Interactive Semantic Parsing for If-Then Recipes via Hierarchical Reinforcement Learning

Ziyu Yao[1], Xiujun Li[2,3], Jianfeng Gao[2], Brian Sadler[4], Huan Sun[1]

[1]The Ohio State University, [2]Microsoft Research, [3]University of Washington, [4]U.S. Army Research Lab

INTRODUCTION

If-Then Recipes/Programs

A conditional statement of "*If This, Then That*": whenever the *trigger* condition ("*This*") is satisfied, the *action* ("*That*") will be performed.

4 Components:
Trigger Channel ("tc"), Trigger Function ("tf"), Action Channel ("ac"), Action Function ("af").

Semantic Parsing for If-Then Recipes

Parsing a natural language (NL) description to a corresponding If-Then recipe.

Example:
"*Create a link note on Evernote for my liked tweets*" →
[tc: `Twitter`, tf: `New liked tweet by you`, ac: `Evernote`, af: `Create a link note`]

Application:
Widely adopted for **Task/Routine Automation** and **Smart Home**: "Text me if the door is unlocked", "Send me the weather report every day at 7AM", etc.

MOTIVATION

Description Ambiguity

An NL description can be *ambiguous* or contain *incomplete* information.

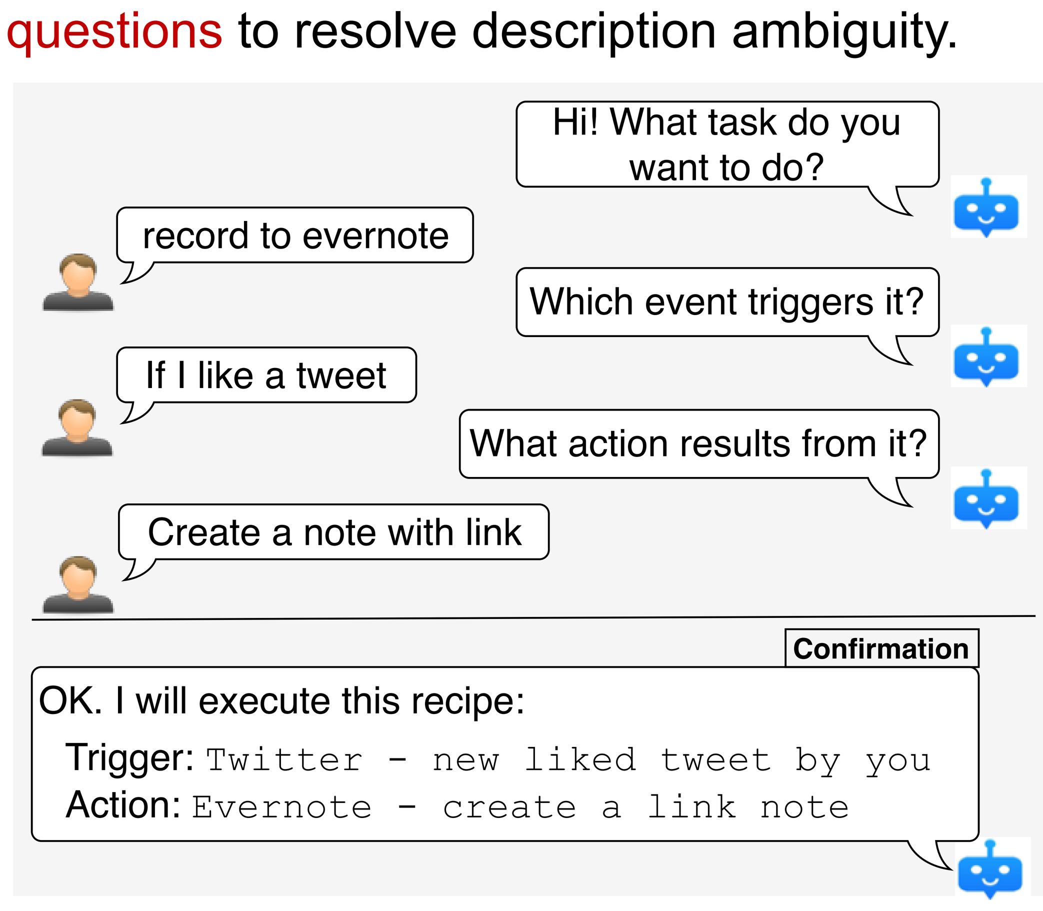
User: "record to evernote"
Ground-truth recipe: [tc: `Twitter`, tf: `New liked tweet by you`, ac: `Evernote`, af: `Create a link note`]
(Liu et al. 2016): [tc: `Phone Call`, tf: `Leave IFTTT any voicemail`, ac: `Evernote`, af: `Append to note`]

- ❖ Based on ~4K recipes collected from real users [1], **80%** of recipe descriptions are ambiguous!
- ❖ May fail a well-trained semantic parser.

INTERACTIVE SEMANTIC PARSING

Our Solution: Ask Human Questions

An intelligent agent can ask clarifying questions to resolve description ambiguity.

