Doctoral Dissertation

Reflective Response of Dialogue System Focusing on User's Event

Shohei Tanaka

March 17, 2023

Graduate School of Science and Technology
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Science and Technology,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Shohei Tanaka

Thesis Committee:

Professor Satoshi Nakamura	(Supervisor)
Professor Taro Watanabe	(Co-supervisor)
Associate Professor Katsuhito Sudoh	(Co-supervisor)
Affiliate Professor Koichiro Yoshino	(Co-supervisor)
Professor Giuseppe Riccardi	(University of Trento)

Reflective Response of Dialogue System Focusing on User's Event *

Shohei Tanaka

Abstract

This dissertation addressed dialogue systems that generate reflective responses and actions to user utterances. The existing dialogue systems tend to generate not reflective responses and actions that are passive to the user utterances. We proposed architectures to generate reflective responses and actions by focusing on user's events to solve this problem. Since dialogue systems are traditionally categorized into non-task-oriented dialogue systems or task-oriented dialogue systems, we tackled the following three problems of dialogue systems based on this categorization. First, we proposed a model to generate reflective responses on non-task-oriented dialogue. The model selects reflective responses based on events included in user utterances and system responses. Second, we investigated a model to select reflective actions on task-oriented dialogue. The model selects reflective actions based on causality relations between events included in user utterances and system actions. Finally, we developed a model that integrates multimodal information to select reflective actions on multimodal task-oriented dialogue. The model selects reflective robot actions by utilizing user utterances and events included in situations surrounding the user.

Keywords:

dialogue system, event, causality, ambiguous request, reflective response, reflective action

*Doctoral Dissertation, Graduate School of Information Science,
Nara Institute of Science and Technology, March 17, 2023.

Contents

List of Figures

1. Introduction

This dissertation describes our studies on the reflective responses of dialogue systems. Since existing dialogue systems struggle to generate reflective responses to user utterances, we solved this problem by focusing on user's events. This chapter outlines the scope of the dialogue systems investigated in this dissertation, the specific definition of the research problem of non-reflective responses, and its approaches and contributions to the research problem.

1.1. Background

Dialogue systems, which support users through dialogues in various tasks, are categorized as non-task-oriented or task-oriented, depending on their purposes, including daily conversation, tourist navigation, or domestic chores. The main purpose of non-task-oriented dialogue systems is to entertain a user through daily conversation. Building a good relationship with a user through conversation can relieve stress or facilitate negotiations that benefit both the system and the user, such as product sales.

Task-oriented dialogue systems aim to satisfy user requests for specific tasks, such as tourist navigation or domestic chores. Dialogue systems that support users in limited tasks have already been put to practical use as smart speakers [1] and digital signage [6]. Although these systems have traditionally been categorized into different systems, they share the same purpose: supporting and satisfying users. Interest is growing in the development of systems that can interact with users in non-task-oriented and task-oriented dialogues by integrating these systems [64, 114]. In other words, future dialogue systems will not distinguish between them; they will satisfy users in both situations.

Most existing dialogue systems generate responses and actions to user utter-

ances by learning them from large corpora [104,108]. However, the responses and actions learned from the current large corpora tend to follow what users explicitly requested. In other words, the responses and actions generated by existing dialogue systems are often passive. To solve this problem, this dissertation proposed a new definition of the reflective responses of dialogue systems to achieve reflective responses for them that focus on user's events in non-task-oriented and task-oriented dialogues.

1.1.1. Reflective Responses of Dialogue Systems

Reflective generally means "thinking deeply about something" [5]. On the other hand, the *reflective* responses and actions of dialogue systems in this dissertation denote responses and actions that are not explicitly requested by users, although they probably satisfy them. For example, a user said "I'm exhausted." Our non-task-oriented dialogue system generates such reflective responses as "Why don't you relax?" The response can be regarded as reflective because it is not simple back-channeling or a sympathetic response and encourages the user to talk more with the system. Our task-oriented dialogue system takes such reflective actions as "Should I search for a cafe around here?" This action is reflective because it is not explicitly requested by the user, although again it probably satisfies a user's implicit needs. Although these responses and actions have different system intentions depending on their tasks, they share an identical purpose: satisfying the user by including contents that they didn't actually request. These reflective responses and actions resemble feedback that satisfies a user's potential needs, among those included in information seeking, which includes questions to which the user seeks answers, and directives, which include suggestions of interest to the user, for dialogue acts [17,18,111]. Humans can take reflective actions using commonsense reasoning [15] based on their experiences and knowledge. For example, when their interlocutor complains, "I'm exhausted," a person might suggest "How about going to a cafe?", based on such commonsense reasoning as "exhausted" $\rightarrow$ "need a rest" $\rightarrow$ "go to a cafe." The dialogue systems of this dissertation also utilize commonsense reasoning for generating reflective responses and actions.

Reactive and *proactive* have similar meanings to *reflective*. *Reactive* generally means "behave in response to what happens to them, rather than deciding in

advance how they want to behave" [4]. Its meaning resembles the responses of existing dialogue systems that follow a user's requests [31, 33]. These reactive responses and actions can be viewed as replies that are found in such simple back-channeling of auto/allo-feedback as "I see" and information providing, which just follows a user's requests, such as "I reserved a hotel," of dialogue acts [17, 18, 111]. For example, when a user says, "I'm exhausted," existing non-task-oriented dialogue systems tend to generate such simple, rather banal responses [104] as "I see." In addition, existing task-oriented dialogue systems cannot generate responses because they cannot understand user requests [16]. *Proactive* generally means "intended to cause changes, rather than just reacting to change" [3]. Such a system takes the initiative and offers suggestions to the user. In other words, the proactive actions of dialogue systems are the efforts/steps taken on tasks in which the system assumes control, such as advertisements or promotion [48]. These proactive responses and actions can be regarded as answers that are included in directives, which include suggestions that directly benefit the systems, such as "Based on your purchase history, I recommend this product to you," of dialogue acts [17, 18, 111]. Based on this definition, proactive responses are more suitable for task-oriented dialogues than for non-task-oriented ones. For example, when the user says, "I'm exhausted," if the systems are on streaming services [48], dialogue systems with proactive responses offer such suggestions as "How about relaxing by watching a movie?" Although proactive responses are similar to reflective responses, they differ based on whether they focus on the benefits of systems or the potential demands of users.

1.1.2. User Events in Dialogue

To achieve the reflective responses of dialogue systems defined in Section 1.1.1, this dissertation focuses on the events included in user utterances or situations surrounding them and utilizes them as their events. *Event* generally means "something that happens, especially when it is unusual or important. You can use events to describe all the things that are happening in a particular situation." [2]. Everything that happens in the world is an event based on this definition. Predicate-argument (PA) structure [22, 23], which consists of a predicate and its arguments, is an event representation of the existing studies of natural

language processing. PA structure is the most general event representation because it is domain-independent. Abstract Meaning Representation (AMR) [9] is a graph depiction that represents the predicates and the arguments of PA structures as nodes and their dependency as edges. AMR is suitable for the graphical understanding of events in texts. In addition, specific things can be defined as events based on the tasks of systems. For example, Kim and Klinger [51] defined aspects that trigger the emotions of characters as events for analyzing emotions in fictional texts.

This dissertation treated events mentioned by a user as items that are included in user utterances. For example, if the user says, "Hello, John. Actually, I'm exhausted," *I'm exhausted* becomes an event mentioned by the user and included in their utterance. Humans can focus on the events mentioned by their interlocutors to make such reflective suggestions (and pose questions as well) as "You must relax," which are in-depth to the contents of the interlocutor's utterances [118]. Therefore, this dissertation aims to develop architectures for response generation and action selection based on the events included in user utterances to create reflective responses of dialogue systems.

