ChatIoT: Zero-code Generation of Trigger-action Based IoT Programs

Yi Gao, Kaijie Xiao, Fu Li, Weifeng Xu, Jiaming Huang, **Wei Dong**
♠College of Computer Science and Technology
♣Zhejiang University

Proc. of IMWUT/UbiComp 2024

Presented by Lili Pan

Authors Team

Research Interests

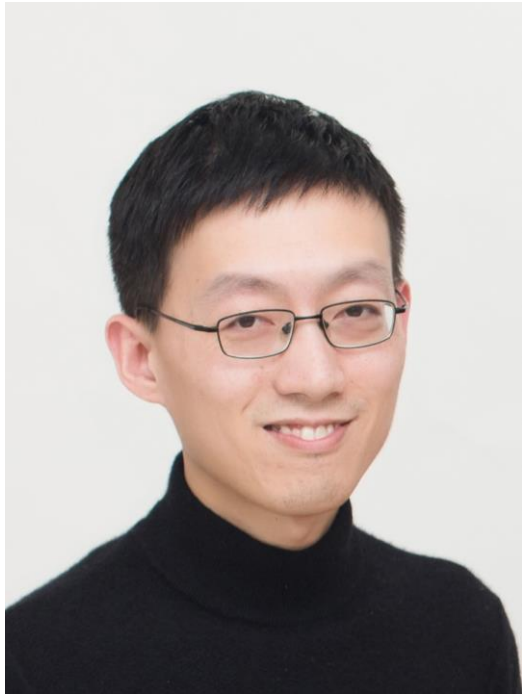
- **智慧物联网AIoT**
- 边缘计算和边缘 AI
- 无线网络和物联网安全

Recent Projects

- 物联网的机密传感计算
- **AI 赋能的 IoT**

（如何为资源受限的 IoT 和边缘设备定制复杂的 AI 模型）

- 低功耗无线
- **低代码 IoT 编程（如何采用 LLM 进行物联网开发）**
- 新颖的物联网系统和应用程序



Wei Dong’ s homepage (emnets.cn)

Outline

1 Background

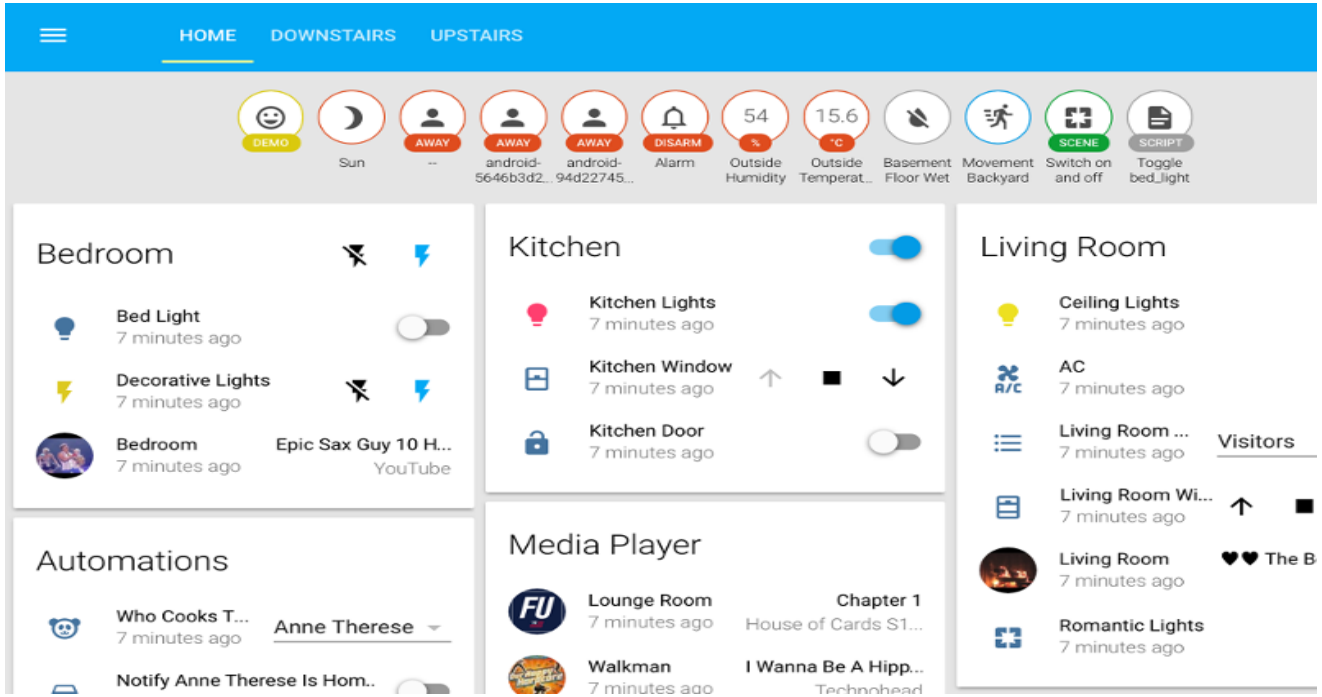
2 Related work

3 Design

4 Experiments

5 Conclusion

Background



If trigger, then action. 简称TAP

```
rule "Tests for type1 machine"
salience 100
when
   machine : Machine( type == "Type1" )
then
   Test test1 = testDAO.findByKey(Test.TEST1);
   Test test2 = testDAO.findByKey(Test.TEST2);
   Test test5 = testDAO.findByKey(Test.TEST5);
   machine.getTests().add(test1);
   machine.getTests().add(test2);
   machine.getTests().add(test5);
   insert( test1 );
   insert( test2 );
   insert( test5 );
end
```

