Say Your Reason: Extract Contextual Rules In Situ for Context-aware Service Recommendation

YUXUAN LI, Department of Computer science and Technology, Tsinghua University, China
JIAHUI LI, Department of Computer science and Technology, Tsinghua University, China
LIHANG PAN, Department of Computer science and Technology, Tsinghua University, China
CHUN YU*, Department of Computer science and Technology, Tsinghua University, China
YUANCHUN SHI, Department of Computer science and Technology, Tsinghua University, China

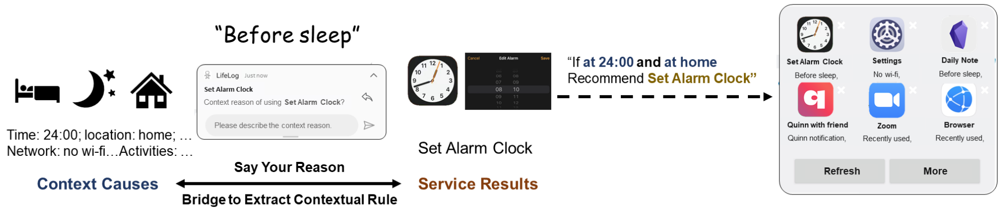


Fig. 1. When user use services on their smartphones, SayRea proactively requests user to say the reason in situ. SayRea use the reason as bridge between context causes and service results to extract contextual rule for context-basd service recommendation. Next time if the context meets rules, our system provide quick access for the services on home screen and lock screen.

This paper introduces SayRea, an interactive system that facilitates the extraction of contextual rules for personalized context-aware service recommendations in mobile scenarios. The system monitors a user's execution of registered services on their smartphones (via accessibility service) and proactively requests a single-sentence reason from the user. By utilizing a Large Language Model (LLM), SayRea parses the reason and predicts contextual relationships between the observed service and potential contexts (such as setting the alarm clock deep in the evening). In this way, SayRea can significantly reduce the cognitive load on users in anticipating future needs and selecting contextual attributes. A 10-day field study involving 20 participants showed that SayRea accumulated an average of 62.4 rules per user and successfully recommended 45% of service usage. The participants provided positive feedback on the system's usability, interpretability, and controllability. The findings highlight SayRea's effectiveness in personalized service recommendations and its potential to enhance user experience in mobile scenarios.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; **Interactive systems and tools**.

*Indicates the corresponding author.

Additional Key Words and Phrases: large language model, recommender system

ACM Reference Format:

Yuxuan Li, JiaHui Li, Lihang Pan, Chun Yu, and Yuanchun Shi. 2024. Say Your Reason: Extract Contextual Rules In Situ for Context-aware Service Recommendation. In *CHI '18: ACM CHI Conference on Human Factors in Computing Systems, May 11–16, 2024, Hawaii, USA.* ACM, New York, NY, USA, 19 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

With the increasing number of mobile applications and in-app services, accessing services occupies more and more users' attention, sometimes even surpassing the services themselves[21]. Context-aware recommendation enables efficient and comfortable user access to smartphone services. Contexts (e.g., deep in the evening) and services (e.g., setting the alarm clock) have strong relationships. Existing systems often utilize contextual rules (with the context as the cause and the service as the effect) that can directly describe these relationships (such as "if it's late at night and about to sleep, then recommend the alarm clock service"). Introducing end-user to edit contextual rules meets users' requirements for quick service accessing and offers interpretability and controllability. However, existing user rule editing systems[12, 25] have high entry barriers. Users have a heavy operation load on selecting context attributes and a significant cognitive load on anticipating future needs to build rules deliberately in advance.

In order to solve the high entry barrier problem, this paper proposes SayRea (Say your Reason), an interactive system that facilitates the extraction of contextual rules for personalized context-aware service recommendations in mobile scenarios using a Large Language Model(LLM). The rule extraction method in SayRea has two characteristics: simple interaction mechanism and appropriate interaction timing. (1) SayRea's interaction mechanism requests users of single-sentence natural language reasons for services instead of complete and accurate information about the contexts and services. The users do not need to manually select any context attributes, SayRea does the job for them. (2) SayRea extracts rules in situ by interacting with users for automatically selecting essential contexts when the usage of services occurs. In situ, users do not need to deliberately anticipate future needs, and the system can automatically perceive necessary surrounding contexts.

SayRea utilizes a pre-trained large language model(LLM)[5] to fully leverage the semantic information in users' reasons, in order to automatically identify essential context attributes and bridge the contexts and services when building contextual rules. In most cases, SayRea identifies essential context attributes using only two components: (1) semantic translations of contexts and services in the form of natural language representations; (2) users' reasons. It can then extract contextual rules based on the identified context attributes. In some cases, as contextual rules accumulate over time, SayRea is able to identify the existing contextual rules that match the current context attributes. This allows users to engage in interactive rule extraction without even being required to provide a natural language reason, thereby further reducing the burden on users.

SayRea also enables users to delete extracted contextual rules when false or expired rules accumulate. SayRea uses a tree structure to manage rules and ranks recommendations when multiple rules satisfy the contexts after the accumulation.

We implemented SayRea on the Android platform. SayRea collects and labels commonly used services in different applications and monitors the services on users' smartphones with the Accessibility Service. It applies context awareness[24] to collect the contexts and semantically translate them using feature engineering. We also designed interactive interfaces for rule extraction and service recommendation.

A 10-day field study with 20 participants was conducted to evaluate the SayRea system. SayRea extracted an average of 62.4 rules from each user, indicating the high usability of the extraction method. The service coverage(a service being used is covered if it is also being recommended) increased over time and reached a maximum of 45%, demonstrating the system's recommendation capability. Users' subjective feedback showed a high acceptance of SayRea, and users thought that the system had good usability, interpretability, and controllability.

In summary, the main contributions of this paper are:

- The proposal of an in situ interactive method for extracting contextual rules. This method enables low-burden user interaction at appropriate times to extract contextual rules between contexts and services using a Large Language Model.
- The implementation of a context-aware service recommender system with contextual rule extraction. The system is interpretable, controllable, and able to personalize.

In the following sections, we first summarize related work on context-aware service recommender systems, rule editing. In Section 3, we introduce the method for extracting contextual rules. Then we elaborate on the context-aware service recommendation in Section 4. Section 5 is the implementation of SayRea's system modules. And section 6 presents the field study of SayRea and its results. Finally, we provide discussions, limitations, and a conclusion for this work.

2 RELATED WORK

This section will summarize the previous work on context-aware service recommender systems and rule editing. We mainly introduce the contributions of the SayRea system presented in this paper compared to these related works.

2.1 Context-aware Service Recommender System

Context-aware service recommendations[10, 33] can optimize users' smartphone service usage processes. Currently, commercial application recommendations on smartphones, such as recommendations in Siri Suggestion[7], and some research work in the field, like DeepApp[31], CMARA[40], generally, adopt end-to-end deep learning models for recommendations. These models are intelligent but have issues in interpretability[22, 39] and controllability[13, 32], leading to lower usage rates. Although some context-aware recommender systems attempt to provide explanations for recommender models[29], such as Amit[20] and Zhang[38]'s work, and CAESAR[18], they still require a large amount of data and time to accumulate data.

The LLA application[14] introduces context management and service recommendations based on specific algorithms with relatively fewer contexts. These systems show a certain degree of interpretability and can adapt to user usage. Compared to SayRea, however, these systems are more complex to manage and still lack controllability. Context-aware service recommendations can also be sourced from association rule mining[37] under the guidance of machine learning methods[26], such as ABC-ruleminer[28]. Recommender systems based on such association rules will have better interpretability, but still have data dependency, potential redundancy of rules[36], and lack of controllability.

SayRea interacts with users to extract causal rules for context-based service recommendations, requiring minimal data support. It has advantages in a mobile intelligent system environment where users value data privacy while achieving low implementation costs, interpretability, and controllability.