Situations, where the user and the system are engaged in a dialogue, can include events that are not mentioned by the user even though they are relevant to them. For example, "The user has a glass" and "There is a glass in the kitchen" are also events. Users usually do not mention every such surrounding event. Humans can focus not only on their interlocutor's utterances but also on the events surrounding it to take such reflective actions as bringing a glass when the interlocutor says, "Let's have a drink." This dissertation aims to achieve reflective action selection for a dialogue system by focusing on the events surrounding users in addition to those included in their utterances.

We described above how the responses and actions of existing dialogue systems are non-reflective because they statistically learn responses and actions from large corpora that consist of non-reflective responses. In contrast, considerate humans can make reflective responses and actions that they inferred from the contents of their interlocutor's utterances and the surrounding situations. This dissertation proposed response generation and action selection by focusing on user's events, which are included in user utterances and surrounding situations, for making dia-

logue systems that resemble considerate humans. In other words, this dissertation summarizes ambitious efforts to incorporate example-based causal inferences as done by humans into existing dialogue systems based on statistical methods.

1.2. Problems and Existing Work

In Section 1.1, we explained that the goal of this dissertation is to create reflective responses for dialogue systems. We focus on events included in the utterances and situations surrounding users to achieve this goal. In this section, we describe the problems of non-reflective responses and actions, the reasons why focusing on user's events helps generate reflective responses and actions, and the challenges for creating reflective responses in dialogue systems. Our system incorporates three perspectives including both non-task-oriented and task-oriented dialogues. We also describe existing work.

1.2.1. Dull Response Problems on Non-task-oriented Dialogues

As defined in this dissertation, reflective response has a broad meaning. A specific metric is necessary to conduct a subject evaluation for the responses of a dialogue system. Dialogue continuity [12], which is a metric for evaluating the user satisfaction for a non-task-oriented dialogue system, indicates whether a user wants to continue a conversation with a system. Responses with high dialogue continuity can be looked upon as replies that are included in information seeking, including questions that the user wants to answer, and directives, which include suggestions in which the user is interested, of dialogue acts [17,18,111]. Although Neural Conversational Model (NCM) [104] has been researched widely, such dialogue models often generate simple and dull responses due to the limitation of their ability to take dialogue context into account. These dull responses are basically empty responses that provide no useful information, examples among which are those included in the simple back-channeling of auto/allo-feedback, such as "I see," and information providing, which just expresses agreement or disagreement, such as "OK," of dialogue acts [17, 18, 111]. As a result, dialogue continuity decreases,

and dialogue breakdown happens, creating a dull response problem [58]. A loss function based on Maximum Mutual Information (MMI) [58] and the Mechanism-Aware Neural Machine [36] addressed the dull response problem by focusing on response diversity. Although these methods improved response diversity, they failed to improve dialogue continuity because response diversity is based on the relationship between responses.

Bohus and Rudnicky [12, 71] proposed a method based on the coherency between a user utterance and a system response. Their method focused on dialogue continuity to improve response generation. A specific definition of coherency is required to deal with the coherency between a user utterance and a system response. In this study, related predicate-argument (PA) structure pairs, such as *be stressed out* and *relax*, are treated as event pairs that are included in user utterances and system responses. PA structure is a central semantic representation of a sentence. The coherency of the PA structures in two sentences is related to the coherency between them. In other words, when a PA-structure pair in a user utterance and a system response has high coherency, the latter is coherent with the former.

Event causality is one relation between PA structures that have coherency. Event causality is defined as the relation of a cause and an effect between two PA structures [93,94]. For example, based on this definition, *be stressed out* is a cause and *relax* is an effect. Event causality relations have been used in why-question answering systems to focus on the causalities between questions and answers [79–81]. A non-task-oriented dialogue system using event causality relations can also generate responses preferred by users [34]. Unfortunately, these studies did not investigate whether dealing with the coherency between PA structures in utterances improves the coherency between utterances and dialogue continuity.

Coherence Model [109] is another study that focuses on the idea of coherency. Coherence Model estimates the coherency of a target sentence to the antecedent document based on the part-of-speech tags of the words that appear in the document and the distributed representation of the sentence. Cervone et al. [19] used the coherency scores output by this model for dialogue response generation. Coherence Model does not investigate whether the model improves the coherency of the system responses of the user utterances and the dialogue continuity.

Table 1.1.: Levels of ambiguity in requests (queries) [98, 99]

Level	Definition
Q1	Actual, but unexpressed request
Q2	Conscious, within-brain description of a request
Q3	Formal statement of a request
Q4	Request as presented to the dialogue system

1.2.2. Ambiguous User Requests on Task-oriented Dialogue

Unfortunately, existing spoken dialogue systems assume that users will provide clear, specific requests to systems [113], overlooking that their requests are sometimes ambiguous [110]. *Ambiguous user request* denotes that users failed to clearly define and verbalize their requests even when they have such potential appeals [110]. Taylor [98, 99] categorizes user states in an information search into four levels by their clarity (Table 1.1.) Most existing task-oriented dialogue systems [69, 102] convert explicit user requests (Q3) into machine readable expressions (Q4). Future dialogue systems need to take appropriate actions even in such situations as Q1 and Q2, where the users fail to clearly verbalize their requests [110].

User query disambiguation is another conventional, important research issue in information retrieval [29, 55, 101, 105]. These studies mainly focused on problems of lexical variation, polysemy, and keyword estimation. In contrast, our study focuses on cases where the user intentions are unclear.

Interactive systems that shape user intentions are another research trend [40, 42]. Such systems clarify user requests by asking clarification questions. Both studies assume that the user has a clear goal request; our system assumes that a user's intention is ambiguous. In the corpus collected by Cohen and Lane [27], which assumes a car navigation dialogue system, the system responds to user requests classified as Q1, such as suggesting a stop at a gas station when the user needs gas. Our study collected a variety of ambiguous user utterances to cover wider situations.

When user requests are ambiguous, human guides can reflectively recommend information needed by users. For example, when a user comments, "I love the view here," a guide might respond, "Should I take a picture for you?" This ability is derived from the experience accumulated by guides who deal with a variety of ambiguous requests and infer causal relationships from them. In other words, they choose reflective actions for ambiguous user requests using generalized appropriate patterns distilled from accumulated information.

Two problems surface when we implement such action selection in a dialogue system. First, a large training corpus is necessary to use the statistical methods that are essential for recent systems [48,110]. The Wizard of Oz (WOZ) method, in which two subjects play user and system roles, is widely used for collecting a user-system dialogue corpus [16, 48]. However, non-expert humans, who are not concierges, struggle to respond reflectively to every ambiguous user request. Second, since a system's actions are constrained by its API calls, the collected actions sometimes are infeasible. In addition, ambiguous user requests can be regarded as the antecedent requests of multiple system actions. For example, if functions *searching for fast food* and *searching for a cafe* are invoked to satisfy antecedent request "I'm hungry," both are reflective actions. Completely annotating multi-class labels is impractical in actual data collection [61]. We define the problem of training a model on incompletely annotated data in which only one system action is associated with a user request. We tested with completely annotated data where multiple actions are associated with a request.

Moreover, since human experts use causal relations generalized from their own experience, we expect our system to achieve more accurate classification if it uses knowledge distilled from training data.