Background

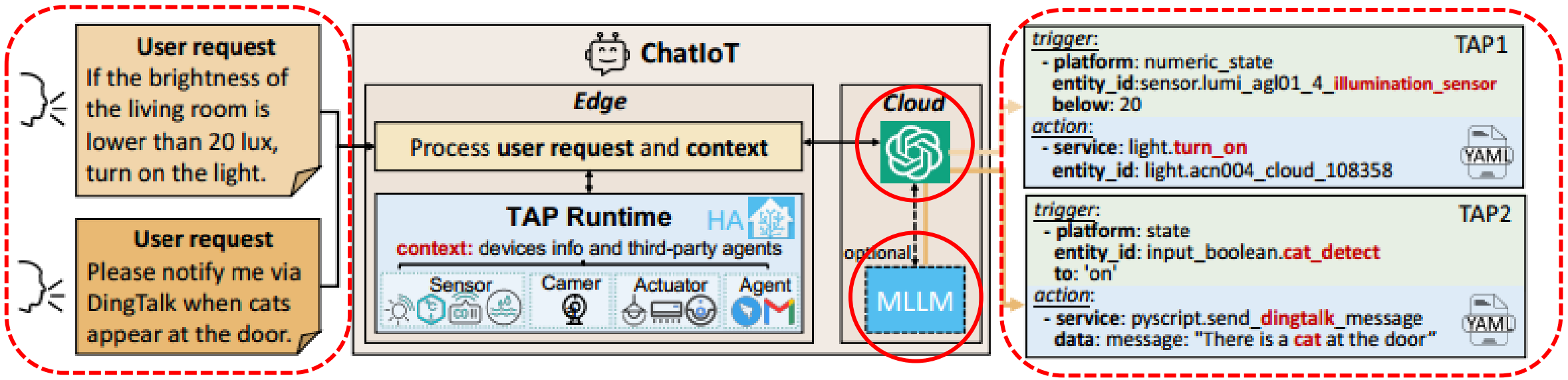


Fig. 1. ChatIoT usage.

Input		Output
user request-->	user request+home information-->cloud	direct TAP rules
natural language	ChatIot+LLM(MLLM)-->TAP generation	

