Integration of Robot-Initiated Dialog into Task-Oriented Dialog by Adding Hidden Tasks - Application to Monitoring for Elderly

Masahiro Kawamura[1,*,†], Takatsugu Suzaki[1,†] and Masayuki Numao[1,†]

[1]The University of Electro-Communications, Chofugaoka 1 5 1, Chofu, Tokyo, Japan

Abstract

Dialog systems have been studied separately as task-oriented dialog (TOD) to accomplish a specific task, such as booking a trip, and non-task-oriented dialog (CC: Chit-Chat) to entertain the user for the purpose of the dialog and the continuation of the dialog itself. However, when dialog systems are applied to the elderly, not only TOD but also robot-initiated dialog, which is classified as CC, is required. This is because, in order to encourage users to use the system over the long term, it is necessary to implement a system that suggests useful tasks for the user, in addition to tasks that the user selects on his/her own, such as QA for schedule confirmation and physical condition management. The purpose of this study is to add autonomy to TOD to activate users and encourage long-term use. Therefore, we propose to define the tasks of the dialog system as hidden tasks and to integrate dialog robot-initiated dialog into TOD. In particular, the system is expected to enable long-term use of user data and minimally invasive evaluation of the user's cognitive functions in monitoring the elderly.

Keywords

Task-Oriented, Robot-Initiated, Chit-Chat, Dialog System, Elderly, Hidden Task

1. Introduction

Dialog systems have been studied separately as Task-Oriented Dialog (TOD) to accomplish specific tasks, such as travel reservations, and Non-Task-Oriented Dialog (CC: Chit-Chat) to entertain users for the purpose of the dialog or the continuation of the dialog itself. Therefore, as the conditions for constructing TOD, natural dialog and user consideration are required, but research on user consideration is not as advanced as CC on natural dialog. Therefore, this study proposes to integrate TOD and CC by using a technique of swapping user information storage and conversation control to expand the potential of TOD. Specifically, the proposed method makes it possible to integrate TOD and CC by defining tasks that operate separately from the original task, called hidden tasks. The hidden task is explained in section 3.

One application of this integrated method is in the area of elderly monitoring. When applying dialog systems to the elderly, not only TOD but also Robot-Initiated Dialog classified as CC are necessary. This is because, in order for users to use the system for a long time, it is necessary to implement a system that proposes useful tasks for users, such as checking schedules and managing their health, in addition to tasks that users choose themselves.

This paper describes the integrated method and its effective application.

1.1. Socially Responsible AI for Well-being

In this study, we propose the system that can be applied to elderly monitoring, which is believed to have the effect of reducing the social isolation of the elderly and improving or delaying cognitive decline. To measure the effectiveness of this system, we utilize a user information management ontology to measure factors such as the user's social relationships, memory, and sleep quality. These factors can be recorded as usage history, allowing for the system to capture dynamic changes. The system primarily uses interfaces such as voice recognition, speech synthesis, and display screens, with no variation in format based on the user, ensuring fairness. However, since the system includes a configuration to adapt dialog content to the user, there is a risk of losing fairness in determining the effects on the user based on usage frequency.

2. Related Work

TOD is mainly a retrieval-based system that defines a dialog scenario for each task, asks questions to collect information necessary to accomplish the task, and manages the scenario with a rule-based system. In the traditional method of slot filling[1][2][3] predefined frame structure with a set of slots is used to define a task, and the dialog is controlled by the status of slots. Recently, Seq2Seq,

AAAI 2023 Spring Symposia, Socially Responsible AI for Well-being, March 27–29, 2023, USA

*Corresponding author.

†These authors contributed equally.

k2231032@edu.cc.uec.ac.jp (M. Kawamura)

such as BERT, has realized a system that uses machine learning to pre-training a large dialog corpus to infer tasks from user input sentences and generate questions to gather information[4][5]. CC is also realized by a neural network pre-trained on a large dialog corpus.

Recently, however, there have been studies on the integration of the two systems, Kai et al.[6] have proposed an integration system by adding CC to enhance TOD. They reported that they succeeded in making the system look more natural and friendly by adding the immediate response from the CC to the tasteless conversation that only achieves TOD.