1.2.3. Difficulty of Selecting Reflective Action on Text-based Dialogues

Both dialogue systems in Sections 1.2.1 and 1.2.2 assume text-based dialogues. Humans can take reflective actions based not only on the utterances of their interlocutors but also on situations surrounding the same interlocutors. For example, when an interlocutor says, "There isn't another one," taking a reflective action

is difficult based solely on the utterance. If we see him holding a glass, we might take such a reflective action as bringing another glass. Implementing such a multimodal dialogue system is an important challenge to mount a dialogue system on a robot that has a physical body [7, 24, 70]. If we implement a dialogue system on a robot, it must derive the reflective actions required by users from user utterances and their surrounding situations. In other words, the systems need to integrate events obtained from text, speech, images, and other observations for situation understanding. Existing interactive robots/agents using multimodal features have focused on question answering from images [54, 100], request analysis [37], and conversations about images [24, 56, 60, 115]. However, they have not yet tackled the problem of identifying the needed reflective actions from ambiguous user requests and the results of situation understanding. No current corpus possesses such interactions.

To build such systems in the real world, we also face the problem of dataset scalability [44]. Although conventional machine learning methods require large-scale training data, collecting such a large-scale dataset for individual robots is impractical because we need to collect it based on the physical characteristics of each robot [57]. One possible solution is transferring the data collected in a simulated world to the actual world (sim2real); however, transferring the knowledge acquired in simulated worlds remains challenging [107]. Effective feature extraction must be investigated so that systems can work in actual situations [112].

1.3. Approaches in this Dissertation

In Section 1.2, we defined the obstacles that complicate achieving the reflective responses of dialogue systems in this dissertation from three perspectives. This section outlines the proposed solutions to the problems and our experimental results.

1.3.1. Non-task-oriented Response Re-ranking

In Section 1.2.1, we defined the reflective responses of non-task-oriented dialogue systems as those have high dialogue continuity. We also described a method that deals with the coherency between PA structures (events) in user utterances and

system responses to improve dialogue continuity. We improved the dialogue continuity of system responses using this method by proposing a response re-ranking method that focuses on the coherency of PA structures included in response candidates and a dialogue context. Re-ranking selects candidates based on any metric in such language generation tasks as why-question answering and dialogue systems [11, 45, 80, 82].

We proposed methods that utilize event causality relations or Coherence Model to implement re-ranking. Our methods re-rank response candidates by calculating the coherence scores between PA structures based on event causality relations or Coherence Model to select responses with high dialogue continuity. We used event causality pairs extracted from a large-scale corpus [93, 94] to calculate the scores. We also used distributed event representation based on the Role Factored Tensor Model (RFTM) [106] for the robust matching of event causalities. We experimentally evaluated coherency using Pointwise Mutual Information (PMI) in addition to automatic evaluations using reference responses. A human evaluation was conducted to determine whether the proposed method improved the coherency of dialogue contexts and dialogue continuity.

Experimental results showed that although these methods improved the coherency in such automatic evaluations as PMI, they actually decreased the coherency in human evaluations, although they improved dialogue continuity. The results seem contradictory. Based on these results, we formalized and analyzed the following three hypotheses: (1) Improving coherency based on words does not necessarily contribute to greater coherency in human evaluations. (2) Coherency and dialogue continuity in human evaluation have a low correlation. (3) Improving coherency based on words, rather than coherency in human evaluation, improves dialogue continuity in human evaluations. We conducted a correlation analysis of the scores of human evaluation and a case analysis. Our results showed that the three hypotheses are satisfied to some extent. In addition, our results suggest that dialogue continuity improves when the methods select responses that include PA structures related to dialogue contexts.

1.3.2. Reflective Action Selection on Text-based Dialogue

In Section 1.2.2, we described how collecting a corpus consisting of reflective actions is complicated because taking reflective actions is difficult even for humans. To solve these problems, we pre-defined 70 system actions and asked crowdworkers to provide antecedent requests for which each action could be regarded as reflective. Bapna et al. [10] collected a corpus and modeled the collection process with pre-defined dialogue acts. This corpus assumes that a user has a clear goal request; our corpus assumes that their intention is ambiguous.

Another problem is that when user requests are ambiguous, multiple system actions can be regarded as reflective for one ambiguous user request. Thus, we investigated whether ambiguous user requests have other corresponding system actions among the 69 actions other than those pre-defined in the corpus collection. We isolated a portion of the collected ambiguous user requests from the corpus and added more annotations using crowdsourcing. Our results show that an average of 8.55 different actions for one ambiguous user request were additionally regarded as reflective, even when choosing from the 69 system actions. On the other hand, completely annotating multi-class labels is impractical in actual data, as described in Section 1.2.2.

To train the model on incomplete training data, we applied the positive/unlabeled (PU) learning method [20, 32], which assumes that some data are annotated as positive, but not all of them. In addition, to achieve causal knowledge distillation as human experts do, we introduced a causality detection model in which ambiguous user requests and reflective system actions are defined as causes and effects. We incorporated a causality detection model as additional features for the reflective action classification model. The experimental results show that both the PU learning method and the causality detection model improved the classification performances.

1.3.3. Reflective Action Selection on Multimodal Dialogue

In Section 1.2.3, we explained why a dialogue system installed in a physical robot has to integrate user utterances and multimodal information to take reflective actions. No current multimodal corpus consists of reflective actions. We constructed

a dataset composed of a robot's reflective actions that correspond to user utterances and images. This approach assumes a robot's first-person viewpoint for gathering observations from its environment and user situations in implementing a system for selecting a robot's reflective actions, based on multimodal understanding results. We adopted a crowdsourcing method in Section 1.3.2 to collect from humans the situations used as antecedent observations for pre-defined robot actions. Then we recorded these collected situations as multimodal data.

The dataset size is much smaller than typical text-based corpora [16, 48] because collecting a multimodal dataset is expensive. We annotated the recorded situations in the multimodal data with descriptions to extract effective features for reflective action selection, even from the limited available data, that represent the user's surrounding situation (event). We developed baseline reflective-action selection systems using the annotated features. Experimental results show that we significantly improved the selection accuracy of the reflective actions by applying the descriptive features obtained from the images, even if only a small dataset is available as training data. We confirmed that adding such descriptive features to the pre-trained models widely used in recent studies is an effective way to improve selection accuracy, even when these features are automatically recognized. This result demonstrates the importance of designing appropriate recognition models for the surrounding situations among which a robot takes reflective actions.

1.4. Contributions of Dissertation

Existing dialogue systems cannot generate reflective responses because they are trained with a large corpora of human dialogues and focus on taking actions requested by users. This dissertation's contributions are summarized below from three perspectives corresponding to Sections 1.2 and 1.3.

- We defined the reflective responses of a non-task-oriented dialogue system as responses with high dialogue continuity, in which a user wants to continue their dialogue with the system. To generate responses with high dialogue continuity, we proposed methods that re-rank the response candidates created by a response generation model based on event causality relations and

the coherency of PA structures (events) between user utterances and system responses. We conducted various analyses of our experimental results showing that our proposed method can select responses with high dialogue continuity (but low response coherency) and clarified the relationship between the coherency and dialogue continuity of system responses.

- We defined the reflective actions of a task-oriented dialogue system as actions that satisfy a user's potential requests even when they has not explicitly verbalized them. For creating such a task-oriented dialogue system with reflective actions, we proposed a method to collect a corpus consisting of reflective actions, which is even difficult for humans. We applied PU learning for training the action selection model to solve the problem that the collected corpus is an incompletely labeled dataset. We improved the selection accuracy of the reflective actions by incorporating event causality knowledge into the PU learning.

