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

Fu Li, Jiaming Huang, Yi Gao, Wei Dong
lif,huangjm,gaoy,dongw@emnets.org
College of Computer Science, Zhejiang University
Hangzhou, China

ABSTRACT

Trigger-Action Program (TAP) is a popular and significant form of Internet of Things (IoT) applications, commonly utilized in smart homes. Existing works either just perform actions based on commands or require human intervention to generate TAPs. With the emergence of Large Language Models (LLMs), it becomes possible for users to create IoT TAPs in zero-code manner using natural language. Thus, we propose ChatIoT, which employs LLMs to process natural language in chats and realizes the zero-code generation of TAPs for existing devices.

ACM Reference Format:
Fu Li, Jiaming Huang, Yi Gao, Wei Dong. 2023. ChatIoT: Zero-code Generation of Trigger-action Based IoT Programs with ChatGPT. In *7th Asia-Pacific Workshop on Networking (APNET 2023), June 29–30, 2023, Hong Kong, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3600061.3603141

1 INTRODUCTION

The Internet of Things has revolutionized the way we interact with our environment. However, creating IoT programs can be challenging and time-consuming, while TAPs (i.e. a set of rules with "if this then that" syntax) have arisen to simplify this process. IoT TAPs are commonly used in smart homes, where hands-free interaction such as voice assistant (i.e. Google Assistant, Amazon Alexa, Apple Siri, etc.) are already available to execute user commands. However, they have less ability to generate TAPs.

With the development of Natural Language Processing (NLP), LLMs become richer in reasoning. Compared with prior works of TAP generation, LLMs can empower users to generate and deploy TAPs by chatting. Likewise, RecipeGen [3] is designed for TAP generation, but the generated TAPs need further refinement and deployment manually. Besides, LLMs are applied in an attempt to push the limit of traditional IoT devices by leveraging cross-modal information.

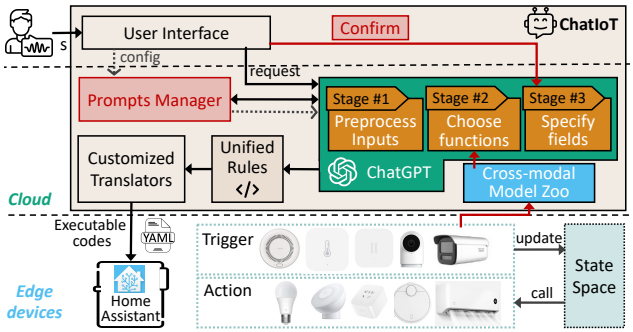

Figure 1: The overview of ChatIoT.

However, there are two challenges to achieve ChatIoT. One is how to make efficient use of LLMs. It is critical to improve the quality of TAP generation while reducing the tokens required for prompt. The other one is how to make efficient use of cross-modal information.

We propose **ChatIoT** to provide zero-code rule generation while addressing the above challenges. The Prompts Manager is designed to guide accurate rule generation based on LLMs. With additional confirmation, rules will be refined to be more accurate and safe to execute. To protect user's privacy while avoiding overhead on edge devices, we develop Cross-modal Model Zoo in the cloud to distribute functionally customized models of appropriate sizes to users.

2 DESIGN

Fig. 1 shows the overview of ChatIoT. It is designed to chat with user to generate unified rules by utilizing ChatGPT. Furthermore, we realize a translator to deploy unified rules to Home Assistant (HA). We will then detail two critical parts of ChatIoT.