2.2 Rule Editing

The ability for end-users to edit rules is the foundation of personalization and controllability in SayRea. There are related work in the field of human-computer interaction. Rule editing is common in task automation systems[9], such as IFTTT (If This, Then That)[23], which allows defining actions by connecting applications to complete tasks and actions automatically. According to IFTTT's website, more than 600 applications and devices collaborate with IFTTT, such as Twitter, Telegram, Google Drive, and Alexa. Similarly, in the Shortcuts on iOS platforms[3], specific conditions can be defined under which a shortcut should automatically run. Users need to choose from a list of many trigger conditions.

Rule editing through a "form-filling" method is considered very tedious and boring[15]. Some research has proposed systems like SECE (Sense Everything, Control Everything)[4] and DOTT (Do This on That)[8], which allow users to edit rules using natural language, and "one app to rule it all"[15] enables users to edit rules through methods such as drag-and-drop. These works focus on optimizing the methods for user rule editing but still rely on a priori rule editing and require formalized expressions, which can lead to cognitive and interaction burdens. The timing of in situ interactions has been discussed in feedback procedures for recommender systems, like AppEcho[30]. SayRea uses in situ interaction timing to allow users to provide simple reasons, resulting in very low burden rule editing in terms of interaction and cognition.

2.3 In situ interaction

An important challenge in user intervention in context-aware recommendation systems (e.g., editing rules) lies in the timing of interactions. Context is constantly changing, leading to a significant gap between the context in which users need a service and the context in which they interact with the system [14]. This introduces an additional mental burden: users must recall or envision a non-existent scenario. This gap also makes it difficult for users to get immediate result on their interventions, which not only reduces their confidence but also diminishes their willingness to interact.

Sayrea applies in situ interaction to address the problem mentioned above. In situ interaction meaning that the interaction takes place where and when the target context exists. It has been widely used in visualization systems [16]. For examlpe, reaseachers tried to augment code with in situ visualizations to help program understanding [11]. However, this method has not been applied to recommendation systems yet.

3 CONTEXTUAL RULE EXTRACTION

This section mainly introduces the contextual rules for context-based service recommendation in SayRea and its in situ interactive extraction method.

3.1 Contextual Rules of Contexts and Services

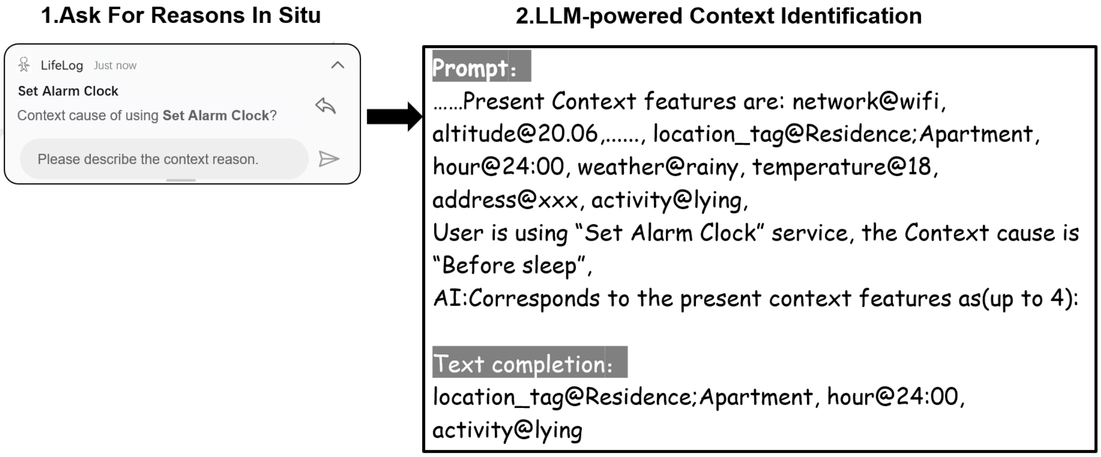
Contextual information is often multi-dimensional, involving aspects like time, location, and activity status. Each dimension contains various features. For instance, time could have features such as the day of the week and the specific hour. At any given moment, each feature has a unique value, such as [Monday, 7 a.m.]. We call these pairs of features and values "context attributes." These attributes provide LLMs with the necessary information to identify the most relevant attributes for making service recommendations. The results in the contextual rules are services, which include apps and in-app services on mobile phones.

Context-aware service recommendations are based on these contextual rules, expressed as "if a certain context occurs, recommend a certain service." The context cause is a combination of multiple context attributes, describing complex contexts. The formalized expression is as follows:

$$if\ A_1\ and\ A_2\ and\ A_3\ ...\ then\ recommend\ S_A.$$

For example, in contextual rule "if the user is in the dormitory at 24:00 at night and lying down, recommend the alarm clock service", A_1 represents being in the dormitory, A_2 represents that it is 24:00 at night, A_3 represents lying down, and S_A represents the alarm clock.

3.2 Contextual Rule Extraction



1.Ask For Reasons In Situ

LifeLog Just now

Set Alarm Clock

Context cause of using **Set Alarm Clock**?

Please describe the context reason.

2.LLM-powered Context Identification

Prompt:

……Present Context features are: network@wifi, altitude@20.06,……, location_tag@Residence;Apartment, hour@24:00, weather@rainy, temperature@18, address@xxx, activity@lying,
User is using "Set Alarm Clock" service, the Context cause is "Before sleep",
AI:Corresponds to the present context features as(up to 4):

Text completion:

location_tag@Residence;Apartment, hour@24:00, activity@lying

Fig. 2. An example of contextual rule extraction through two steps.

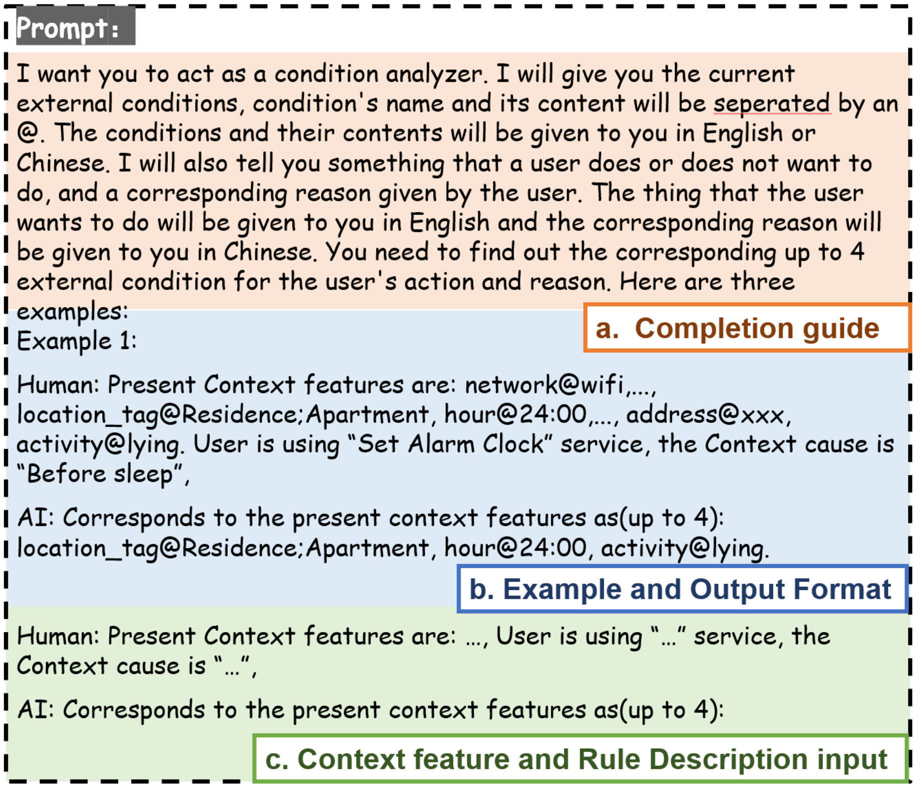
SayRea uses proactive interaction requests in situ to allow users to establish contextual rules by saying the reasons in natural language, reducing the burden of users and lowering the entry barrier for user rule editing. The overall process is shown in the figure2. SayRea first requests interaction with users for them to say natural language reasons. The system then uses an LLM to predict identify essential context attributes. The cause (a set of identified context attributes) and the result (the service) then form a contextual rule,

3.2.1 Ask For Reasons In Situ. When SayRea recognizes(further elaborate in 5.2) that the user is using a service, it initiates an interactive request to ask the user for a reason.