- We hypothesized that dialogue systems should utilize user's events obtained from images to realize a dialogue system that takes reflective actions on multimodal dialogues. Before testing this hypothesis, we proved that our method, which collects a corpus of reflective actions that is effective in text-based dialogues, can be applied to a corpus collection for multimodal dialogues. In addition, we assigned descriptive labels (events) that outline the situations surrounding a user to effectively utilize the collected multimodal dataset. We proved that multimodal information is effective for selecting reflective actions by training a pre-training model with the constructed multimodal dataset.

Figure 1.1 shows the contributions of this dissertation as steps with which to develop dialogue systems that generate reflective responses. Such dialogue systems are categorized into three steps based on their capabilities. To focus on the research question–whether reflective responses and actions can be generated by dialogue systems–this dissertation concentrates on the generation of a single reflective response or action to a user utterance. In addition, each study focuses on response generation on non-task-oriented or task-oriented dialogues to clarify the problem settings. This is the first step to develop dialogue systems that

generate reflective responses. Here the dialogue systems of each study focus on different tasks. However, if a system can determine which response is the most reflective to a given user utterance, these dialogue systems can be integrated and implemented as a single dialogue system. This is the second step for developing dialogue systems that generate reflective responses. Actual dialogue systems are generally required to manage multi-turn dialogues [16, 110]. Thus, actual dialogue systems that generate reflective responses also need to generate reflective responses based on multi-turn dialogue contexts that consist of user utterances and system responses. This is the third step to develop dialogue systems that generate reflective responses. This dissertation aims to achieve the first step by developing dialogue systems that generate reflective responses. Although the dialogue systems of each study are independent, they will be integrated as the core of a multi-turn dialogue system that supports users on non-task-oriented and task-oriented dialogues in the future.

1.5. Outline of Dissertation

This dissertation is organized as follows. Chap. 2 introduces the neural networks and algorithms that constitute its proposed methods. Chap. 3 introduces the modules of its dialogue systems and describes the details of the modules studied by this dissertation. Chap. 4 describes the re-ranking methods based on the coherency of PA structures, which are proposed to generate reflective responses with high dialogue continuity for a non-task-oriented dialogue system, and discusses the relationship between coherency and dialogue continuity from various perspectives. Chap. 5 describes a corpus collection method for developing a task-oriented dialogue system that selects reflective actions, a training method for an action selection model based on the characteristics of the collected corpus, and detailed analyses of the experimental results of the proposed training method. Chap. 6 describes a method that extends a text-based dialogue corpus with reflective actions to the multimodal corpus and analyzes whether information that describes the situation surrounding a user is useful to select reflective actions based on various comparative experiments. Chap. 7 concludes this dissertation and describes future directions.

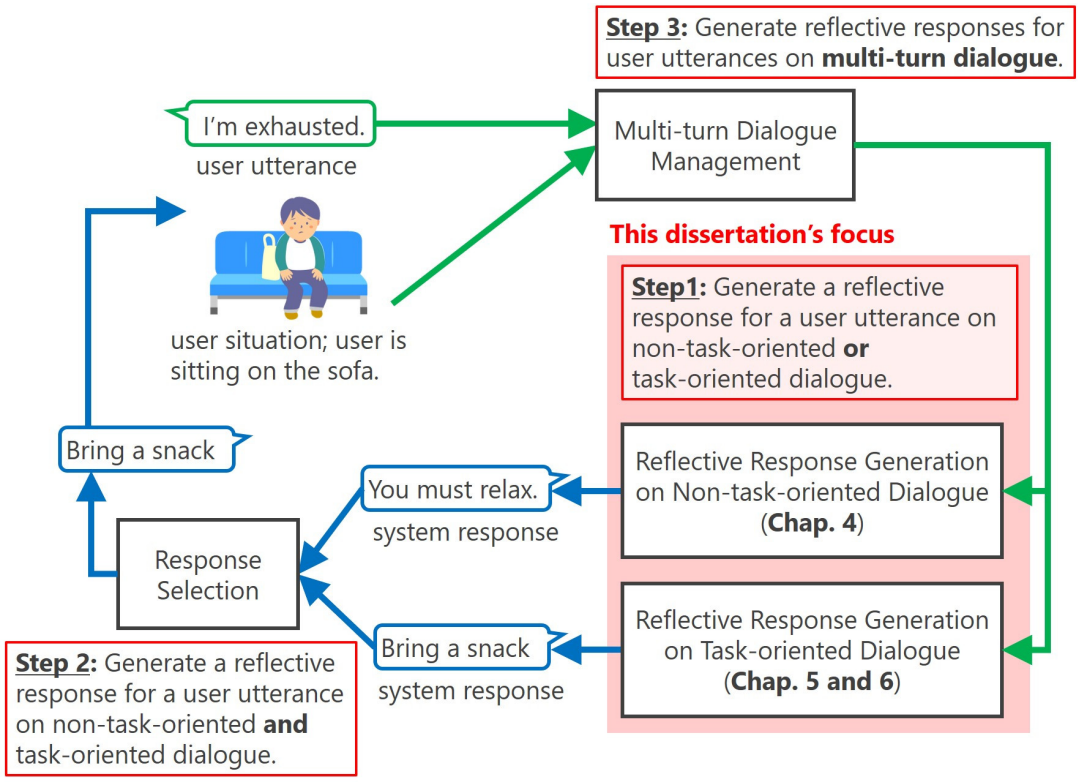


Figure 1.1.: Steps to develop a dialogue system that generates reflective responses

2. Fundamental Technologies

This chapter describes neural networks and algorithms that constitute the proposed methods of this dissertation. Input and output of the described models are texts because all of the proposed systems of this dissertation assume that the input is text.

2.1. Multi-layer Perceptron (MLP)

MLP is a neural network that applies an arbitrary number of linear and nonlinear transformations to an input vector. Figure 2.1 shows the overview of MLP. Arrows between nodes represent a single linear or nonlinear transformation. In Figure 2.1, the output vector $\mathrm{y} = [y_1, y_2]^{\mathrm{T}}$ is obtained by applying linear and nonlinear transformations to the input vector $\mathrm{x} = [x_1, x_2, x_3]^{\mathrm{T}}$ two times. In general, tanh function, ReLU function [38] and sigmoid function are used for nonlinear transformation. All neural networks used in this dissertation, including MLP, are trained using backpropagation method [89] for loss functions.

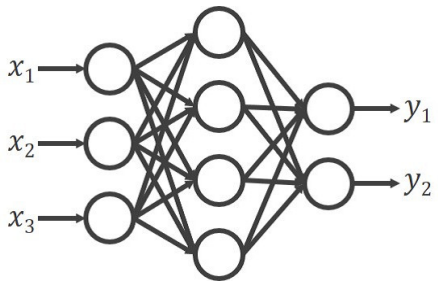


Figure 2.1.: Multi-layer Perceptron (MLP)

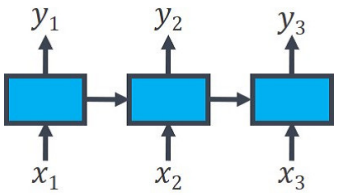


Figure 2.2.: Recurrent Neural Network (RNN)

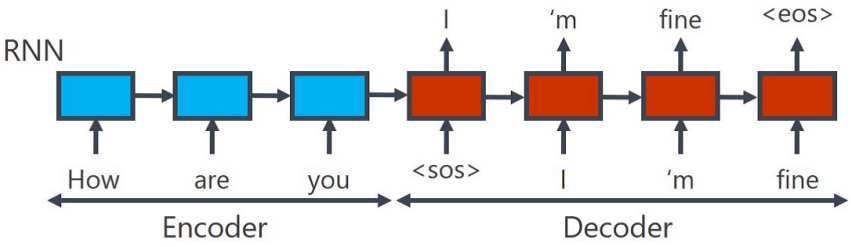


Figure 2.3.: Encoder-Decoder

2.2. Recurrent Neural Network (RNN)

RNN is a neural network to deal with sequential data such as text or speech. As shown in Figure 2.2, RNN predicts its output based on the current and past input. RNN has improved models to efficiently process inputs such as Long Short-Term Memory (LSTM) [43] or Gated Recurrent Unit (GRU) [25, 26].

