constrain the language used
Current work: Broader vocabulary
Current Work

1. More complex control structures

I like hot coffee when it’s cold outside and iced coffee when it’s hot outside. I would want to have a breakfast sandwich on days when I have early meetings.

2. Foraging and modifying scripts

Get my regular breakfast order at Starbucks, except instead of iced coffee, I want hot coffee.

3. Script Sharing between Users
AI Issues being Investigated

- How can the user know what can be said?
  – Whatever the apps on the smartphone can do
- How can the user know what vocabulary can be used?
  – The words and concepts in those apps
- What kinds of conditionality (control structures) can be expressed?
- What is the “program” going to do?
  – Feedback shown in structured steps
- How can the user influence and repair what the system learns, and edit the program?
Teaching Intelligent Agents New Tricks: Natural Language Instructions plus Programming-by-Demonstration for Teaching Tasks

Brad A. Myers and Toby Jia-Jun Li
Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
http://www.cs.cmu.edu/~bam
bam, tobyli @cs.cmu.edu
Twitter: @bradamyers @TobyJLi

Acknowledgements:
This work was supported in part by the
InMind project
Future funding: New NSF Grant!

© 2018 – Carnegie Mellon University