The main content of the interactive request is to ask the user to provide a natural language reason for using the service. A simple reason is all that the user need, without attention for expressing the context details or using formatted sentences.

This in situ(service recognized) timing can reduce the burden of interaction: 1) The operational burden is insignificant: the system perceives current contexts and services, so the user only needs to say the reason. 2) The cognitive burden is negligible: the user is in the current context and using a needed service, not requiring anticipation of future needs.

3.2.2 LLM-powered Context Attributes Identification. LLMs are able to map the open-ended natural language reason to the corresponding context attributes using general knowledge (e.g., the natural language reason "very hot" corresponds to the context attribute "temperature: 28 degrees Celsius"). Identifying essential context attributes can be transformed into a text completion task. The input is the user's natural language context reason, the system's perception of the current service, and the semantic representations of all context attributes. The output is a list of predicted context attributes, and their combination represents the context cause in the contextual rule. To better accomodate users' natural language reasons, we choose a dialogue-form prompt[6] design based on small-scale trial with LLMs. Figure 3 shows the prompt and its purpose decomposition, including a) guidance for text completion; b) an example of the completion format, which outputs a list of context attributes; c) the current natural language reason of the context reasons and input of the context attributes.



Prompt:

I want you to act as a condition analyzer. I will give you the current external conditions, condition's name and its content will be seperated by an @. The conditions and their contents will be given to you in English or Chinese. I will also tell you something that a user does or does not want to do, and a corresponding reason given by the user. The thing that the user wants to do will be given to you in English and the corresponding reason will be given to you in Chinese. You need to find out the corresponding up to 4 external condition for the user's action and reason. Here are three examples:

Example 1:

Human: Present Context features are: network@wifi,…, location_tag@Residence;Apartment, hour@24:00,…, address@xxx, activity@lying. User is using "Set Alarm Clock" service, the Context cause is "Before sleep",

AI: Corresponds to the present context features as(up to 4): location_tag@Residence;Apartment, hour@24:00, activity@lying.

Human: Present Context features are: …, User is using "…" service, the Context cause is "…",

AI: Corresponds to the present context features as(up to 4):

a. Completion guide
b. Example and Output Format
c. Context feature and Rule Description input

Fig. 3. Prompt design for LLM-powered context cause extraction with its purpose decomposition.

Since the task has been transformed into a text completion task, any LLM with text completion capability can be used, not limited to GPT-3.5[5] (which we used). Accurate identification of the essential context attributes is important to the correct accumulation of contextual rules and performance of the recommender system. We have validated the LLM's ability in this task through further evaluation (see Section 7).

3.3 Accumulation and Optimization

After a contextual rule is extracted, the user's single-sentence reason is accumulated along with the contextual rule in the system. During another contextual rule extraction process, if SayRea detects that the current context attributes meets an accumulated contextual rule's cause, it directly asks whether a contextual rule can be extracted based on the single-sentence reason corresponding to the context cause, without requesting the user to give the reason. Such as the "*Home time* and *Reading finished*" in Figure7(a), shown as predicted reasons on the user interface. The contextual rule extraction process is optimized by reducing the interaction steps with accumulated context causes and reducing interaction burdens.

4 SERVICE RECOMMENDATION

SayRea recommends services by generating a list of services and recommendation reasons based on the current context, accepts quickly accessing services and feedbacks for incorrect recommendations from users.

SayRea has the following characteristics for service recommendation: **Low cost**: SayRea mainly uses contextual rule-based recommendation, saving time and professional personnel for data collection and model training. **Interpretability**: SayRea semantically analyzes all the contexts and services involved in the process, and the recommendation based on contextual rules has explicitly recommendation reasons. **Personalization and controllability**: the contextual rules in SayRea origin from user editing, and users can manually modify the rules based on their specific needs.

4.1 Cold-start Recommendation

Cold-start recommendations consist of a list of recommended services in the order of the user's most recent use. This is similar to the existing service recommender systems[7] on smartphones.

4.2 Contextual Rule-based Recommendation

SayRea acquires all the current context attributes and recommends the corresponding result services when the contextual rules are met.

4.3 Recommendation Feedback

The user can give feedback for incorrect recommendations on the recommended results of the SayRea system. SayRea system initiates the contextual rule extraction again when recommendations are rejected. This time the contextual rule differs from the positive contextual rules by being negative:

$$if\ A_1\ and\ A_2\ and\ A_3\ ...\ then\ not\ recommend\ S_A.$$

4.4 Accumulation and Optimization

SayRea accumulates the contextual rules in a context rule tree, which is used for optimizing the service recommendation process.

4.4.1 Rule Management: Context Rule Tree. Wu's item recommendation rule tree [34] inspired our context rule tree. As shown in Figure 4, the nodes of a context rule tree represent context attributes, and the path from the root node to a node is a combination of context attributes. If there is a service at a node, it means that this node corresponds to a contextual rule: if the combination of context attributes of the node is satisfied, the corresponding service is

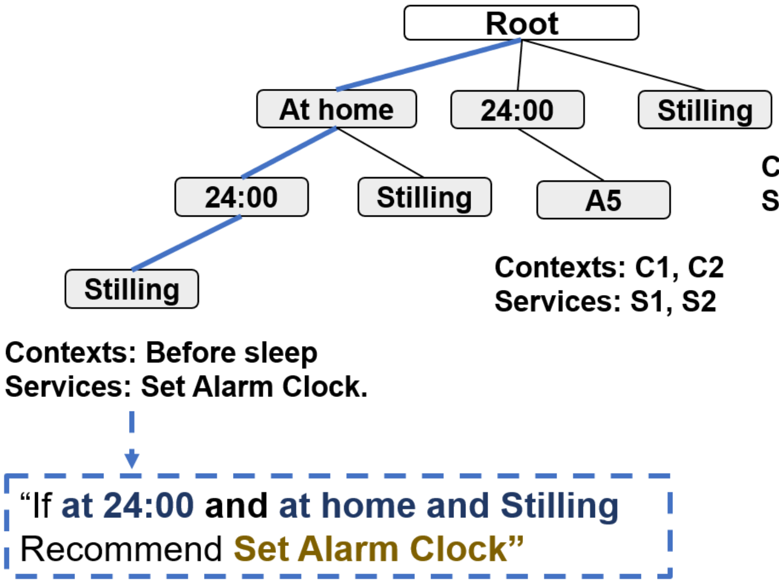


Fig. 4. A context rule tree Example. Nodes of the tree represent context attributes(At home, 24:00, Stilling, A5), and some nodes also store the context reasons(Before sleep, C1, C2, C4) and service results(Set Alarm Clock, S1, S2) for contextual rules.

recommended. If the node has a context reason, this context reason is a natural language representation of the node's context attributes combination. Context rule tree can store accumulated different contextual rules for a same service or a same contextual rule for different services. To prevent redundancy, if two contextual rules share the same combination of context attributes, only one branch will be stored in the tree because we enforce context attributes of the parent node superior than those of the child node. Using such a tree structure can cope with the situation of sparse context attributes but complex context feature combinations, save storage space, and perform queries quickly.