Outline

1 Background

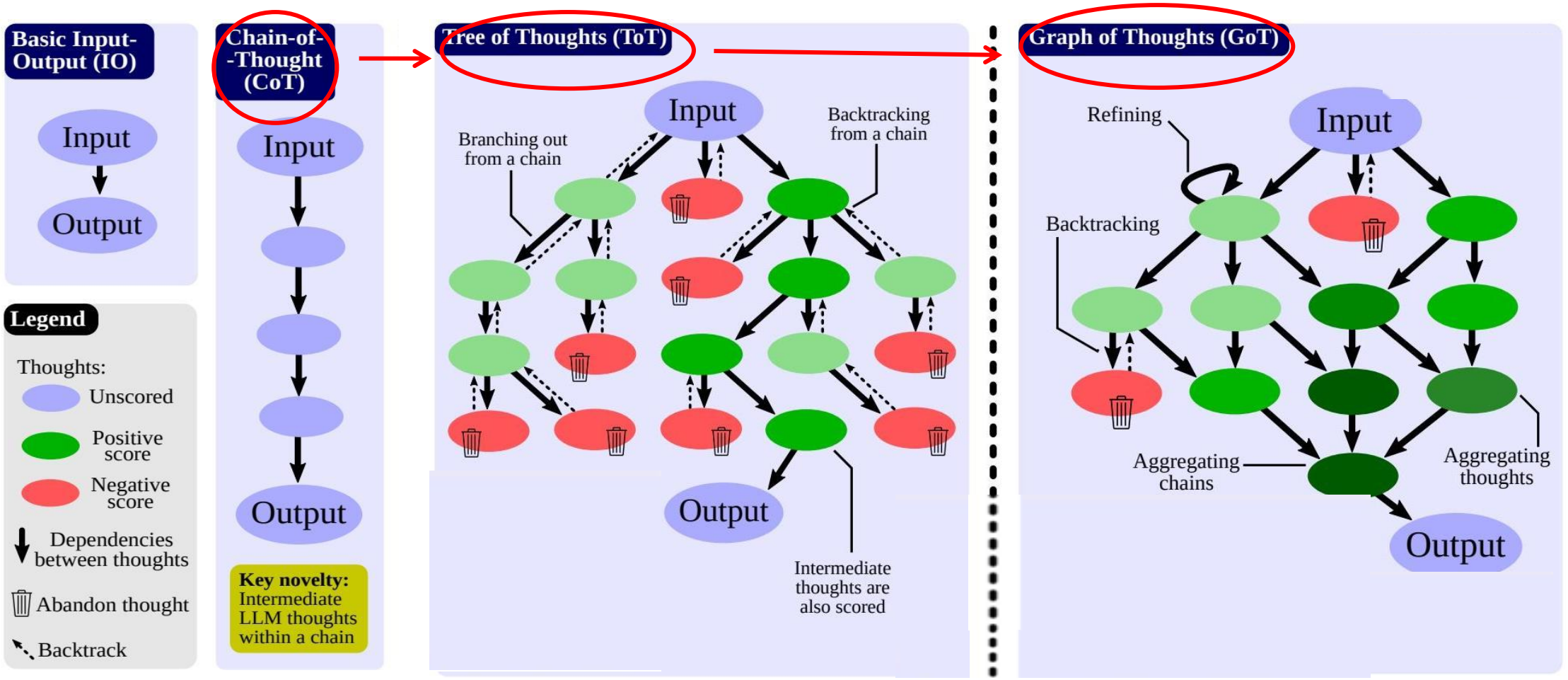
2 Related work

3 Design

4 Experiments

5 Conclusion

Prompting Schemes



Prompting Schemes

ReACT

问题	科罗拉多造山带东部区域延伸到的区域的海拔范围是多少?
思考1	我需要搜索科罗拉多造山带，找到科罗拉多造山带东段延伸的区域，然后找到该区域的海拔范围。
动作1	搜索 [科罗拉多造山带]
观察1	科罗拉多造山带是科罗拉多及其周边地区造山运动的一段。
思考2	它没有提到东部地区。所以我需要查一下东部地区。
动作2	查找 [东部地区]
观察2	东部区域延伸至高平原，称为中原造山带。
思考3	科罗拉多造山运动的东部区域延伸到高平原。所以我需要搜索高平原，找到它的海拔范围。
动作3	搜索 [高平原]
观察3	高平原指的是两个截然不同的陆地区域之一

. . .

Home Automation Approaches

Table 1. Difference between existing home automation approaches and ChatIoT

Method	User learning cost	User interaction	Model customization
In-situ programming [40, 43, 48]	High	✓	×
Trace-driven programming [19, 39, 60]	Low	×	×
ChatIoT (Ours)	Zero	✓	✓

In-situ programming: Allow users ***Interact directly*** with intelligent devices or systems in a specific physical environment

Trace-driven programming: Synthesize TAP rules from **traces** or records of historical operations and events

Edge Model Deployment Schemes

What Edge Model Deployment?

Install the machine learning model on the device, deploying a model to a device

Why Edge Model Deployment?

Real-time response、Privacy protection、Cost effectiveness

Challenges?

Resource Constraints、Model size

Solutions

Model quantification、Model pruning、Knowledge Distillation

Model merging

Multiple models synthesized into a more compact model

Chatlot’s DRL-based model customization !

Outline

1 Background

2 Related work

3 Design

4 Experiments

5 Conclusion

1. Context-aware Compressive Prompting

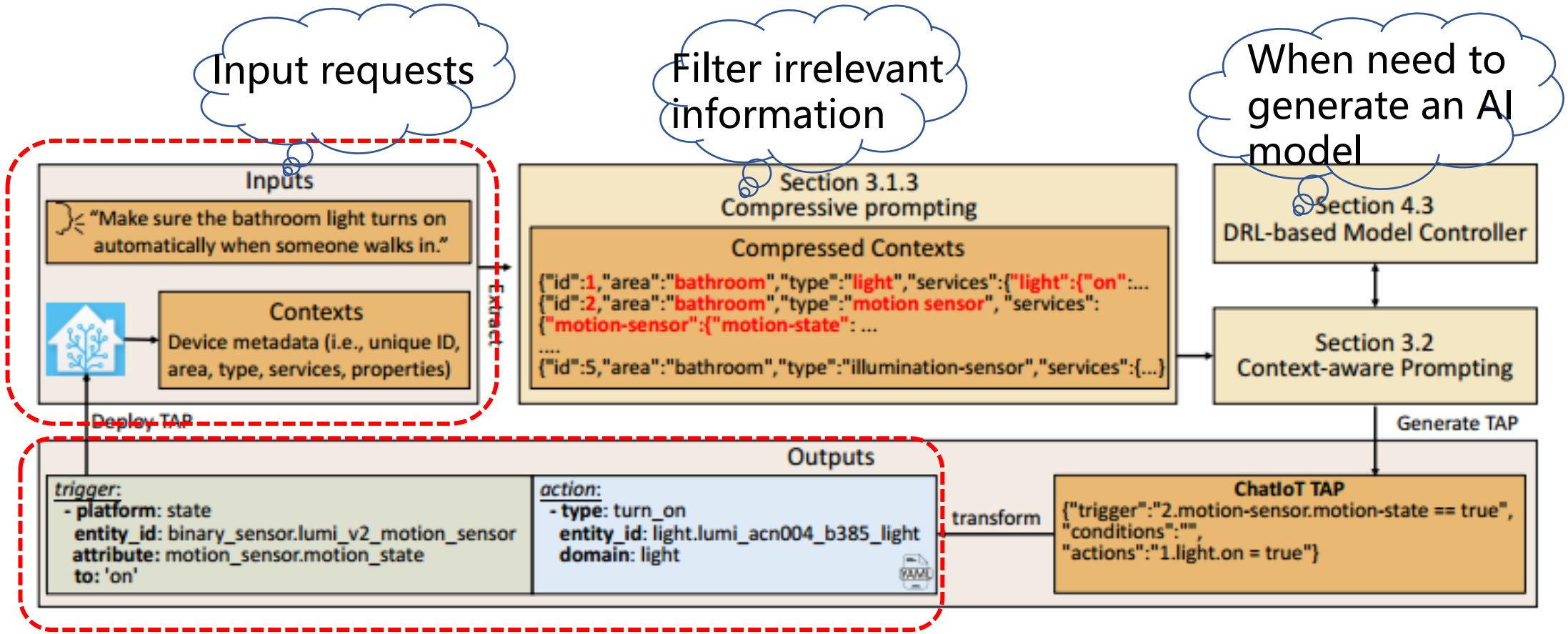
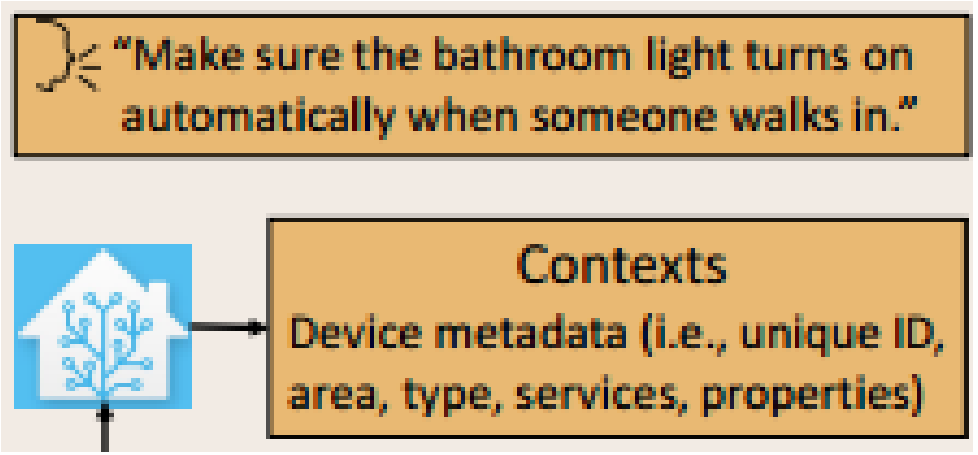