On the other hand, unlike Kai et al., we aim at the integration method that use CC in order to make also effective use of the free time when TOD is not being performed. This approach is also a solution to the problem that, in a monitoring system that requires all-day operation, TOD is defined as a conversation for medical checkups and schedule confirmation, but no conversation takes place during the time when the user is not engaged in such conversations unless the user asks.

3. Proposal

This section describes how to structure TOD and CC collaboration, the hidden task used as a clue for TOD and CC collaboration, and its application to monitoring the elderly. The main components of the dialog robot used in the experiment are described in section 4.

3.1. How to structure TOD and CC Collaboration

Here, we explain how to compose the coordination of TOD and CC under the assumption that the dialog system is always in operation. First, we consider the coordination of TOD and CC, and adapt TOD-based composition to both of them. The proposed system segregates intentions (boredom or tension) in chats and sets hidden tasks such as "getting close" or "relaxing", and fills slots with emotions inferred from tone of voice, facial expressions, gestures, topics, and so on, through multimodal coordination. Then, according to the hidden task, the speaker's response or the next topic is selected and the chatting proceeds. Specifically, the scenario proceeds by estimating the speaker's intentions, setting tasks, analyzing responses, deciding slots, and then evaluating action policies.

3.2. Hidden Task

Hidden Task assist the dialog robot in autonomous interaction, allowing it not only to integrate CC into TOD, but also to elicit information from the user and make diagnoses based on that information. To give an example, if the user's emotions are inferred from his/her facial expressions and boredom (intent) is read, a hidden task such as "activating the user" is given, which is a condition for selecting the next response or topic based on the topic and user information at that time. It is necessary to cooperate with the task of TOD, which is also operating originally, but this is made possible by making the task and the hidden task necessary conditions in the action policy decision. Figure 1 shows the configuration diagram. The module for setting hidden tasks is described in section 4.

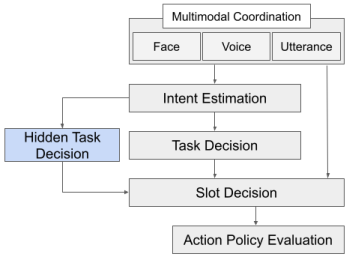

Figure 1: Proposed Data Flow Chart

3.3. Application to Monitoring The Elderly

We are interested in a monitoring robot for elderly and assume that it will be used by elderly. In TOD for elderly, it is required to conduct QA such as schedule confirmation and physical condition management. In addition, unlike ordinary TOD systems, it is essential to consider the use of the system in such a way that the system suggests the user some task that might be useful to the user, rather than in such a way that the system waits for the user to ask for a specific task the user chooses by himself/herself. Therefore, our method, integrating CC with TOD, is suitable for dialog robots for the elderly.

In this section, referring to the content of tasks, since elderly people spend a lot of time in institutions and homes, where there are few topics of conversation, it is essential for the dialog system to spontaneously provide topics for chats. Furthermore, when an elderly person shows a tendency toward dementia, it is necessary to ask questions to confirm the disease. Considering the consistency of the dialog and the structure of the scenario that takes the elderly into consideration, there is a requirement to ask questions in the form of chats rather than formalizing dementia screening.

Therefore, this study envisions the use of User Information Management Ontology (UIM Ontology) to manage user information and apply it to chats and the identifica-

tion of dementia tendencies. For example, when there is a topic about a favorite food, the user and the food can be managed in the ontology, and the user's favorite food can be guessed in the next dialog, or the memory of the previous dialog can be checked, thereby identifying the tendency toward dementia while entertaining the user as chats. A simple data flow diagram is shown below. The structure of UIM Ontology is also described in section 4.

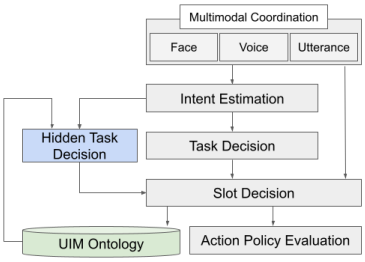

Figure 2: Proposed Data Flow Chart for The Monitoring The Elderly

4. Implementation

The proposed system mainly consists of four modules (Interface Manager, Interpreter, Dialog Manager, and Generater), and the roles of each module are based on existing dialog systems for TOD[7]. The configuration diagram is shown in Figure 3. First, Interface Manager controls actuators (speakers, displays, etc.) used for the robot's actions and sensors (microphones, cameras, etc.) used for understanding user's requests, and integrates data acquired from each sensor. Next, Interpreter is responsible for semantic interpretation and structuring the data acquired from the interface, such as Natural Language Understanding(NLU). Then, Dialog Manager updates the state of the scenario and selects the robot's actions. Finally, Generator generates the robot's actions from the dialog state by using interfaces such as Natural Language Generation(NLG). Current implementation includes simplified versions of intention estimation and slot decision in the Interpreter. Each implementation is described in the specific module description section. The hidden tasks and UIM ontology of the proposal are also described as subcomponents. Then, the scenario description method and its application to dementia screening will be explained with dialog examples.