2.3. Encoder-Decoder

Encoder-Decoder [96] consists of an RNN called Encoder that converts an input sequence into a hidden vector, and an RNN called Decoder that generates an output sequence from the hidden vector passed from Encoder. Figure 2.3 shows the overview of Encoder-Decoder. The minimum unit of the input and output sequence is called token. When generating the output sequence, Decoder receives the token generated in the previous step.

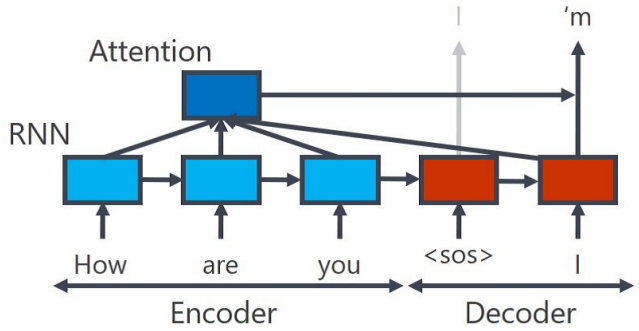


Figure 2.4.: Attention Mechanism

2.4. Attention Mechanism

Usual Encoder-Decoder has the problem that Decoder receives only the hidden vector generated by Encoder in the final step, making it difficult to take into account the information in the beginning part of the input sequence. Attention Mechanism [8, 67] was proposed to solve this problem. As shown in Figure 2.4, Attention Mechanism refers to the information at each step of Encoder based on the current hidden vector of Decoder to generate the output sequence. Additive Attention [8] and Dot-Product Attention [67] are mainly used for this reference process.

2.5. Hierarchical Recurrent Encoder-Decoder (HRED)

HRED [91, 95] is a neural network for generating responses of dialogue systems that deal with dialogue contexts consisting of multiple utterances. As shown in Figure 2.5, HRED hierarchically encodes the dialogue context. HRED first converts each utterance of the dialog context into a hidden vector using Utterance Encoder, and then obtains the hidden vector that represents the entire dialog context by processing utterance hidden vectors. Based on the context hidden vector, Decoder generates the system response.

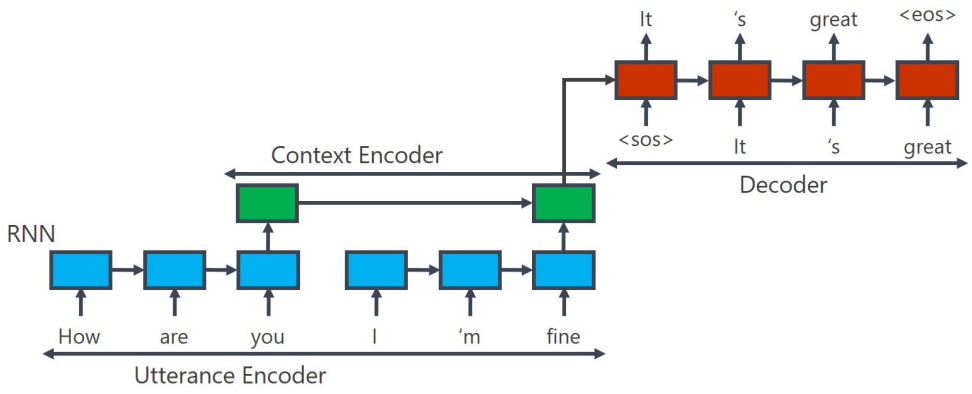


Figure 2.5.: Hierarchical Recurrent Encoder-Decoder (HRED)

2.6. Beam Search

Beam Search is a search algorithm. Figure 2.6 shows the overview of Beam Search. The values represent probabilities of transition from one node to another. At each step of the search, Beam Search preserves the paths to the N nodes with the highest probabilities of transition from the initial node, and discards other paths in the next step of the search. Thus, Beam Search is equal to Greedy Search when $N = 1$, and it is equal to Breadth-First Search when $N = \infty$. Greedy Search is memory efficient, but does not guarantee that the resulting path is the global optimal solution. In contrast, Breadth-First Search always gets the global optimal solution, but requires a large amount of memory if the path length is long. Beam search is a search algorithm that is more memory efficient than Breadth-First Search and is expected to get paths closer to the global optimal solution than Greedy Search.

2.6.1. Transformer, BERT, and RoBERTa

Transformer [103] is a neural network that utilizes Attention Mechanism instead of RNN. Transformer has new Attention Mechanisms called Self-Attention which Encoder and Decoder focus on their own hidden vectors and Multi-Head Attention in which attention vectors are divided into small parts, resulting in high per-

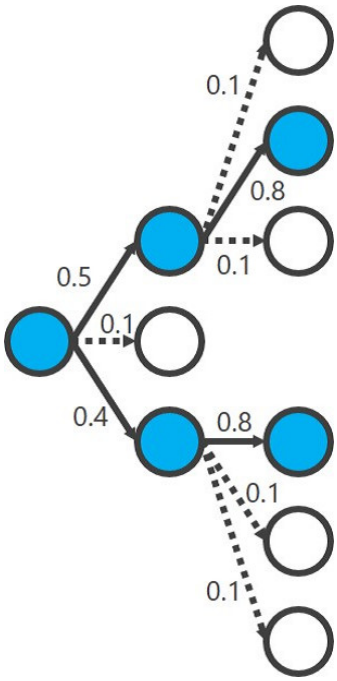


Figure 2.6.: Beam Search ($N = 2$)

formances in language generation tasks such as machine translation. Figure 2.7 shows the overviews of Self-Attention and Multi-Head Attention. BERT [28] and RoBERTa [63] are Encoder parts of Transformer that were trained on pre-training tasks such as masked word prediction. They are used to convert sentences into distributed representations.

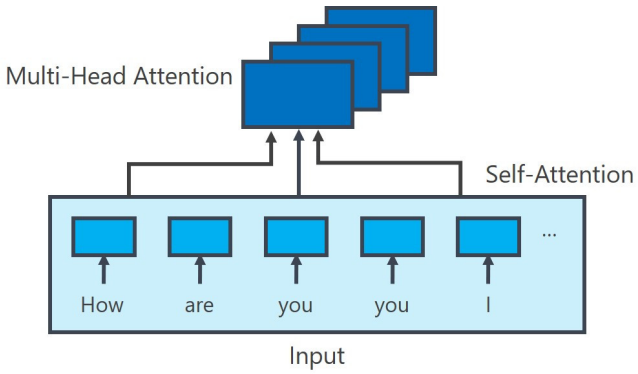


Figure 2.7.: Self-Attention and Multi-Head Attention

3. Dialogue System Architecture

This chapter describes the modules of the dialogue system in this dissertation. Figure 3.1 overviews the dialogue system architecture. Note that this figure illustrates the position of the methods proposed in this dissertation based on a general dialogue system architecture, whereas Figure 1.1 illustrates the steps to develop dialogue systems that generate reflective responses. If the system is a spoken dialogue system, the system needs to transcribe the user utterance into text with an Automatic Speech Recognition (ASR) module [76]. This module is not necessary if the system works on a chat tool. If the system deals with multimodal information such as images, Multimodal Recognition (MR) modules such as image recognition [88, 97] are necessary to extract the information of the surrounding situation. The transcribed user utterance and the recognized situation are analyzed by a Situation Understanding (SU) module. Based on the analysis by the SU module, the system defines its next action using a Dialogue Management (DM) module. The actions defined by the DM module are category actions such as *suggest a cafe* or *bring a snack*. Based on the action defined by the DM module, the system generates a specific response or action using a Response Generation (RG) module. For example, the system searches for the specific cafe or generates the response text. Finally, if the system is a spoken dialogue system, the generated response is converted into speech using a Text-to-Speech (TTS) module. If the system is mounted on a robot, the action is converted into the actual manipulator operation using a Robot Operation (RO) module. This dissertation studies the three modules of the dialogue system: SU, DM, and RG. Each module is described in detail below.