Fig. 2. Workflow of TAP generation.

1. Context-aware Compressive Prompting



Input

User request + Contexts

(the sensors and actuators in the home)

Algorithm 1: Compress context

Input: TAP generation request Q, Device list $D = [d_1, d_2, d_3, \ldots]$ in context

Output: Compressed device list $D_{\text{compressed}}$

Compress context

Input: request Q + device D

Output: $D_{\text{compressed}}$

ChatIoT TAP

{"trigger":"2.motion-sensor.motion-state == true",
"conditions":"",
"actions":"1.light.on = true"}

Output

IF <triggers> THEN <actions>

1. Context-aware Compressive Prompting

Ask LLM to generate TAPs from the user request and the context

(zero-shot prompting) ×

Reasons?

- Too many devices, increasing the **token cost** significantly
- Such complex information leads to low **TAP generation accuracy**
- Lack of information , e.g. the exact room temperature

Solution

- Extract key information

  Lamp in a bathroom --> “area: bathroom, type: light, service: light, fan”

1. Context-aware Compressive Prompting

Preprocessing

↓

Creating Service

↓

Generating TAP

↓

Evaluating TAP

#Role
You are the preprocessor of a smart assistant, find the properties in the context that are related to the user request. Make sure you dont miss any properties that are related to the user request.
 # Input
1. User request.
2. Context: it contains information about all devices, including their id, area, type, and services. Each device may contain multiple services. Each service may contain multiple properties.
 # Workflow
Solve this task with interleaving Thought, Action, Observation steps. Your response will be a json {"Thought": <Thought>, "Action": <Action>}. Action can be two types:
(1) CreateService: Only camera can be added a new service. Parameter is a json include "service" field which is also a json include "camera_id", "service_name", "description", "question" fields. You must remember that camera now does not have any service.
(2) Finish: you return the property list and finish the task. Parameter is json include "property_list" field. "property_list" is a list of properties involved. Make sure the properties are expressed in the correct format "id.service.property".
 # Output
Your response will be a json {"Thought": <Thought>, "Action": <Action>}.

1. Context-aware Compressive Prompting

Preprocessing

ChatIoT: TAP generation request : When a person appear at the door, turn on the light.
Context:[{"id":1, "area": "doorway", "type": "camera", "services":{"dog_detect":{"state":{}}}},
{"id":2,"area":"doorway", "type": "light", "services": {"light":{"on":{},"brightness":{}}, "light-extension": {"brightness-delta":{}}}}...]
LLM: Thought 1: Identify the trigger, conditions, and the actions from the user description. Trigger is "when a person appear at the door" , Action is "turn on the light". Find device and service. For trigger, I can use camera (id:1) to detect person, but there is no service about person detect in it. For action, turn on the light (id:2) at door by the property "on" of service "light".
Action 1: CreateService [{"id":1, "service": "person_detect" , "description": "detect person", "question": "Is there person"}]
ChatIoT: Observation 1: UPDATE CONTEXT {"id":1, "area":"doorway", "type":"camera", "services": {"dog_detect":{"state":{}},"person_detect":{"state":{}}}}
LLM: Thought 2: so I can use camera (id:1) to detect person by property "state" of service "person_detect".
Action 2: Finish[["1.person_detect.state", "2.light.on"]]