4.4.2 Context Rule Tree-Based Recommendation. The context rule tree can find all paths that satisfied the context attributes of each node by the current context, and so acquire all services that match the contextual rules in the current context. The number of occurrences of the services and the depth of the tree correspond to the degree of matching with the current context and the complexity of the context cause, respectively. When recommending services, the services can be ranked according to the number of occurrences and the depth of the tree.

For the negative contextual rules extracted based on users' feedback, SayRea builds a negative context rule tree. For the current context, services that are acquired from the negative context tree are removed from the service recommendation list.

5 IMPLEMENTATION

SayRea's basic framework is shown in Figure 5. The system consists of two important processes: the contextual rule extraction process and the service recommendation process. The core contextual rule extraction module and service

Table 1. Context in SayRea system

Type	Awareness	Features
Time	Timestamp	day of the week, time period, o'clock
Network	Network manager	network type, SSID, speed
Bluetooth	Bluetooth sensor	Bluetooth state, connected counts, connected devices
Weather	Third party API	weather type, temperature
Location	GPS, Third party API	altitude, address, location tag, position of interest(POIs)
Activities	Activities recognization	on foot/running/walking/in vehicle...
Service	Mobile phone logs	recently used app, unlock screen time
Notification	Notification manager	recent notification, current cnotifications

recommendation module, as well as their corresponding accumulation modules, have been described in the previous section. SayRea also implements a semanticization module for contexts and services, and a service recognition module to support those two processes. This section describes the design and implementation of these two modules and the system's user interface.

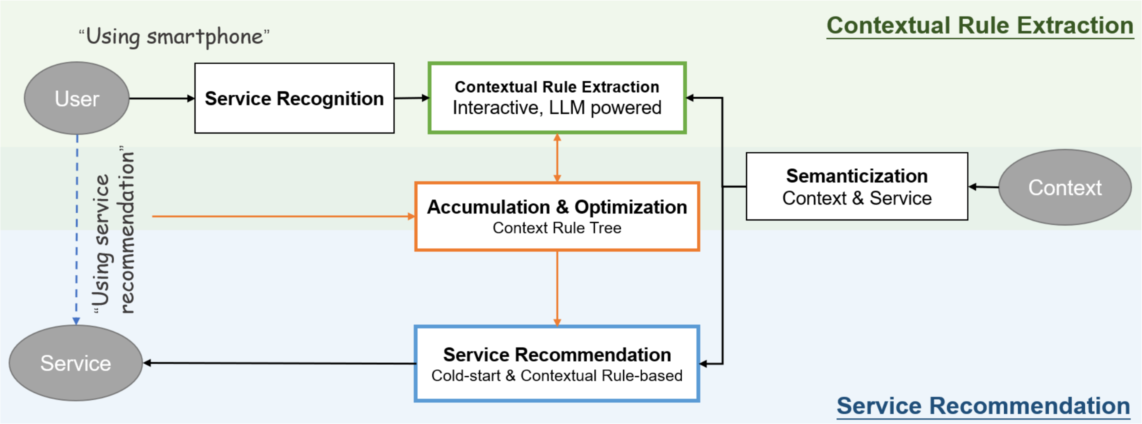


Fig. 5. SayRea's system flow. SayRea extracts contextual rules while users using smartphones, and supports using service recommendations.

5.1 Semanticization Module for Service and Context

Semanticization modules are designed for interpretable interaction with users, and for better inputs of the LLM to take advantage of general knowledge.

5.1.1 Service Semantic. SayRea supports 41 selectively labeled in-app services from 14 different types of applications, and an unlimited number of "launch an application" type services (i.e., opening applications installed on the user's phone, e.g., "open Clock"). We provide a semantic label for each service (e.g., "set alarm clock"), and in particular, the semantic of an "open an app" service is defined as the name of the application.

5.1.2 Context Semantic. SayRea uses multiple contexts for service recommendation, as shown in Table 1. Our system obtained the context data through context awareness and multiple context dimensions through feature engineering. The whole context at any moment has a value taken in each context dimension, which can be regarded as the set of context attributes. Templated translations("temperature with a value of x can be translated to *x degrees*") of the context attributes yields the context semantics (such as Monday, 10 degrees, etc.). Any new context dimensions can be easily translated into contextual semantics for use in SayRea, showing the system's scalability.

5.2 Service Recognition Module

This module ensures an in situ interaction timing for the contextual rule extraction method when users are using services. SayRea's service recognition process for labeled in-app services is shown in Figure 6. First, the system labeled a predefined set of services with the texts from the service operation sequences and the keywords (static text or content descriptions[2] on the page) in the page sequence.

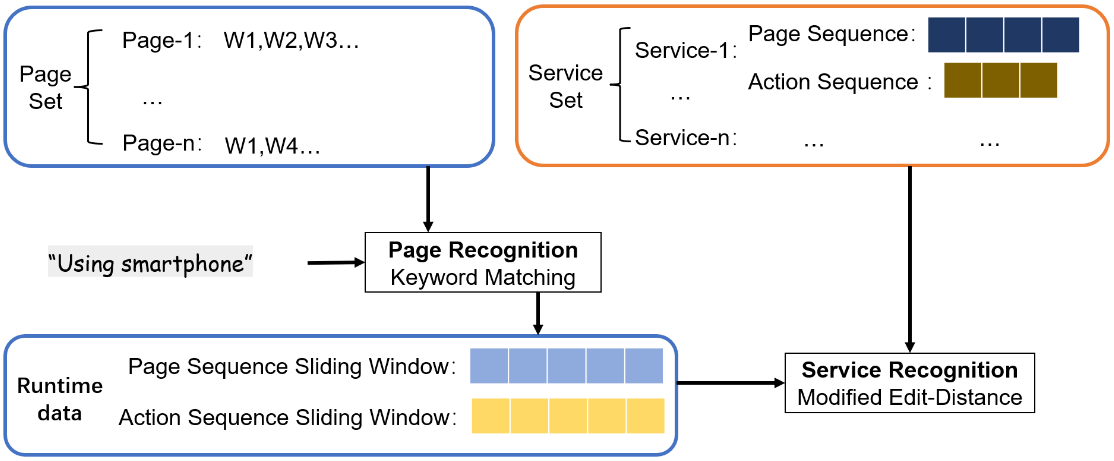


Fig. 6. Service recognition produce.

We use the Accessibility Service[1] to obtain the pages and actions running on the phone. Page recognition is performed by keyword matching (if the run-time page's 80% keywords match the labeled page's keywords), and sliding windows for both pages and actions are maintained separately. When a page is recognized SayRea uses a modified shortest edit distance algorithm(Levenshtein distance[17], the edit distance for deleting a sliding window sequence is set to 0 to avoid a large sliding window sequence causing a large edit distance) to calculate the distances between reversed sliding windows and every labeled sequence of page and action separately. A labeled service is recognized run-time if the edit distance is less than 1.5.

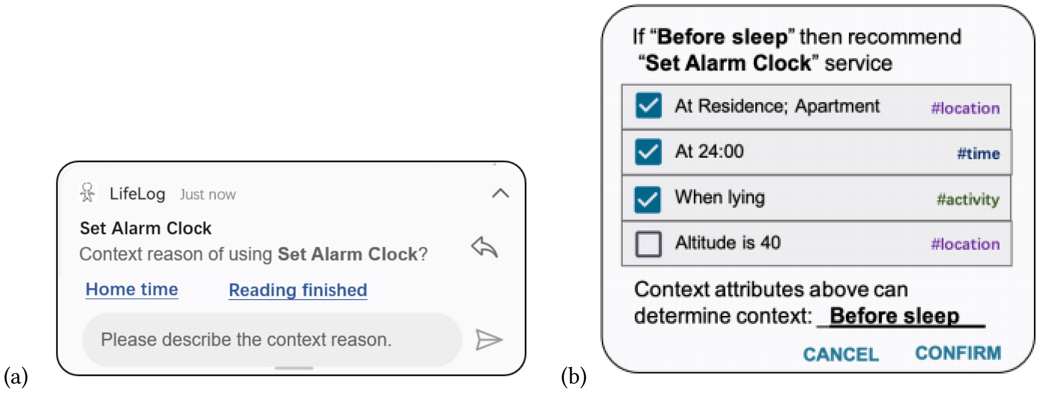
And the package-changing events can be monitored by Accessibility Event[1] to recognize the opening of any applications.