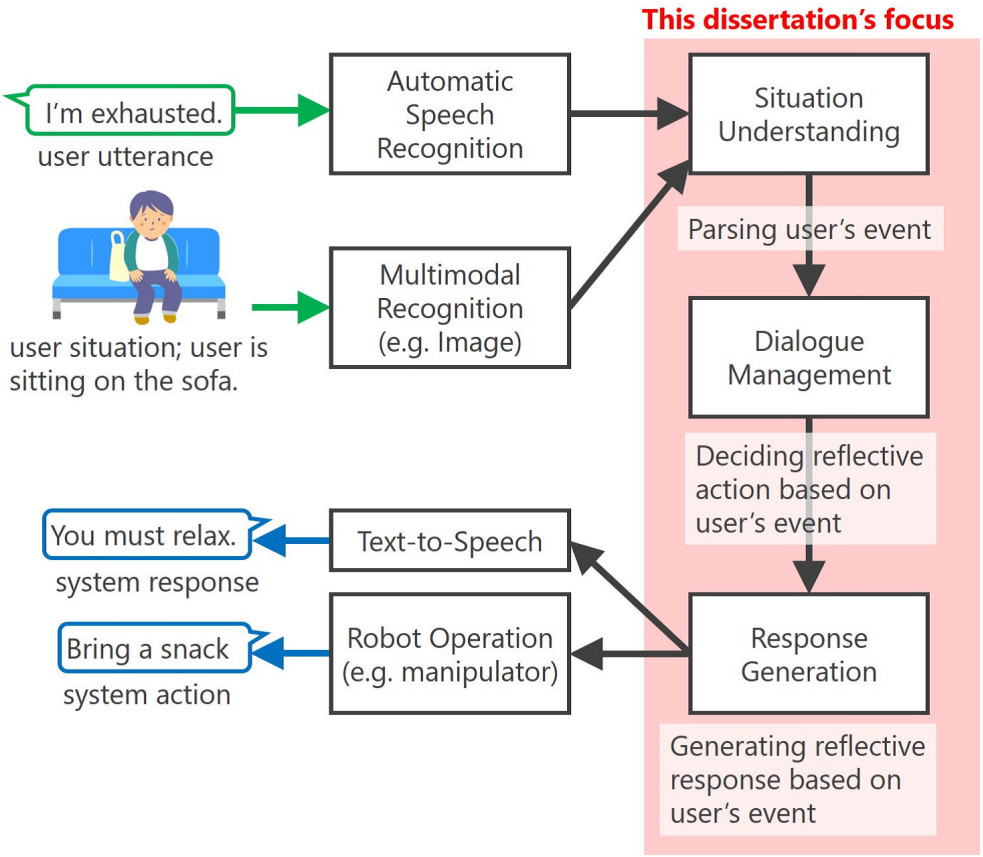


Figure 3.1.: Dialogue system architecture on this dissertation

3.1. Situation Understanding

This module analyzes the transcribed user utterance and the recognized situation based on the purpose of the dialogue system. For example, task-oriented dialogue systems such as car navigation or restaurant reservation [14, 16, 31, 33] extract values to fulfill predefined slots that are required to complete the tasks, such as user's destinations or genres of the restaurant. This parsing process is effective when the user utterance includes the explicit values to fulfill the slots. However, in dialogues of this dissertation, this parsing process is not effective because the user does not verbalize the explicit values especially on the task-oriented dialogues.

For non-task-oriented dialogue systems, defining slots is not effective because the system needs to talk about various topics. One possible solution is categorizing the user utterance into dialogue act [111] such as a question or back-channeling. This parsing process is effective in maintaining coherency at the dialogue act level such as responding with Yes/No to the user's confirmation. However, more detailed parsing is necessary to generate reflective responses or actions to the content of the user utterance.

In this dissertation, we use PA structure analysis as the parsing process for the user utterance and extract PA structures as the user's events. PA structure is a unit of natural language processing centered on predicates, which are generally treated as events included in texts. In this definition, an event has one predicate and multiple arguments such as subject or object [22, 23]. PA structures can be automatically extracted with PA structure analyzer [50, 90] which was trained with a statistical method as shown in Figure 3.2. Based on this definition, we treat PA structures that are extracted from text-based user utterances as the events included in the user utterances. PA structure analysis does not need predefined slots because it is a domain-independent process to extract dependency structures of predicates and arguments included in texts. Moreover, PA structures are effective features for generating a reflective response or action based on the user utterance because it extracts more detailed contents of the user utterance compared to the dialogue act.

For multimodal dialogue systems such as the system in Chap. 6, the system needs to analyze the events of the situation surrounding the user using the SU module. In this dissertation, the events of the situation surrounding the user

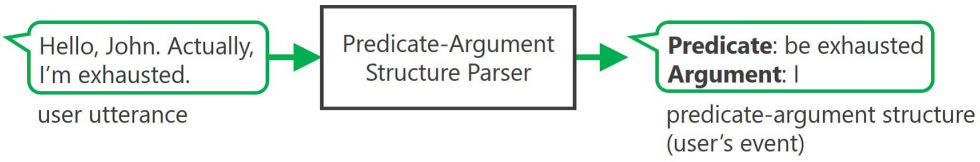


Figure 3.2.: Parsing predicate-argument structure from user utterance

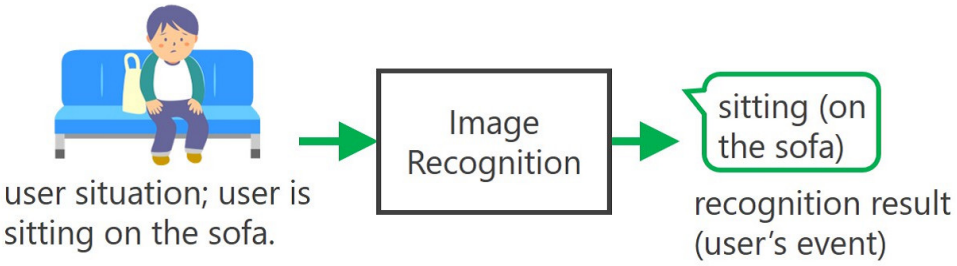


Figure 3.3.: User's event recognition with image recognition

represent the texts (description) that describe the user and its surrounding situation such as "the user in the kitchen has a glass." or "the user is sitting on the sofa." Although the user does not verbalize these events, the system can recognize them with an image recognition model [88, 97] as shown in Figure 3.3. The system can extract PA structures using the PA structure analyzer if the texts are sentences. The system in this dissertation does not use the PA structure analyzer to recognize the descriptive events because they are words such as "glass" or "sitting."

3.2. Dialogue Management

This module defines the system's next action based on the user utterance and the surrounding situation which were analyzed by the SU module. This action decision has multiple granularities. For a dialogue robot equipped with a mobile manipulator, it has to decide whether to talk with the user or operate something with the manipulator. When the robot chooses to talk with the user, it has to

decide the dialogue act for the response generation because the system response is categorized into a dialogue act such as a question or back-channeling [17, 18]. When the robot chooses the manipulator operation, it has to decide the rough action category such as bringing something or putting something away. The DM module decides the system action based on the dialogue contexts if the system talks with the user over multiple turns.