4.1. Main Components of The Proposed System

Dialog Manager(DM) Dialog manager performs dialog state updating and action selection, and uses a finite

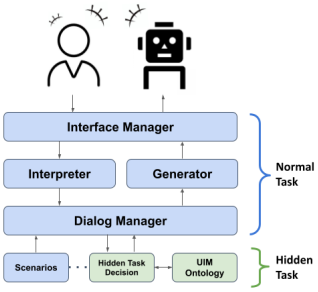

Figure 3: Configuration of the dialog system

state machine to control the dialog. The current intention and the frame representation are used for the dialog state. The task of a state in a finite state machine is to fill a frame representation. The dialog state is maintained until the end of the current scenario. There are two ways of state transitions: when a frame representation is entered, and when metacommands such as timeout, repeat, cancel, etc. are used. The method of state transition when a frame representation is input is determined by the slot value of the frame (e.g., if a slot value is empty or filled) and can be defined in a scenario. In Figure 3, DM determines the next task based on a frame received from Interpreter and scenarios read by Scenarios, and passes the next action obtained from Action Policy Evaluation to Generator.

Interface Manager For input, we implemented voice recognition from a microphone, keyboard input, various sensors (visible light camera, thermo camera, air temperature sensor, and humidity sensor) for temperature sensing, and metacommands using buttons. For output, functions such as speech synthesis and image display are implemented. Speech recognition and speech synthesis are performed using Google Cloud API Speech-to-Text and Text-to-Speech, respectively.

Interpreter The Interpreter consists of a NLU module for understanding the meaning of the user's speech, a body temperature estimation module for estimating body temperature from several sensors, and a person recognition module for estimating facial expressions and hand poses from a camera. The NLU module performs rule-based intention estimation and slotting, and implements keyword extraction methods using morphemes, regular expressions, and the Knowledge Base (KB), which can be selected on the scenario description. The temperature estimation module is implemented using machine learning with multiple regression analysis. The person recognition module is implemented by using mediapipe to extract contours and calculate angles to determine facial expressions and hand poses.

The files contained in the KB are divided into three main types: an intention dictionary, a slot dictionary, and a meta-word dictionary. The intention dictionary describes a combination of scenario names and keywords, and enables the user to invoke scenarios by inferring intentions from the keywords in the utterances. The slot dictionary is divided into json files according to abstraction level, and in order of abstraction level, the files are (slot: pattern name), (pattern name: pattern). Slots are extracted by pattern of the assigned pattern name. Finally, the meta-word dictionary describes combinations of commands and keywords, allowing users to freely manipulate the scenario using commands by uttering keywords during scenario activation.

Generator The Generator consists of a NLG module for generating the robot's speech text and an image acquisition module for displaying images. The NLG module generates the speech text by assigning the dialog text defined in the scenario to the dialog state. The image display module is implemented in such a way that it retrieves and displays images by specifying the path of the location where the images are stored or URL .

4.2. Subcomponents of The Proposed System

Hidden Task Decision In Figure 3, it is placed outside DM to make it clear that it was added later to the main structure of the dialog system, but as shown in Figure 2, it plays the same role as Task Decision, so Hidden Task Decision is implemented within DM. The role of Hidden Task Decision is to determine the hidden task by estimating the user's intention in his/her free time, as described in section 3. Here, a diagram of an example dialog is shown below.