Creating Service

ChatIoT: Service request : {"id":1, "service": "person_detect", "description": "detect person", "question":"Is there person"}
Context :[{"id":1, "area": "doorway", "type": "camera", "services": {"dog_detect": {"state": {"description": "detect dog", "model_id":1, "format": "bool","access": ["read"]}}}}]
model zoo: [{"model_id":1,"description":"dog detect"}]
LLM: Thought 1: The User want to add "person_detect" service to the camera (id:1). Since there is no model for person detect in model zoo, I need to request a new model for this service.
Action 1:RequestModel["Is there person"] ↔ Model Controller
ChatIoT: Observation 1: UPDATE CONTEXT {
model zoo: [{"model_id":1,"description":"dog detect"}, {"model_id": 2, "description":"person detect"}]}
LLM: Thought 2: add "person_detect" service based on person detect model(model_id:2)
Action 2: AddService[{"id":1,"service":"person_detect", "question":"Is there person", "model_id": 2}] ↔ 
ChatIoT: Observation 2: UPDATE CONTEXT {new context}
LLM: Thought 3: Service request done.
Action 3: Finish[UPDATE CONTEXT {"id":1, "area": "doorway", "type": "camera", "services":{"dog_detect": {"state":{...},"person_detect":{"state":{...}}}}]