5.3 User Interface

5.3.1 *Rule Extraction User Interface.*

Notification to ask for interaction. At the timing of in situ requests after service recognition, we use notification reminders to ask for reasons and show predicted reasons, as in Figure 7(a). There is not always a contextual rule for the service, and users are not always wishing to extract. Notification reminders do not interrupt user's current interaction, with which the user can judge whether contextual rules need to be extracted.

List display to select context attributes. Notice, this part of manual selection is solely designed to validate LLM's ability of correctly identifying essential context attributes. The result of the selection is only used for validation of the alignment between LLM's identification of essential attributes and users' intentions. Users use the list selection method to choose the context attributes corresponding to the contextual rules. We attach colored tags to context attributes for users to find and identify them easily.



Set Alarm Clock

Context reason of using **Set Alarm Clock**?

Home time Reading finished

Please describe the context reason.

If "**Before sleep**" then recommend "**Set Alarm Clock**" service

☑ At Residence; Apartment #location

☑ At 24:00 #time

☑ When lying #activity

□ Altitude is 40 #location

Context attributes above can determine context: Before sleep

Fig. 7. (a) Notification to ask for reasons and to show the predicted context reasons(*Home time*, *Reading finished*). (b) List for users to select context attributes.

5.3.2 *Service Recommendation User Interface.*

Home screen widget. SayRea implements a home screen widget that displays the recommended results, as shown in Figure 8(a). The icon, label, and recommendation reason are displayed for each service result, and clicking on them triggers the corresponding service. It also provides a "×" delete button in the upper-right corner of the icon to remove the corresponding service recommendation result from the list and initiate an error feedback interaction. The home screen widget is similar to the current application recommendation on smartphones, imposing low learning costs for users. In addition, recommendation reasons make SayRea more interpretable.

lock screen notifications. SayRea additionally implements a grid list notification, as shown in Figure 8(b). For each service result, only the icon and label of the service are displayed and clickable to trigger the service. The widget still requires unlocking the phone, and the phone may not show the screen where the widget is located, lock screen notifications are relatively faster.

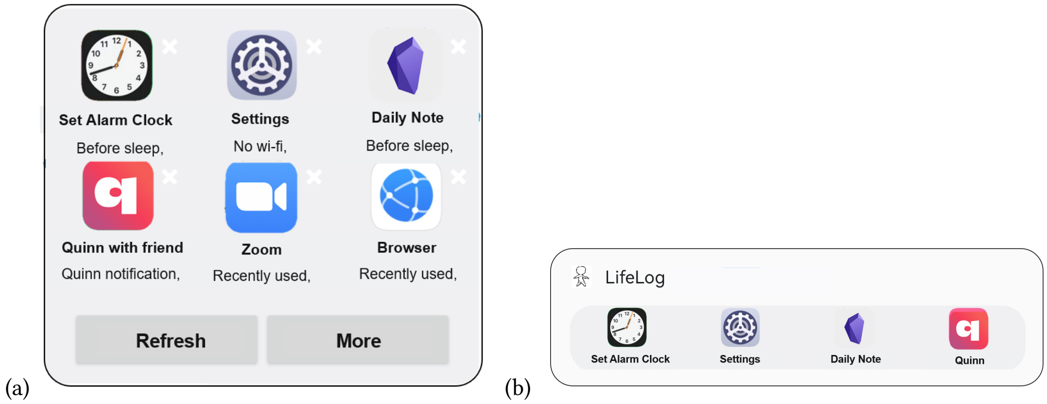


Fig. 8. (a) Home screen widget, showing the recommended services and reasons. (b) Lock screen notification, showing the recommended services.

6 USER STUDY

We conducted a 10-day field study with a total of 20 participants. We first asked each participant to sign an informed consent before the experiment. Notice that the university ethics review board approves human-subjects research and they approved this project. We then introduced SayRea to the participants and installed the application on their smartphones. We demonstrated how to say reasons for using particular services. We also ask the participants to choose corresponding context attributes for validation use. We required participants to place the widget that shows recommended services in an obvious place on the home screen. We also asked them to give necessary permissions to enable information gathering and internet connection of the application.

Participants are asked to use SayRea naturally on their smartphones, without being required of any deliberate use. Throughout the 10-day field study, we logged users' behaviours and used objective metrics and subjective metrics to evaluate the system.

6.1 Result

6.1.1 Accuracy of LLM's predictions. We use GPT-3.5-turbo[5] to predict context attributes based on semanticized contexts, services and users' reasons. A list of LLM's predicted context attributes is considered **accurate** when more than 75% of users' choices of context attributes overlap with LLM's identification. In our field study, 85% of LLM's predictions are accurate. This considerably high accuracy shows that LLM has a trustworthy ability to identify context attributes, subsequently to accumulate correct contextual rules.

6.1.2 Accumulation of contextual rules. We calculated the number of contextual rules accumulated by each user. 50% participants had accumulated more than 60 rules by the end of the experiment. 95% participants accumulated more than 20 rules. One user accumulated 163 rules at the end of the experiment. Figure 9 shows the accumulation of contextual rules of each user.

Results show that SayRea can accumulate a considerable number of contextual rules without participants' deliberate use of the system. Although the number of contextual rules accumulated may vary due to different smartphone using habits, it can be concluded that SayRea has shown the usability of the extraction method.

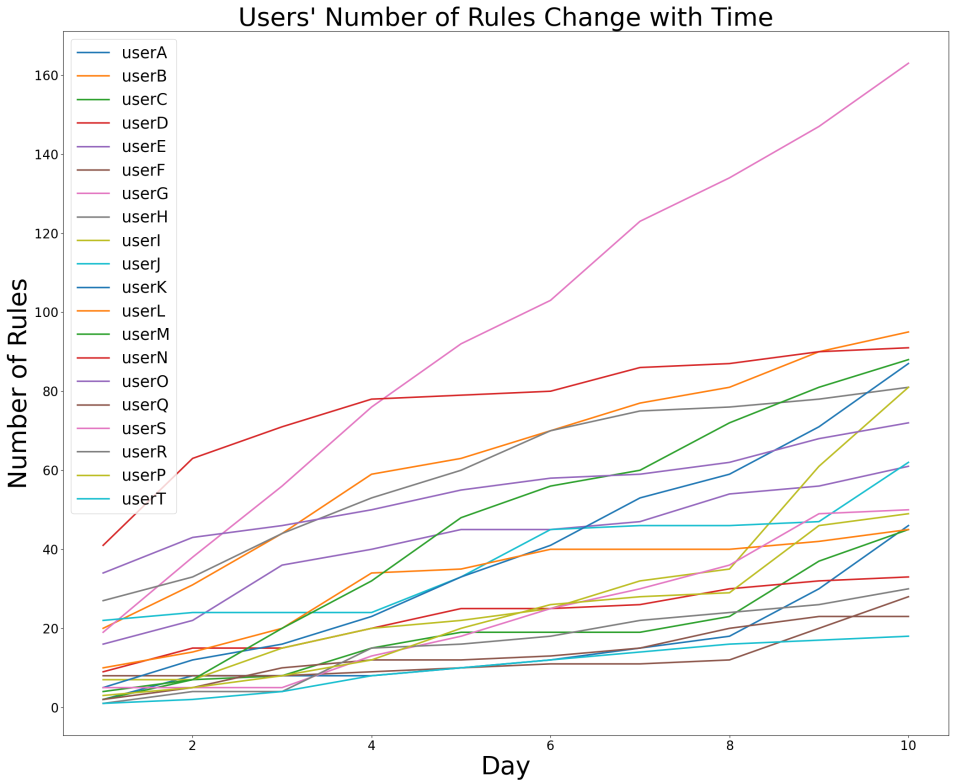


Fig. 9. Accumulation of contextual rules of each user.