The role of Hidden Task Decision is to determine the hidden task by estimating the user's intention in his/her free time, as described in section 3. Here, a diagram of an example dialog is shown below. The blue frame is user-initiated and the green frame is robot-initiated. The flow of the dialog is as follows: First, the TOD has a high priority, so when the user asks a question, the robot responds to it with priority. In this example, the user greeted the robot, so the first task is to respond to the greeting. At the same time, since the task after the greeting is not set in stone, a hidden task works for the next topic. Here, the hidden task of "activating the user" is set to continue the dialog, since the user's emotion recognition by his/her facial expression was read to be "smile". Then, in the middle of this dialog, the scenario is interrupted by the user's question, "Wait a minute". In this way, the introduction of Hidden Task Decision enables the system to present topics according to the user's needs while fulfilling its role as a TOD. In addition, in order to make the dialog more user-friendly, we have added UIM Ontology in this study.

Figure 4: Example of Proposed System Usage

User Information Management Ontology (UIM Ontology) UIM Ontology only interacts with Hidden Task Decision as shown in Figure 4. Therefore, functions are defined in Hidden Task Decision to manipulate the ontology using owlready2 in order to mediate with the ontology. In this section, we describe how the ontology is created by the manipulation.

An example of the ontology is shown in Figure 5. We assume that the user's name is Yoko. First, we describe the input example. When the user advances the topic of his/her favorite food, the user's favorite food comes back as a slot value. Then, the slot value is entered as a subclass in the food class. Furthermore, we connect the user's name and the food's subclass with a property. In this way, information about the user can be accumulated each time the dialog is repeated. As an example of output, when the topic of lunch is first advanced, information about the food is given. The information about the food is queried to the ontology and relevant information is elicited. In this case, we know that it is the user's preference, and this information can be applied to the dialog.

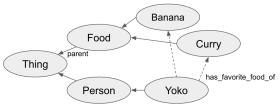

Figure 5: Example of Proposed System Usage

4.3. Scenario Description and Execution

As for scenario description, scenario languages such as AIML for ChatBot have been proposed[8], which use question-answer templates to compose dialog. These languages can handle simple questions and answers, but they are insufficient for TOD where multiple questions

must be asked. Therefore, we designed a scenario language that can describe complex questions flexibly. The proposed scenario descriptions can be defined by XML. The scenario defines the domain, the control of the frame representation requested to the user, the actions of the system to ask the user for the frame representation, and the way to cooperate with external functions. The following is an example of scenario description by HDS-R.

Listing 1: hdsr_scenario.xml

```
<scenario name="hdsr">
  <sequential>
    <frame name="greeting" order="1">
      <actions>
        <action device="speaker"> Hello, let's begin diagnosis
        . </action>
      </actions>
    </frame>
    <frame name="qname" order="2">
      <actions>
        <action device="speaker"> How old are you? </action>
      </actions>
      <request timeout="15">
        <slot name="age"/>
      </request>
    </frame>
    ...
  </sequential>
</scenario>
```

The scenario tag is the root tag, with the attribute name representing the intent. The sequential tag selects child elements in sequence until all of them have been executed. The sequential tag plays the role of action selection. In addition to the sequential tag, the following action selection tags are implemented: conditional, which selects child elements according to conditions, random, which selects child elements at random, and loop, which keeps selecting until the conditions are satisfied. These action selection tags can define action selection tags, frame tags, etc. for child elements, and can be used for scenarios with complex state transitions. The frame tag is used to fill the frame representation. The actions tag is a tag that summarizes the actions of the system and simultaneously executes the contents of the actions tag of the child elements. The request tag specifies a set of slots that the user is expected to fill. The slot tag defines the slots and can specify attributes such as default values or a list of slot values.

4.4. Application to Dementia Screening

In this section, we explain the application to monitoring system for elderly. The system is normally working as TOD with multiple tasks: if the user asks to perform some task, the system responses as a normal TOD. In the proposed method, during free time, system will start CC by selecting non-serious topics, such as greeting, simple game, etc. During conversation he system collects the user's responses and predictions are made based on the conversation data obtained by the topics. When the possibility of depression or dementia is detected, the transition can be also made from CC to HDS-R.

Early Detection Methods for Dementia Medical tests for dementia, such as Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET), have been cited as necessary but time-consuming and expensive. There are also methods such as the Mini Mental State Examination (MMSE) and the Hasegawa Dementia Scale-Revised (HDS-R), but direct question-and-answer sessions with these professionals may be invasive to the patients. Therefore, patient-oriented testing is now required. As an example, an automated screening test based on the processing of patients' conversational and communicative abilities is attracting attention.