1. Context-aware Compressive Prompting

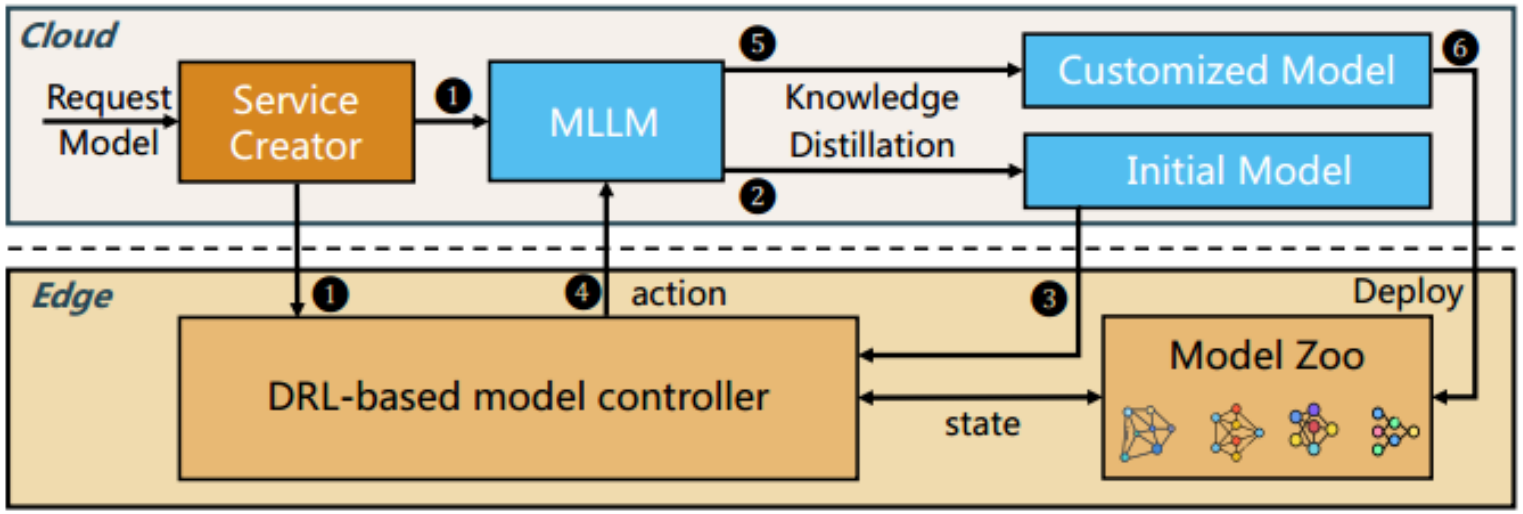
Generating TAP

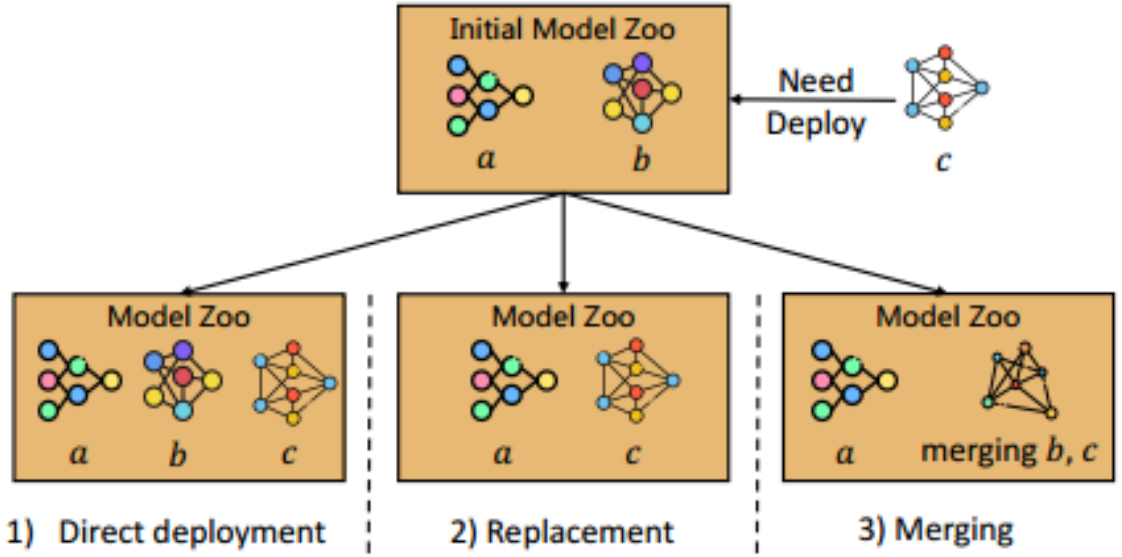
ChatIoT: Properties in TAP: ["1.person_detect.state", "2.light.on"]
TAP generation request : When a person appear at the door, turn on the light.
Context:[{"id":1, "area": "doorway", "type":"camera", "services":{"person_detect":{"state":{"description": "detect person", "model_id":2, "format": "bool", "access":["read"]}}}},
{"id":2,"area":"doorway","type":"light","services": {"light":{"on":{"description": "Switch Status", "format": "bool", "access":["read", "write", "notify"]}}}}]
LLM: Thought 1: Complete the TAP. trigger is "1.person_detect.state == true", action is "2.light.on = true"
Action 1: Finish[{"trigger": "1.person_detect.state == true", "conditions": "", "actions": "2.light.on = true"}]

Evaluating TAP

ChatIoT: TAP generation request : When a person appear at the door, turn on the light.
Context:[{"id":1,"area":"doorway","type":"camera", "services": {"person_detect": {"state":{"description": "detect person", "model_id":2, "format": "bool", "access":["read"]}}}},
{"id":2, "area":"doorway", "type": "light", "services": {"light":{"on":{"description": "Switch Status", "format": "bool", "access":["read", "write", "notify"]}}}}]
TAP: {"trigger": "1.person_detect.state == true", "conditions": "", "actions": "2.light.on = true"}
LLM: Thought 1: The trigger and actions are correctly identified based on TAP generation request and the context. The value setting are in correct format. No correction to TAP is required.
Action 1: Finish[{"trigger": "1.person_detect.state == true", "conditions": "", "actions": "2.light.on = true"}]