The dialogue management based on action decision is effective for task-oriented dialogue systems [31, 33, 110, 113] because system actions can be categorized as user utterances are categorized. For example, if the user utterance is a question, the system can decide the rough action such as providing information to guarantee that the system selected the appropriate action at least on the rough category level. The system needs to convert the action category into the specific response or action using a rule-based model or a response generation model [46] of the RG module because the action category of the DM module is a rough category.

The systems of Chap. 5 and Chap. 6 are categorized into the DM module because they select reflective action categories on the task-oriented dialogues. When the user says "I'm exhausted." as shown in Figure 3.4, the systems define reflective action categories, such as suggesting going to a cafe if the user is exhausted while sightseeing, or bringing a snack if the user is sitting on the sofa in a living room. We developed architectures that do not only learn the correspondence between user utterances and system actions with End-to-End learning, but also utilize user events, such as PA structures included in the user utterances or the situation surrounding the user, to select the reflective actions. Specifically, we tested the effectiveness of a model that learns action selection with event causality knowledge between the user utterances and the systems responses, and a model that integrates the user utterances and the events surrounding the user.

3.3. Response Generation

This module traditionally converts rough action categories defined by the DM module into specific system responses or actions. Recent End-to-End dialogue systems using neural networks [58, 104] integrate the SU, DM and RG modules and directly generate system responses with statistical methods as shown in Fig-

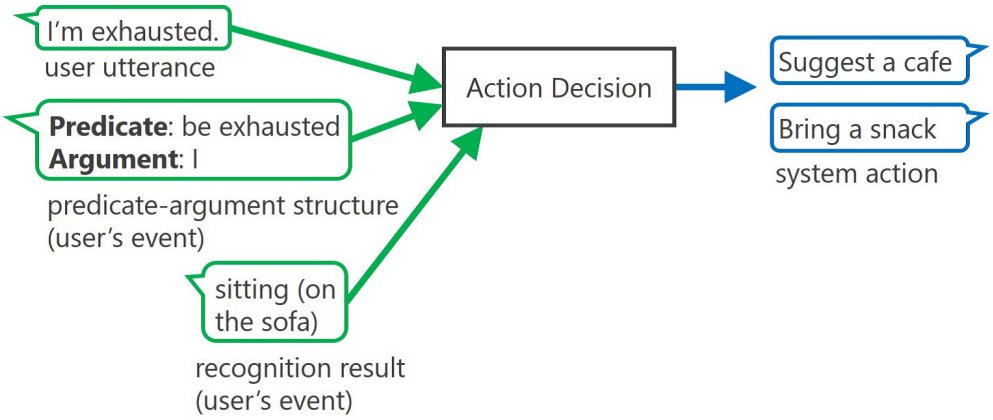


Figure 3.4.: Action decision based on user's event

ure 3.5. These architectures are also effective for the task-oriented systems that assume ambiguous user requests studied in Chap. 5 and Chap. 6 because they do not require the detailed definition of slots for the SU and DM modules.

The End-to-End architecture is adopted by existing task-oriented dialogue systems that treat user requests which are not always clear [48,59,75]. They assume that the system can still make recommendations even if the user lacks a specific request, in particular, dialogue domains such as movies or music. On the other hand, the systems of Chap. 5 and Chap. 6 focus on conversational utterances or monologues, which can trigger reflective actions from the system. Note that the systems of Chap. 5 and Chap. 6 are categorized into the DM module because they focus on action decisions rather than response generation.

Multimodal dialogue systems and robots generate responses and actions based not only on the user utterance but also on the image or other modals [7,37,54,100]. The systems have to integrate linguistic and visual information because they cannot generate appropriate responses based solely on the user utterances in the tasks. The system of Chap. 6 integrates the user utterances and the images that represent the surrounding situations to select the reflective actions. Note that the system is categorized into the DM module because it selects the action categories.

The response re-ranking of Chap. 4 is postprocessing for responses generated by a response generation model. Although postprocessing such as re-ranking has

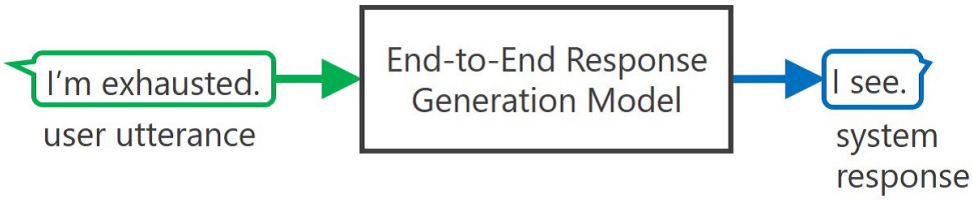


Figure 3.5.: End-to-end response generation

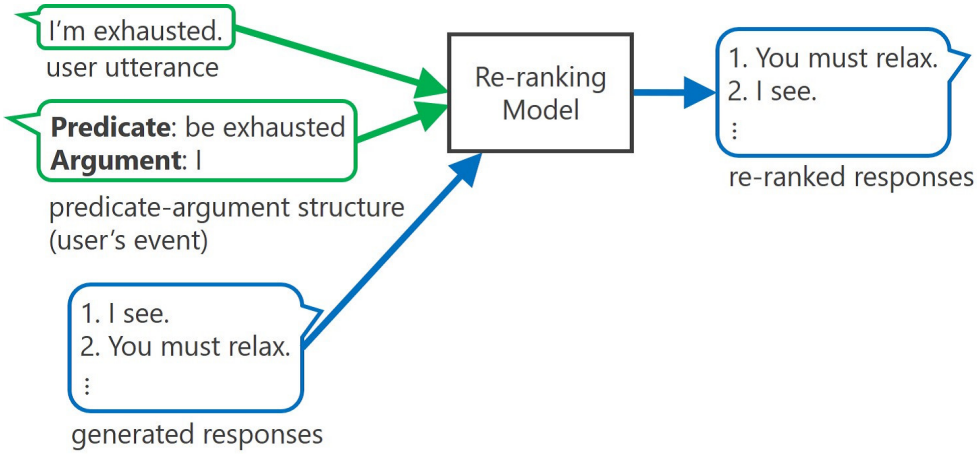


Figure 3.6.: Response re-ranking based on user's event

the disadvantage of increasing the complexity of the system architecture, it has the advantage that we do not need to collect a corpus according to the re-ranking criteria. For example, the response generation based on the coherency of PA structures requires a large corpus if it learns coherent response generation with the End-to-End architecture. It is expensive to collect a large corpus for non-task-oriented dialogues that meet certain criteria. In contrast, the response re-raking does not require a new corpus because it re-ranks the responses generated by a response generation model which was trained with an existing corpus. Therefore, this dissertation aims to generate reflective responses with high dialogue continuity for a non-task-oriented dialogue system by re-ranking responses based on the coherency of events (PA structures) analyzed using the SU module, as shown in Figure 3.6.

4. Non-task-oriented Response Re-ranking Based on Coherency of Sequential Events

This chapter investigates the non-task-oriented dialogue system that generates reflective responses with high dialogue continuity. We propose methods that re-rank response candidates using event causality knowledge or Coherence Model based on coherency between PA structures (events) of user utterances and system responses. We generalize the event causality knowledge with distributed representation to establish the robust matching of the causality knowledge. We evaluate the performances of the proposed methods on dialogue continuity using automatic and human evaluations. In addition, we conduct various analyses for the relation of dialogue continuity and coherency of responses.