6.1.3 Time consumption of rule extraction for users. We calculated the time consumption of rule extraction(from "start reason input" to "finish context attribute selection") for each user. During this time, users can naturally continue their usage of the services when information(contexts, services and reasons) are being processed by the system. Results show that it takes a user 16.12 seconds to go through this process on average. 50% participants spend less than 15 seconds to finish the extraction process. And the user who spends the least time only spends less than 3 seconds. UserS and UseA, who extract rules using predicted context reasons(Section 3.3) the most proportionally (20%), spend significantly less time than the average time consumption. The considerably low time consumption of the rule extraction process shows that SayRea imposes low interaction burdens on users and is easy to use. Figure 10 shows the time consumption for each user. It is worth noticing that this reported time include the contextual attributes selection process for validation purpose, which is not part of the approach.

6.1.4 Coverage. For each time a service is used, it is **covered** if SayRea is recommending it.

$$R_c = \frac{N_c}{N_a},$$

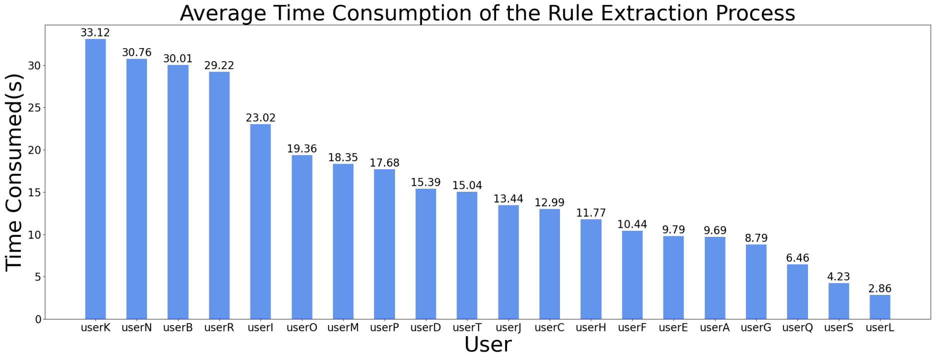


Fig. 10. Time consumption of each user.

where R_c represents coverage, N_c represents the times that services are covered, and N_a represents the total times of service usage. Our system can achieve an average coverage of 45% for all users at the end of the study. 30% participants have an overall coverage of more than 50% throughout the study. 95% participants exceed an overall coverage of 20%. One participant reaches an overall coverage of 69%. The coverage's coefficient of variance is 0.32, indicating a relatively evenly distributed coverage. Figure 11 shows the top 8 coverage of all users.

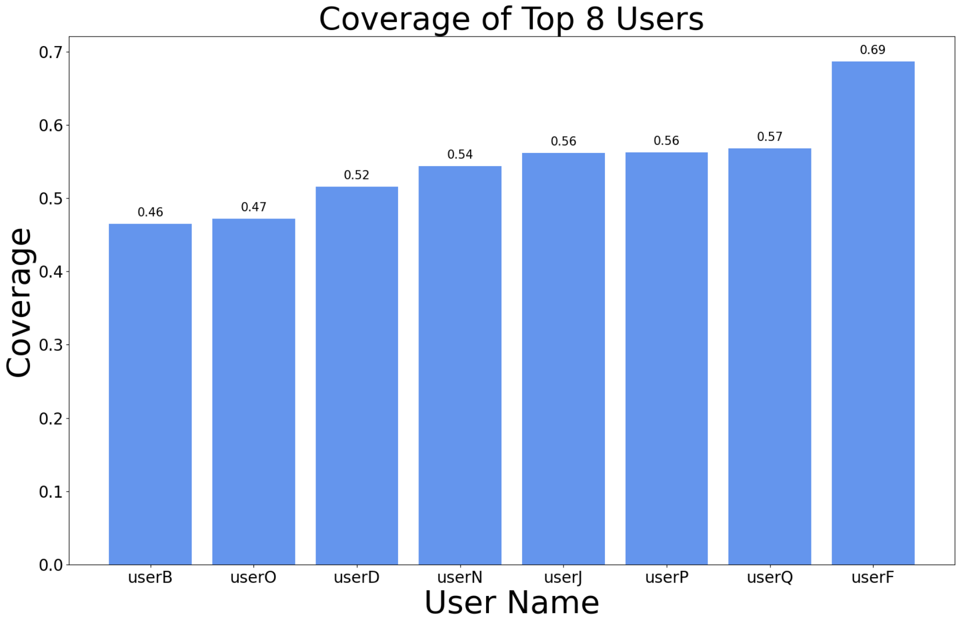


Fig. 11. Top 8 coverage of users

The average coverage shows a tendency to drop at the beginning of the study, and rise until being stable at around 45%. SayRea recommends recently used services as its cold start strategy, accounting for the high coverage at the beginning because of the locality of service usage. Relatively limited rules after the cold start stage cause a drop in the coverage. With the use of our system, rules are accumulated, resulting in coverage's increase. Figure 12 shows how average coverage changes over time from day 2 to day 10. To better display the tendency, we omit day 1, whose data contains users' random exploration of SayRea.

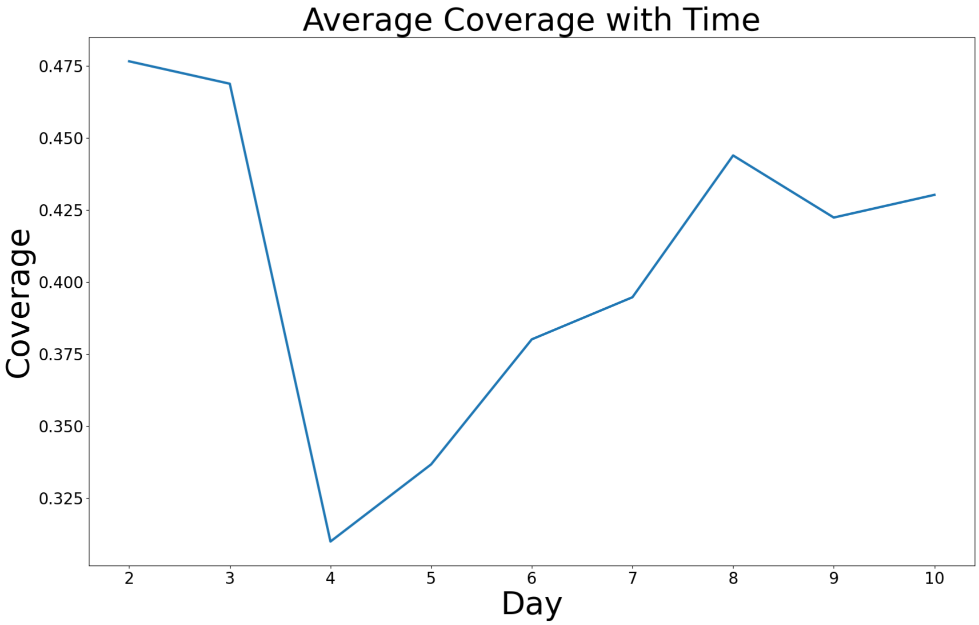


Fig. 12. Change of average coverage with time.

User study shows that the recommendations of SayRea can cover a significant part of users' service usage. Coverage can steadily increase over time before becoming stable. This shows our system's recommendation capability.