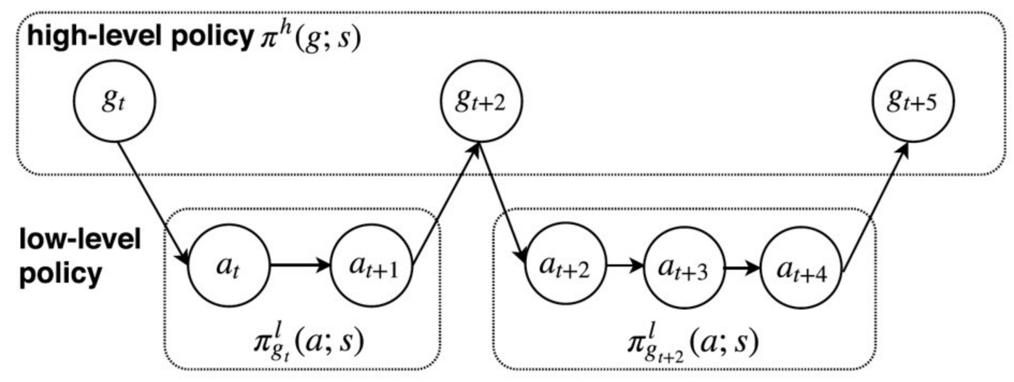


User answers (in NL) are received and utilized for the agent's prediction.

Our Aim:
Improve parsing accuracy with minimal questions, without supervision on when/what to ask.

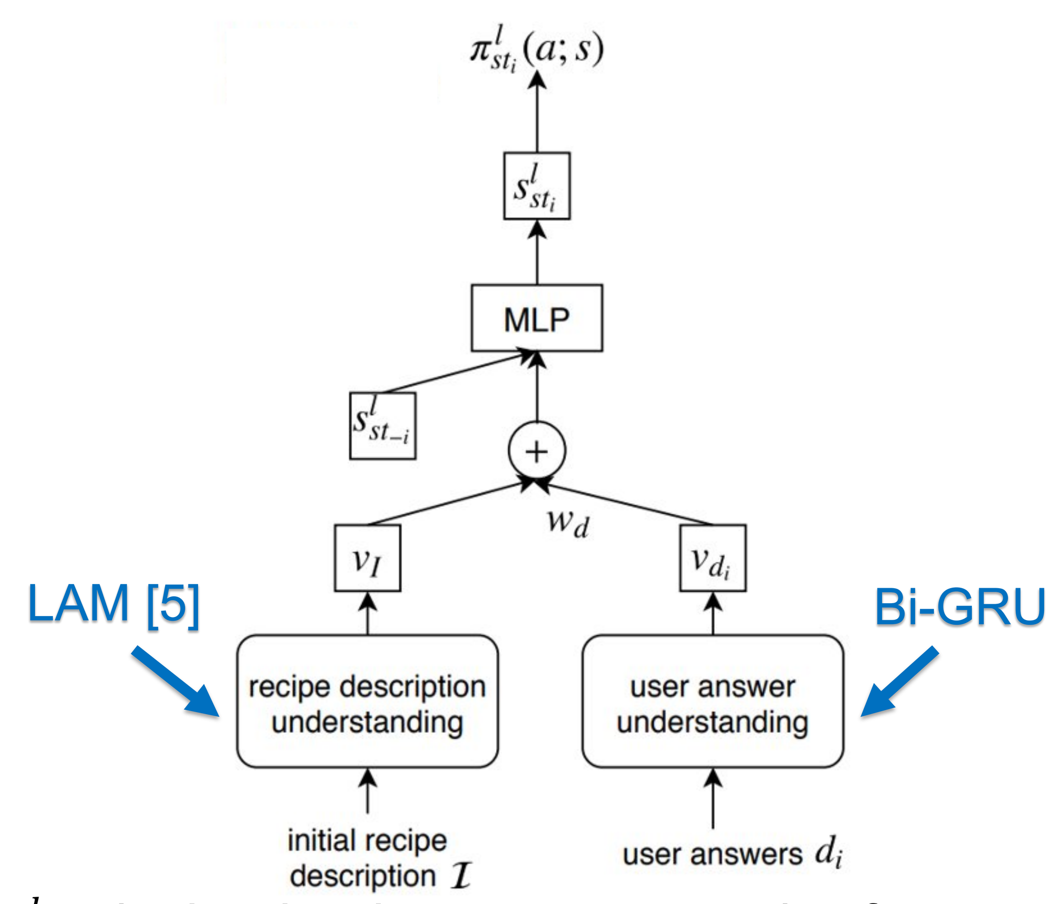
HIERARCHICAL RL (HRL)

Hierarchical Policy [2]



- ❖ *High-level* policy selects a *subtask g_t (i.e., completing one of the 4 components)* to work on.
- ❖ *Low level* policy completes each subtask by taking actions to *either make a prediction or ask user*.
- ❖ Semantic parsing = a sequence of high/low-level decisions.