It has been found that the language ability of patients with dementia is inferior to that of normal subjects, and the evaluation of language ability based on the interaction with the dialog system can be used to statistically predict the suspicion of dementia. Furthermore, prediction can be made based on acoustic features such as voice condition and time between utterances.

Linkage between CC and HDS-R We describe the structure of scenarios to accomplish the hidden task of diagnosis. As scenarios for diagnosis, we attempted to create scenarios by referring to questions about proverbs[**?**] and the HDS-R, which are considered to be a tool that can easily reveal the tendency to dementia. For questions measuring cognitive abilities (arithmetic, sequencing, object recognition, image recognition, recitation, regression, and repetition), questions corresponding to the HDS-R[**?**] were used. For environmental register (name, age, place, date), we used UIM Ontology to automatically make correct and incorrect decisions.

The Developed Dialog Robot So far, we have evaluated the functionality of the dialog robot.

As for the functionality of the dialog robot, the developed dialog robot is equipped with a pre-installed dialog system (Raspberry Pi 3, speakers, microphones, and other devices) in a familiarly shaped container designed for use by the elderly(figure 6). It also implements a specification that allows the user to switch between voice(figure 7) and text-based(figure 8) dialogs on the browser screen. When we conducted a survey of developers and related parties for evaluation, many of them requested a dialog to confirm that metacommands were used and that speech recognition failed. In the future, it is necessary to improve these functional aspects and make the system more suitable for use by the elderly.

Figure 6: WANCO: Dialog Robot for Monitoring The Elderly

Figure 7: Browser Screen Output of WANCO: Example of dialog using image display

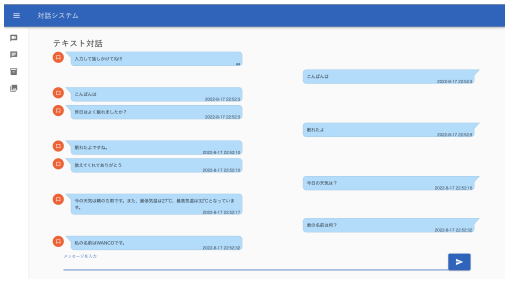

Figure 8: Browser Screen Output of WANCO: dialog history

References

[1] O. Lemon, K. Georgila, J. Henderson, M. Stuttle, An ISU dialogue system exhibiting reinforcement learning of dialogue policies: Generic slot-filling in the TALK in-car system, in: Demonstrations, 2006, pp. 119–122. URL: https://aclanthology.org/E06-2009.

[2] Z. Wang, O. Lemon, A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information, in: Proceedings of the SIGDIAL 2013 Conference, Association for Computational Linguistics, Metz, France, 2013, pp. 423–432. URL: https://aclanthology.org/W13-4067.

[3] S. Young, M. Gašić, B. Thomson, J. D. Williams, Pomdp-based statistical spoken dialog systems: A review, Proceedings of the IEEE 101 (2013) 1160–1179. doi:10.1109/JPROC.2012.2225812.

[4] T. Zhao, A. Lu, K. Lee, M. Eskénazi, Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability, CoRR abs/1706.08476 (2017). URL: http://arxiv.org/abs/1706.08476. arXiv:1706.08476.

[5] C. Yu, C. Zhang, Q. Sun, A chit-chats enhanced task-oriented dialogue corpora for fuse-motive conversation systems, 2022. URL: https://arxiv.org/abs/2205.05886. doi:10.48550/ARXIV.2205.05886.

[6] K. Sun, S. Moon, P. A. Crook, S. Roller, B. Silvert, B. Liu, Z. Wang, H. Liu, E. Cho, C. Cardie, Adding chit-chats to enhance task-oriented dialogues, CoRR abs/2010.12757 (2020). URL: https://arxiv.org/abs/2010.12757. arXiv:2010.12757.

[7] S. Ultes, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, D. Kim, I. Casanueva, P. Budzianowski, N. Mrkšić, T.-H. Wen, M. Gašić, S. Young, PyDial: A multi-domain statistical dialogue system toolkit, in: Proceedings of ACL 2017, System Demonstrations, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 73–78. URL: https://aclanthology.org/P17-4013.

[8] R. S. Wallace, The Anatomy of A.L.I.C.E., Springer Netherlands, Dordrecht, 2009, pp. 181–210. URL: https://doi.org/10.1007/978-1-4020-6710-5_13. doi:10.1007/978-1-4020-6710-5_13.