6.1.5 Subjective Metrics. The results of a 7-point Likert scale survey are displayed in Table 2. The results show that SayRea's usability, interpretability, and controllability have been recognized by most users, along with interactability, evolvability, and un-laboriousness.

We further discussed with participants why they find the system not hard to use (5.84 in 7, as shown in Table 2). Most thoughts point out that they feel SayRea can actually "understand" them with merely a reason. All they need to do is to say a simple reason without heavy burdens of thinking. Many participants also pointed out that SayRea was the first recommender system they had used that took users' own reasons into consideration, making them feel that they were in control of the system which they were building by themselves. This corresponds to the second statement shown in Table 2.

Table 2. Results of a likert scale survey.

Statements	Results
You can tell the reason for the recommendation when a service is recommended.	5.11(sd=1.21)
You feel you have participated in building the system.	5.47(sd=1.09)
You feel the recommender system is recommending services based on your habits.	5.11(sd=1.16)
You can interact smoothly with the system.	5.21(sd=1.24)
You feel the system's ability is evolving.	5.42(sd=1.18)
You feel it is not hard to use the system.	5.84(sd=0.93)

7 DISCUSSION

7.1 About Coverage

Although SayRea reaches an average of 45% coverage and a maximum of almost 70% coverage, it can be argued that there is room for improvement. We believe there are mainly two reasons for the improvable coverage.

- Users' usage of smartphones cannot be totally explained by context reasons[35]. We discussed this with several users who have relatively low coverage. Their reasons for the low coverage can be summarized as they just did not think their usage of service or their reasons have any relationship with the contexts taken into consideration, resulting in a weak or meaningless rule. SayRea now supports relatively limited types of contexts, which may account for the insufficient candidate context attributes. But some services, like opening a Camera to "take photos of a beautiful butterfly", just cannot be mapped to any context attribute according to interviewed participants.
- Users sometimes are not aware of the casual rules. We discuss in Limitation (Section 8.1) that the automatic discovery of new contextual rules can extract contextual rules without deliberate labeling, which can help users find unaware contextual rules to improve coverage.

7.2 Lack of Comparisons with Baselines

In addition to current context attributes and services, SayRea recommends based on users' own reasons, which are highly personalized and changeable. There lacks recommender system baselines that take users' own reasons into consideration. And even such baselines exist, results of our field study cannot be easily compared with those of laboratary experiments in order to demonstrate SayRea's performance. Further, existing recommender system baselines are difficult to run and evaluate[27], making comparable baselines for SayRea hard to exist. Our main novelty is that our approach is low-burden and in-situ, the considerably good results as shown in the user study further reduce the necessity of the comparisons with baselines.

8 LIMITATIONS

8.1 Automatic Discovery of Contextual Rules

SayRea extracts a contextual rule when it recognizes a service, but it is not always the case that the service is a result of a contextual rule. So the frequent request, even with a designed notification reminder, can still cause disturbance to users. Future systems can use sophisticated algorithms to evaluate the possibility of contextual rules and interact with users when such possibility is high. Moreover, they can incorporate automatic contextual rule discovery algorithms like association rule mining [28] to omit non-essential interactions and reduce the interaction burden. However, it is worth noticing that the above methods may require data-driven algorithms and numerous training corpus. The method proposed in this paper is a contextual rule extraction algorithm with high availability for the cold start when the amount of data is low.

8.2 Context Extension

In this paper, SayRea supports eight types of contexts. They do not fully describe the contexts of the service usage, which can confuse the user when selecting context attributes. However, SayRea has good contextual extensibility. Any new context can be converted into context attributes by context semanticization, and different context attributes can be understood with semantic information through LLM. Some works allow users to teach the intelligent system certain context acquisitions by PBD (program by demonstration), such as PUMICE [19]. SayRea can refer to these works to support interactively defined contexts, making full use of the user’s willingness to interact,

8.3 Further Management of Context Attributes

SayRea uses a context rule tree to manage contextual rules, in which all context attributes are considered homogeneous. However, various types of relationships may exist between different types and values of context attributes, such as mutual exclusion, intersection, inclusion, and approximation [15]. These relationships can be utilized to enhance the system’s understanding of the context. Context attributes can also be combined in other forms (e.g., logical or) for context causes in contextual rules (which can only be handled as different contextual rules in the current system), enriching the causes made of context attributes to express more complex contexts.

9 CONCLUSION

In this paper, we propose an in situ interactive method for extracting contextual rules. This contextual rule extraction method enables the low-burden user participation at the appropriate times (in situ) to extract contextual relationships between contexts and services with Large Language Models. It has the potential to be extended to other recommendation or contextual rule extraction fields.

SayRea, a context-based service recommender system with the contextual rule extraction method, has good usability, interpretability, and controllability. We implemented SayRea on Android and conducted field experiments. SayRea performed well in terms of rule accumulation and service coverage, indicating the high usability of the system. Subjective feedback from users showed good acceptance of SayRea, and users perceived the system as having well explainability and controllability.

REFERENCES

[1] Android. 2023. AccessibilityService|AndroidDevelopers. https://developer.android.com/reference/android/accessibilityservice/AccessibilityService

[2] Android. 2023. Make apps more accessible | Android Developers. https://developer.android.com/guide/topics/ui/accessibility/apps#describe-ui-element

[3] Apple. 2023. Shortcuts User Guide – Apple Support. https://support.apple.com/guide/shortcuts/welcome/ios

[4] Omer Boyaci, Victoria Beltran, and Henning Schulzrinne. 2011. Bridging Communications and the Physical World: Sense Everything, Control Everything. In *Proceedings of the 5th International Conference on Principles, Systems and Applications of IP Telecommunications* (Chicago, Illinois) *(IPTcomm '11)*. Association for Computing Machinery, New York, NY, USA, Article 14, 6 pages. https://doi.org/10.1145/2124436.2124455

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models Are Few-Shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 159, 25 pages.

[6] Andrew Cantino. 2021. Prompt Engineering Tips and Tricks with GPT-3.

[7] Manuel Carrasco Molina. 2019. *Siri and Search*. Apress, Berkeley, CA, 139–188. https://doi.org/10.1007/978-1-4842-4291-9_7

[8] Bachir Chihani, Emmanuel Bertin, and Noel Crespi. 2013. A user-centric context-aware mobile assistant. In *ICIN 2013 : 17th International Conference on Intelligence in Next Generation Networks*. IEEE Computer Society, Venice, Italy, 110 – 117. https://doi.org/10.1109/ICIN.2013.6670901

[9] Miguel Coronado and Carlos A. Iglesias. 2016. Task Automation Services: Automation for the Masses. *IEEE Internet Computing* 20, 1 (Jan 2016), 52–58. https://doi.org/10.1109/MIC.2015.73

[10] María del Carmen Rodríguez-Hernández and Sergio Ilarri. 2021. AI-based mobile context-aware recommender systems from an information management perspective: Progress and directions. *Knowledge-Based Systems* 215 (2021), 106740. https://doi.org/10.1016/j.knosys.2021.106740

[11] Jane Hoffswell, Arvind Satyanarayan, and Jeffrey Heer. 2018. Augmenting code with in situ visualizations to aid program understanding. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[12] Sergio Ilarri, Irene Fumanal, and Raquel Trillo-Lado. 2022. An Experience with the Implementation of a Rule-Based Triggering Recommendation Approach for Mobile Devices. In *The 23rd International Conference on Information Integration and Web Intelligence* (Linz, Austria) *(iiWAS2021)*. Association for Computing Machinery, New York, NY, USA, 562–570. https://doi.org/10.1145/3487664.3487806

[13] Dietmar Jannach, Sidra Naveed, and Michael Jugovac. 2017. User Control in Recommender Systems: Overview and Interaction Challenges. In *E-Commerce and Web Technologies*, Derek Bridge and Heiner Stuckenschmidt (Eds.). Springer International Publishing, Cham, 21–33.