Low-Level Policy Function:

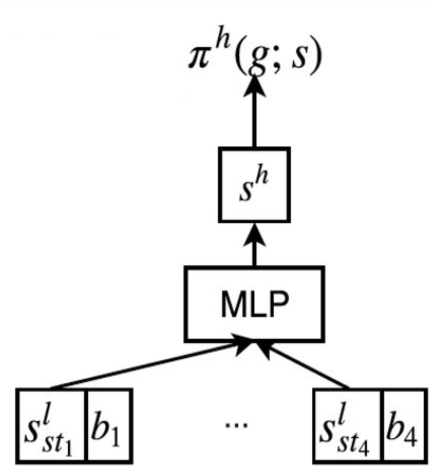


$s^l_{st_i}$: the low-level state representation for subtask st_i.

$$s^l_{st_i} = \tanh(W_{c_i}[s^l_{st_1}; \dots; s^l_{st_{i-1}}; v_i; s^l_{st_{i+1}}; \dots; s^l_{st_4}])$$

We define one policy for *each* subtask.

High-Level Policy Function:



b_i: whether the subtask st_i has been predicted

Training by Rewarding

Low-level reward (when taking action a_t for subtask g_t):

$$r^l_{g_t}(s_t, a_t) = \begin{cases} 1 & \text{if } a_t = \ell_{g_t} \\ -\beta & \text{if } a_t = \text{AskUser} \\ -1 & \text{otherwise} \end{cases}$$

l_{g_t}: true label of subtask g_t. $-\beta$ is the penalty for asking questions.

High-level reward for g_t = accumulative low-level reward for completing g_t.

Optimization:
Maximize accumulative high/low-level reward via REINFORCE [3].

EXPERIMENTS

Experiment Setup

Dataset:
- ❖ Training: 291,285 <NL description, Recipe> pairs from [4].
- ❖ Testing set collected & annotated by [1]:

Test Data	CI	VI		Total
		VI-1/2	VI-3/4	
Size (%)	727 (18.79)	1,271 (32.84)	1,872 (48.37)	3,870 (100)

- **CI**: recipes with descriptions clear in all 4 subtasks for annotators.
- **VI-1/2**: recipes containing 1 or 2 vague subtasks for annotators.
- **VI-3/4**: recipes containing 3 or 4 vague subtasks for annotators.

User Simulation:
We resort to *user simulator* to train the agent, with simulated user answers extracted from training set using templates.

Methods to Compare:
- ❖ **LAM** [5]: state-of-the-art, non-interactive.
- ❖ **LAM-rule**: rule-based agent, ask user when prob of prediction is lower than 0.85
- ❖ **LAM-sup**: agent with "AskUser" action, trained via SL on pseudo labels.
- ❖ **HRL**: our proposed agent trained via RL.
- ❖ **HRL-fixedOrder**: HRL with a fixed high-level order to predict tc – tf – ac – af, following previous work (e.g., [6]).

Metrics:
- ❖ C+F Acc: accuracy when all 4 components are correct.
- ❖ #Ask: number of clarifying questions.

Simulation Evaluation

Model	CI		VI-1/2		VI-3/4	
	C+F Acc	#Asks	C+F Acc	#Asks	C+F Acc	#Asks
LAM	0.801	0	0.436	0	0.166	0
LAM-rule	0.897	1.433	0.743	2.826	0.721	5.568
LAM-sup	0.894	**0.684**	0.803	**1.482**	0.780	2.921
HRL-fixedOrder	**0.950**	1.522	0.855	1.958	0.871	2.777
HRL	0.949	1.226*	**0.888***	1.748*	**0.878***	**2.615***

(when interacting with user simulator)

Human Evaluation

- ❖ Interact with real humans.
- ❖ Collecting 496 conversations on VI-3/4.

Model	C+F Acc	#Asks
LAM	0.206	0
LAM-rule	0.518	2.781
LAM-sup	0.433	2.614
HRL-fixedOrder	0.581	2.306*
HRL	**0.634***	**2.221***

Conclusions

- ✓ Interactive > Non-interactive.
- ✓ Rule-based agent tends to ask redundant questions.
- ✓ HRL vs. HRL-fixedOrder: HRL achieves *significantly better performance with fewer questions*.

[code available online]

Acknowledgement

Ohio Supercomputer Center
An OH·TECH Consortium Member
FUJITSU

REFERENCES

[1] Quirk, C.; Mooney, R. J.; and Galley, M. 2015. Language to code: Learning semantic parsers for if-this-then-that recipes. In *ACL*.
[2] Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112(1-2):181–211.
[3] Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*.
[4] Ur et al., 2016. Triggeraction programming in the wild: An analysis of 200,000 ifttt recipes. In *CHI*.
[5] Liu et al., 2016. Latent attention for if-then program synthesis. In *NIPS*.
[6] Beltagy, I., and Quirk, C. 2016. Improved semantic parsers for if-then statements. In ACL, volume 1, 726–736.