2. DRL-based On-demand Model Customization

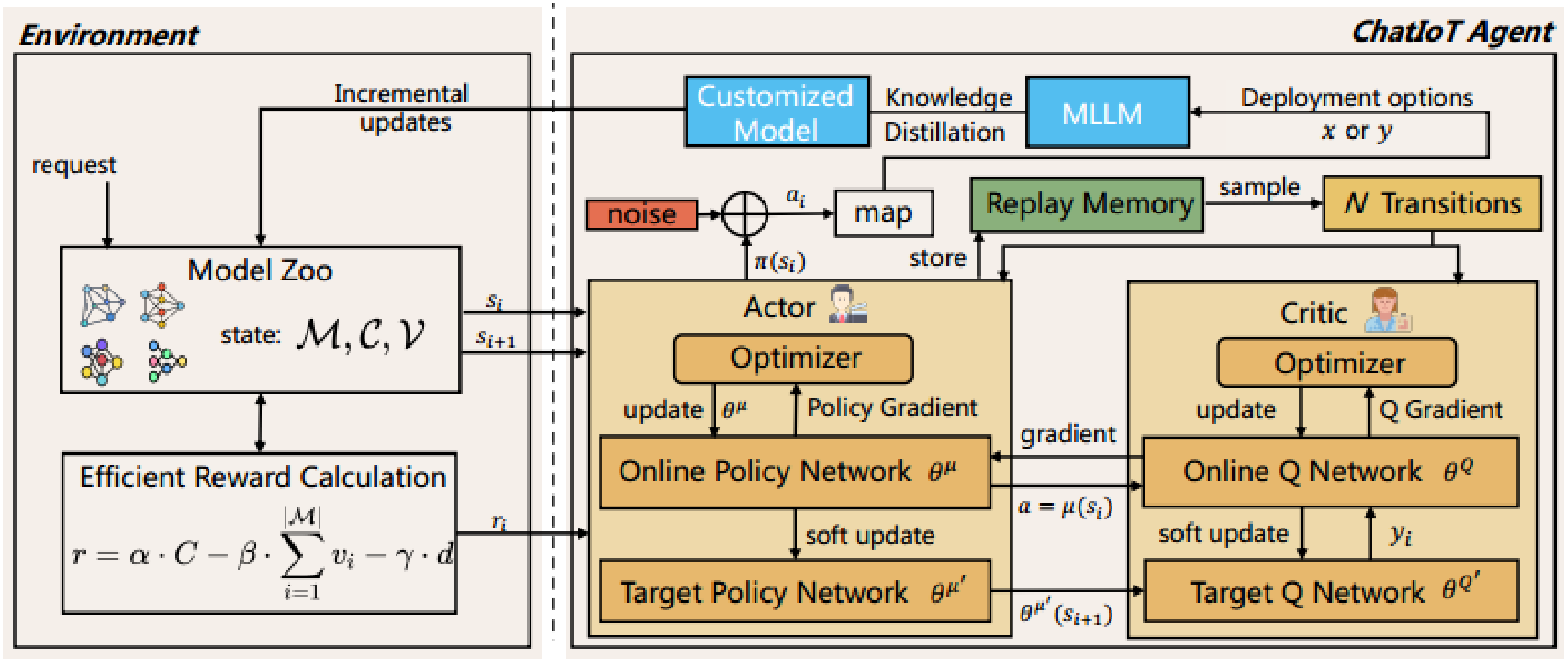




2. DRL-based On-demand Model Customization

基于ddpg的模型部署

深度确定性策略梯度（Deep Deterministic Policy Gradient， DDPG）算法



Outline

1 Background

2 Related work

3 Design

4 Experiments

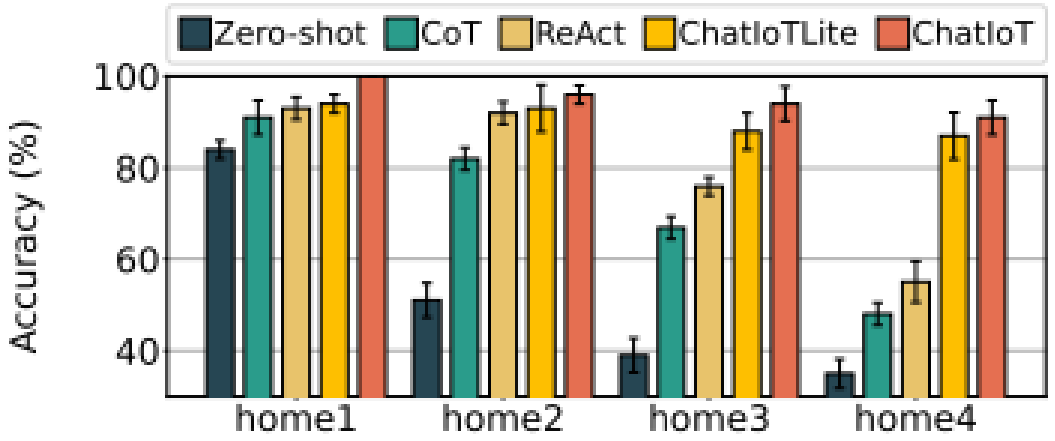
5 Conclusion

Experiments

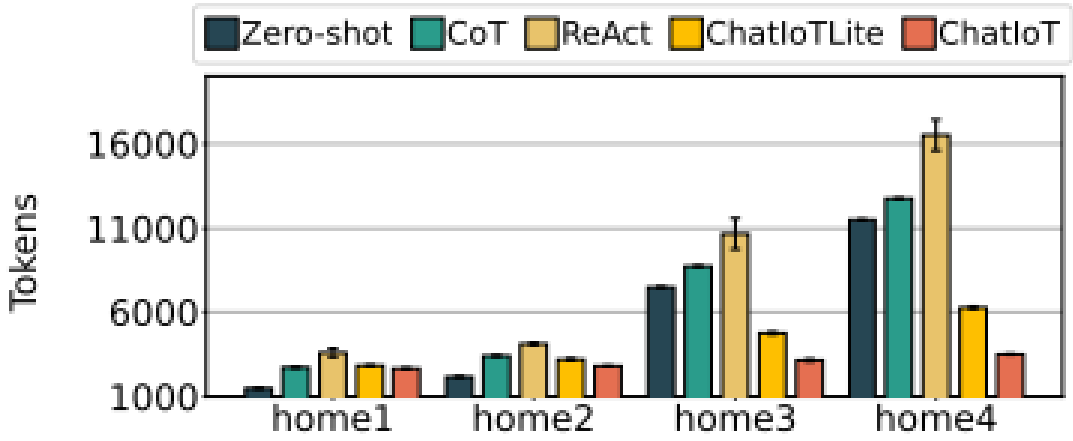
使用python-miio库生成65个设备用于评估
使用Docker部署了四个Home Assistant实例来模拟拥有不同设备数量的家庭
使用gpt-3.5 turbo模型的帮助下为每个家庭生成了一组分类请求

Home	Bathroom	Bedroom	Living room	Kitchen
Home 1	light, motion_sensor	light, air_conditioner	illumination_sensor	None
Home 2	light, motion_sensor, illumination_sensor	light, air_conditioner, illumination_sensor, temperature_humidity_sensor	light, motion_sensor, illumination_sensor	None
Home 3	light, motion_sensor, illumination_sensor	light0, air_conditioner, illumination_sensor, temperature_humidity_sensor, light1, curtain, humidifier	light0, motion_sensor, illumination_sensor, light1, light2, air_purifier, air_conditioner	light, range_hood, motion_sensor
Home 4	light, motion_sensor, illumination_sensor, water_heater	**Bedroom 1:** light0, air_conditioner, illumination_sensor, temperature_humidity_sensor, light1, curtain, humidifier, dehumidifier **Bedroom 2:** light0, air_conditioner, illumination_sensor, temperature_humidity_sensor, light1, curtain, humidifier	light0, motion_sensor, illumination_sensor, light1, light2, air_purifier, air_conditioner	light, range_hood, motion_sensor, ceiling_fan