[14] Riadh Karchoud, Arantza Illarramendi, Sergio Ilarri, Philippe Roose, and Marc Dalmau. 2017. Long-life application. *Personal and Ubiquitous Computing* 21, 6 (01 Dec 2017), 1025–1037. https://doi.org/10.1007/s00779-017-1077-2

[15] Riadh Karchoud, Philippe Roose, Marc Dalmau, Arantza Illarramendi, and Sergio Ilarri. 2019. One app to rule them all: collaborative injection of situations in an adaptable context-aware application. *Journal of Ambient Intelligence and Humanized Computing* 10, 12 (01 Dec 2019), 4679–4692. https://doi.org/10.1007/s12652-018-0846-8

[16] Oliver Korn, Albrecht Schmidt, and Thomas Hörz. 2013. The potentials of in-situ-projection for augmented workplaces in production: a study with impaired persons. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 979–984.

[17] V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10 (Feb. 1966), 707.

[18] Lei Li, Li Chen, and Ruihai Dong. 2021. CAESAR: context-aware explanation based on supervised attention for service recommendations. *Journal of Intelligent Information Systems* 57, 1 (01 Aug 2021), 147–170. https://doi.org/10.1007/s10844-020-00631-8

[19] Toby Jia-Jun Li, Marissa Radensky, Justin Jia, Kirielle Singarajah, Tom M. Mitchell, and Brad A. Myers. 2019. PUMICE: A Multi-Modal Agent That Learns Concepts and Conditionals from Natural Language and Demonstrations. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 577–589. https://doi.org/10.1145/3332165.3347899

[20] Amit Livne, Eliad Shem Tov, Adir Solomon, Achiya Elyasaf, Bracha Shapira, and Lior Rokach. 2022. Evolving context-aware recommender systems with users in mind. *Expert Systems with Applications* 189 (2022), 116042. https://doi.org/10.1016/j.eswa.2021.116042

[21] Yun Ma, Ziniu Hu, Diandian Gu, Li Zhou, Qiaozhu Mei, Gang Huang, and Xuanzhe Liu. 2020. Roaming Through the Castle Tunnels: An Empirical Analysis of Inter-App Navigation of Android Apps. *ACM Trans. Web* 14, 3, Article 14 (jun 2020), 24 pages. https://doi.org/10.1145/3395050

[22] X. Meng, Y. Du, Y. Zhang, and X. Han. 2022. A Survey of Context-Aware Recommender Systems: From an Evaluation Perspective. *IEEE Transactions on Knowledge amp; Data Engineering* 0, 01 (jun 2022), 1–20. https://doi.org/10.1109/TKDE.2022.3187434

[23] Xianghang Mi, Feng Qian, Ying Zhang, and XiaoFeng Wang. 2017. An Empirical Characterization of IFTTT: Ecosystem, Usage, and Performance. In *Proceedings of the 2017 Internet Measurement Conference* (London, United Kingdom) *(IMC '17)*. Association for Computing Machinery, New York, NY, USA, 398–404. https://doi.org/10.1145/3131365.3131369

[24] George W. Musumba and Henry O. Nyongesa. 2013. Context awareness in mobile computing: a review. *International Journal of Machine Learning and Application* 2 (Jan 2013), 1–10.

[25] Steven Ovadia. 2014. Automate the Internet With "If This Then That" (IFTTT). *Behavioral & Social Sciences Librarian* 33, 4 (2014), 208–211. https://doi.org/10.1080/01639269.2014.964593 arXiv:https://doi.org/10.1080/01639269.2014.964593

[26] Shaina Raza and Chen Ding. 2019. Progress in context-aware recommender systems — An overview. *Computer Science Review* 31 (2019), 84–97. https://doi.org/10.1016/j.cosrev.2019.01.001

[27] Steffen Rendle, Li Zhang, and Yehuda Koren. 2019. On the Difficulty of Evaluating Baselines: A Study on Recommender Systems. arXiv:1905.01395 [cs.IR]

[28] Iqbal H. Sarker and A.S.M. Kayes. 2020. ABC-RuleMiner: User behavioral rule-based machine learning method for context-aware intelligent services. *Journal of Network and Computer Applications* 168 (2020), 102762. https://doi.org/10.1016/j.jnca.2020.102762

[29] Christian Scheel, Angel Castellanos, Thebin Lee, and Ernesto William De Luca. 2014. The Reason Why: A Survey of Explanations for Recommender Systems. In *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, Andreas Nürnberger, Sebastian Stober, Birger Larsen, and Marcin Detyniecki (Eds.). Springer International Publishing, Cham, 67–84.

[30] Norbert Seyff, Gregor Ollmann, and Manfred Bortenschlager. 2014. AppEcho: A User-Driven, in Situ Feedback Approach for Mobile Platforms and Applications. In *Proceedings of the 1st International Conference on Mobile Software Engineering and Systems* (Hyderabad, India) *(MOBILESoft 2014)*. Association for Computing Machinery, New York, NY, USA, 99–108. https://doi.org/10.1145/2593902.2593927

[31] Zhihao Shen, Kang Yang, Wan Du, Xi Zhao, and Jianhua Zou. 2019. DeepAPP: A Deep Reinforcement Learning Framework for Mobile Application Usage Prediction. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems* (New York, New York) *(SenSys ’19)*. Association for Computing Machinery, New York, NY, USA, 153–165. https://doi.org/10.1145/3356250.3360038

[32] Chun-Hua Tsai and Peter Brusilovsky. 2021. The effects of controllability and explainability in a social recommender system. *User Modeling and User-Adapted Interaction* 31, 3 (01 Jul 2021), 591–627. https://doi.org/10.1007/s11257-020-09281-5

[33] Norha M. Villegas, Cristian Sánchez, Javier Díaz-Cely, and Gabriel Tamura. 2018. Characterizing context-aware recommender systems: A systematic literature review. *Knowledge-Based Systems* 140 (2018), 173–200. https://doi.org/10.1016/j.knosys.2017.11.003

[34] Zhiang Wu, Changsheng Li, Jie Cao, and Yong Ge. 2020. On Scalability of Association-Rule-Based Recommendation: A Unified Distributed-Computing Framework. *ACM Trans. Web* 14, 3, Article 13 (jun 2020), 21 pages. https://doi.org/10.1145/3398202

[35] Zhang Yujie and Wang Licai. 2010. Some challenges for context-aware recommender systems. In *2010 5th International Conference on Computer Science & Education*. IEEE, -, -, 362–365.

[36] Mohammed J. Zaki. 2004. Mining Non-Redundant Association Rules. *Data Mining and Knowledge Discovery* 9, 3 (01 Nov 2004), 223–248. https://doi.org/10.1023/B:DAMI.0000040429.96086.c7

[37] Chengqi Zhang and Shichao Zhang (Eds.). 2002. *Causality in Databases*. Springer Berlin Heidelberg, Berlin, Heidelberg, 85–120. https://doi.org/10.1007/3-540-46027-6_4

[38] Guoming Zhang, Lianyong Qi, Xuyun Zhang, Xiaolong Xu, and Wanchun Dou. 2019. Context-Aware Point-of-Interest Recommendation Algorithm with Interpretability. In *Collaborative Computing: Networking, Applications and Worksharing*, Xinheng Wang, Honghao Gao, Muddesar Iqbal, and Geyong Min (Eds.). Springer International Publishing, Cham, 745–759.

[39] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101. https://doi.org/10.1561/1500000066

[40] Ke Zhu, Yingyuan Xiao, Wenguang Zheng, Xu Jiao, and Ching-Hsien Hsu. 2021. A Novel Context-Aware Mobile Application Recommendation Approach Based on Users Behavior Trajectories. *IEEE Access* 9 (2021), 1362–1375. https://doi.org/10.1109/ACCESS.2020.3046654

A RESEARCH METHODS