2.1 Prompts Manager

Prompts Manager is proposed to enable ChatGPT to better assist users in generating unified rules, with the goal of reducing input tokens and increasing the efficiency. As shown

APNET 2023, June 29–30, 2023, Hong Kong, China

ACM ISBN 979-8-4007-0782-7/23/06.
https://doi.org/10.1145/3600061.3603141

in Fig. 1, the whole process of rule generation can be divided into three stages:

- **Preprocess Inputs:** Determine if user input contains an explicit rule and extract essential details. Prompts Manager will establish knowledge of the user's IoT system about functions and devices during initialization. Based on the extracted details, ChatGPT will generate specific rules within a definitive scope.
- **Choose Functions:** Choose appropriate functions to create rules. If the target rule extends beyond the definitive scope, ChatGPT will try to use video understanding to satisfy the rule requirement by finding an appropriate model in the Cross-modal Model Zoo.
- **Specify Fields:** Specify fields of the rule. When user input is insufficient to generate all the required fields, the user will be prompted for further specification.

Benefiting from such a design, ChatIoT can generate IoT programs tailored to specific user scenarios.

2.2 Cross-modal Model Zoo

In this section, we will describe what exactly a Cross-modal Model means and how Model Zoo works with edge devices.

Cross-modal Model. Cross-modal Models refer to neural networks that can process and integrate multimodal sensor data. BLIP [1] is such a pre-training model that has the ability to connect visual and linguistic information. When commands exceed system limits, we propose joint inference utilizing cameras to fulfill user's requirements. In addition, the inference result comes with the confidence that allows a smarter judgment of the user's expectations.

Customized Coordination. To protect user privacy, models are deployed close to the user side, but the limited capability of the user-side devices restricts high-performance inference. Accordingly, we propose lightweight model customization based on cloud-side collaboration. We use knowledge distillation to distill customized student models based on user requirements in the cloud, and then dynamically distribute the model to the user side and perform model substitution on user-side devices.

3 EVALUATION

Datasets. To verify ChatIoT's capability for generating rules, we unified the rule format to match the IFTTT platform and evaluate its performance on Gold15 [2] dataset, which contains 305 TAPs on the function-level granularity. We extract 172 functions to initialize Prompts Manager. To align with the targets, we pause the process at the second stage and check the performance of generating rules.

Setup. We conducted case studies considering a hypothetical home with a living room, bedroom, kitchen, and bathroom, equipped with 9 Xiaomi smart devices such as lights, light sensors, cameras, etc. We integrated all devices into HA through MIoT, and cameras are connected through an RTMP server. Further, we implement a translator to convert unified rules into HA automations. When video understanding is required, it creates a corresponding entity and python scripts containing the customized model to update the entity's status.

3.1 Performance

The results show that ChatIoT achieves **94.1-98.5%** accuracy on channel and function prediction with around **350** tokens about functions. Compared to without Preprocessing Inputs, the accuracy is improved by 3-24% and the number of tokens is reduced by 79.41%. With additional confirmation from users, it can also provide extra field parameters, which is actually beyond the capability of RecipeGen.

3.2 Case study

We conducted two cases to show the capacity of ChatIoT. Case 1 demonstrates simple and explicit rule generation, while Case 2 shows complex rule generation that involves cross-modal information.

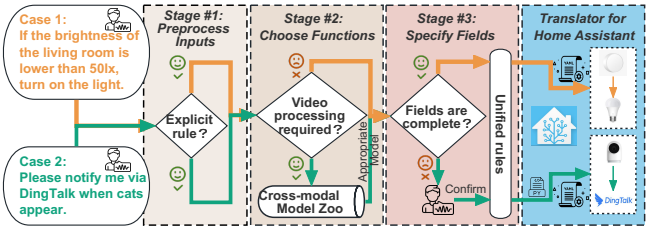

Figure 2: Case study.

4 CONCLUSION

With ChatIoT, users can easily generate IoT TAPs tailored to specific scenarios and leverage the Cross-modal Model Zoo to enhance the capability of their IoT system.

REFERENCES

[1] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*. PMLR, 12888–12900.

[2] Chris Quirk, Raymond Mooney, and Michel Galley. 2015. Language to code: Learning semantic parsers for if-this-then-that recipes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 878–888.

[3] Imam Nur Bani Yusuf, Lingxiao Jiang, and David Lo. 2022. Accurate generation of trigger-action programs with domain-adapted sequence-to-sequence learning. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*. 99–110.