Evaluation



(a) TAP generation accuracy



(b) Token consumption

Home	Context	Zero-shot		CoT		ReAct		ChatIoTLite		ChatIoT	
		Acc. (%)	Token	Acc. (%)	Token	Acc. (%)	Token	Acc. (%)	Token	Acc. (%)	Token
home1	1118	0	1466.14	0	2685.48	80	9634.12	100	5109.3	**100.0**	4580.68
home2	1799	0	2155.64	0	3383.54	96	12353.62	98	5737.18	**100.0**	4888.06
home3	7098	0	7452.36	0	8675.08	68	28132.60	90	9073.28	**92.0**	5368.58
home4	11144	0	11500.7	0	12718.42	64	40438.04	78	11518.16	**86.0**	6067.10

Case Study

❶ Inputs

"If the door is opened while the camera detects a person, turn on the light on the stairs."

Contexts from Home Assistant — 5736 tokens

{"id":1,"area":"living room","type":"illumination-sensor"},
{"id":2,"area":"living room","type":"television"},
{"id":3,"area":"living room","type":"ceiling light"},
{"id":4,"area":"living room","type":"air-purifier"},
{"id":5,"area":"living room","type":"RGB light"},
{"id":6,"area":"living room","type":"table lamp"},
{"id":7,"area":"stairs","type":"light"},
{"id":8,"area":"stairs","type":"motion-sensor"},
{"id":9,"area":"study room","type":"ceiling light"},
{"id":10,"area":"study room","type":"table lamp"},
{"id":11,"area":"bedroom","type":"ceiling light"},
{"id":12,"area":"bedroom","type":"air-conditioner"},
{"id":13,"area":"bedroom","type":"table lamp"},
{"id":14,"area":"bedroom","type":"temperature-humidity-sensor"},
{"id":15,"area":"entrance door","type":"magnet sensor"},
{"id":16,"area":"entrance","type":"camera"}

ChatIoT

❷ Compressor — 1.93s

Compressed Contexts — 329 tokens

{"Device list":[
{"id":7,"area":"stairs","type":"light","services":["light"]},
{"id":8,"area":"stairs","type":"motion-sensor","services":["motion-sensor"]},
{"id":15,"area":"entrance door","type":"magnet sensor","services":["magnet-sensor","battery"]},
{"id":16,"area":"entrance","type":"camera","services":[]}],
"Model zoo":[]}

❸ Preprocessor — 6.88s 808 tokens

{"type":"CreateService", "parameter":{"service": "camera_id": 15, "service_name": "person_detection", "description":"Person detection", "question":" Is there a person?"}}}

❹ Service Creator — 1.90s 545 tokens

{"type":"RequestModel", "parameter": {"question":" Is there a person?"}}

❺ Service Creator — 3.62s 668 tokens

{"type":"AddService","parameter":{"camera":{"id":16,"area":"entrance","type": "camera","services":{"person_detection": {"state": {"description": "Person detection", "model_id":1, "format":"bool", "access":["read"]}}}}}}

❻ Preprocessor — 3.94s 952 tokens

{"type":"Finish","parameter":{"property list": ["15.magnet-sensor.contact-state", "7.light.on","16.person_detection.state"]}}

❼ TAP Generator — 5.09s 858 tokens

{"type":"Finish","parameter": {"TAP":{"trigger":"15.magnet-sensor.contact-state==True", "condition": "16.person_detection.state==True", "action": "7.light.on=True"}}}

❽ TAP Evaluator — 5.48s 858 tokens

{"type":"Finish","parameter": {"TAP":{"trigger":"15.magnet-sensor.contact-state==True", "condition": "16.person_detection.state==True", "action": "7.light.on=True"}}}

❾ Deploy TAP to Home Assistant

Outputs

trigger:
- platform: state
  entity_id: binary_sensor.isa_dw2hl_925a_magnet_sensor_3
  attribute: magnet_sensor.contact_state
  to: "true"

condition:
- condition: state
  entity_id: input_boolean.person_detection
  state: "on"

action:
- service: light.turn_on

target:
- entity_id: light.yeelink_mbulb3_d9ab_light

Outline

1 Background

2 Related work

3 Design

4 Experiments

5 Conclusion

Conclusion

- **Conclusion**
  - 提出了一种新的上下文感知压缩提示方法
  - 提出了一种基于DRL的模型定制方法，在多模态LLM的帮助下，在边缘设备上生成和部署AI模型。
- **Inspiration**
  - 场景拓展：智能家居、智能教室、智能商场...
  - 问题泛化：所有人机交互设备都能用TAP吗？（自助打印机、无人驾驶车辆...）
  - 安全问题？保证对设备的操控在安全范围内进行，遇到冲突时的处理？
  - 与实际生活的轨迹trace相结合，作为备用数据参考。
  - 讲述之前研究limitation的时候，可以说”如果将大模型xxx技术和传统的xxx技术相结合的话，可以将xxx领域推向一个新的level~ “

Q&A

2024.11.10