4.1. Response Re-ranking Based on Sequential Events

Figure 4.1 shows an overview of the proposed re-raking method based on the coherency of events. The re-raking method consists of two parts. First, N-best response candidates are generated from an NCM given a dialogue context (Figure 4.1 ①; Section 4.1.1). Next, response candidates are re-ranked based on the coherency of events (Figure 4.1 ②). We proposed two different methods for this re-ranking. The first re-ranking method uses event causality pairs [93, 94] that are statistically extracted as the external knowledge for coherency of events

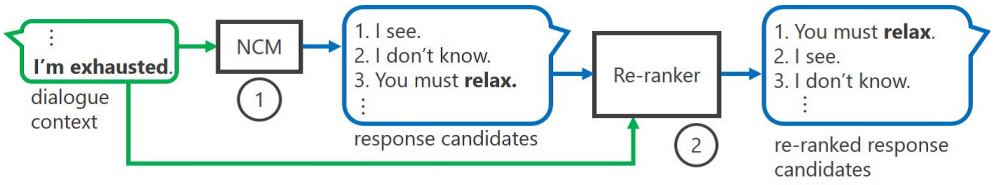


Figure 4.1.: Neural conversational model+re-ranking; Selects the response based on knowledge that *I be exhausted* and *relax* have a causality relation.

(Section 4.1.2). The second re-ranking method estimate coherency of events and whole dialogues using Coherence Model (Section 4.1.3).

4.1.1. Neural Conversational Model (NCM)

NCM learns a mapping between input and output word sequences by using recurrent neural networks (RNNs). NCMs can generate N-best response candidates by using beam search or sampling [68]. We used N-best response candidates using beam search for the re-ranking.

4.1.2. Re-ranking Utilizing Event Causality Pairs

This study deals with two different definitions of coherency. The first definition is coherency between events included in dialogue contexts and responses. Based on this definition, for example, the coherency is high when an event *be stressed out* is included in a dialogue context and an event *relieve stress* is included in a response. The second definition is coherency of responses to whole dialogue contexts. Based on this definition, for example, the coherency is high when "I am stressed out." is the dialogue context and "You are better to relieve stress" is the response.

Our methods re-rank response candidates based on the hypothesis that if a response includes a coherent event to the dialogue context, then the response has high coherency and high dialogue continuity. Based on the coherency of sequential events in a dialogue, the response can be regarded as coherent if it has any causality relation with the dialogue context. We propose the method utilizes event causality relations based on PA structures. First, the method extracts

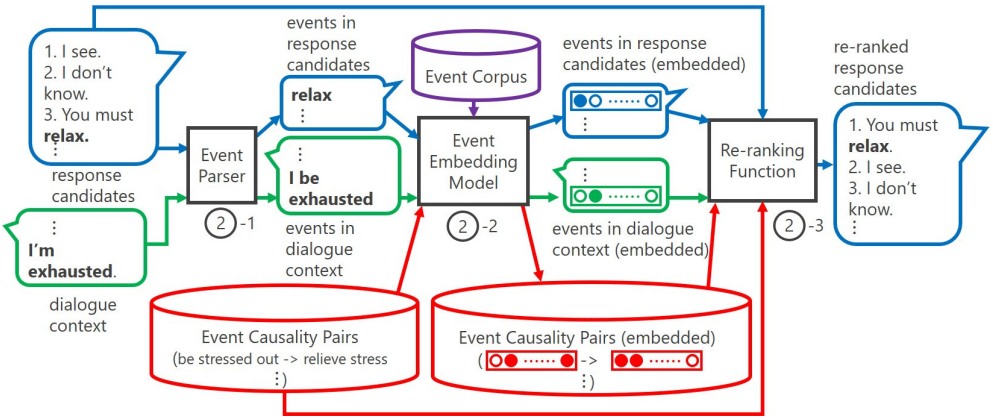


Figure 4.2.: Re-ranking using the event causality pairs; The response is selected by the re-ranking because it has the event causality relation (*I be exhausted* → *relax*) with the dialogue context.

Table 4.1.: Example of event causality relations

predicate 1	argument 1	predicate 2	argument 2	$lift$
be stressed out	I	relieve	stress	10.02

events (PA structures) by an event parser from both the dialogue context and the response candidates (Figure 4.2 ②-1). We used KNP* [50,90] as the event parser. Next, the extracted events are converted into distributed event representations by an event embedding model (Figure 4.2 ②-2; Section 4.1.2). RFTM is used for the embedding. Finally, response candidates are re-ranked (Figure 4.2 ④; Section 4.1.2 and 4.1.2).

Causality Pairs

The proposed method uses event causality pairs. Events in a pair, which have cause-effect relations, are extracted from a large-scale corpus on the basis of co-occurring statistics and case frames [93, 94]. 420,000 entries are extracted from 1.6 billion texts: each entry consists of information denoted in Table 4.1.

*http://nlp.ist.i.kyoto-u.ac.jp/?KNP

Predicate 1 and argument 1 are components of a cause event, and predicate 2 and argument 2 are components of an effect event. Each event consists of a predicate and arguments. The predicate is required, and the argument is optional. We used arguments that have the following roles: nominative, accusative, dative, instrumental, and locative cases. $lift$ is the mutual information score between two events [93], which indicates the strength of the causality relation. $p(\mathrm{e})$ is the occurrence probability of events in Web texts, and $p(\mathrm{e}_h, \mathrm{e}_r)$ is the co-occurrence probability of two events. e_h is an event of a dialogue context, and e_r is an event of a response candidate.

$$lift(\mathrm{e}_h, \mathrm{e}_r) = \frac{p(\mathrm{e}_h, \mathrm{e}_r)}{p(\mathrm{e}_h)p(\mathrm{e}_r)}. \tag{4.1}$$

Using $lift$, we propose a score for re-ranking as,

$$score(\mathrm{h}, \mathrm{r}) = \max_{<\mathrm{e}_h, \mathrm{e}_r>} \frac{\log_2 \mathrm{P}(\mathrm{r} \mid \mathrm{h})}{\Big(\log_2 lift(\mathrm{e}_h, \mathrm{e}_r)\Big)^{\lambda}}. \tag{4.2}$$

$\mathrm{P}(\mathrm{r} \mid \mathrm{h})$ is the posterior probability of the response candidate r provided by NCM for the dialogue context h. λ is a hyperparameter to decide the weight of event causality relations. $lift(\mathrm{e}_h, \mathrm{e}_r)$ is the $lift$ score between an event e_h in the dialogue context, and an event e_r in the response candidate, which is equal to 2 if the pair does not appear in the extracted event causality pairs. Note that $lift(\mathrm{e}_h, \mathrm{e}_r)$ is log-scaled (Pointwise Mutual Information between the events) because it has a wide range of values ($10 < lift(\mathrm{e}_h, \mathrm{e}_r) < 10,000$). In the case where more than one event causality relations are recognized between the dialogue context and the response candidate, the score of the candidate is determined by the relation with the highest $lift(\mathrm{e}_h, \mathrm{e}_r)$. Since the value range of $\log_2 \mathrm{P}(\mathrm{r} \mid \mathrm{h})$ is $(-\infty, 0]$, the larger the $lift$ value, the larger the re-ranking score. We call this model *Re-ranking (Pairs)*.

Distributed Event Representation Based on Role Factored Tensor Model (RFTM)

It is difficult to determine all event causality relations in a dialogue by using only the pairs observed in an actual corpus. Therefore, we introduce a distributed event representation to improve the robustness of matching events in dialogue

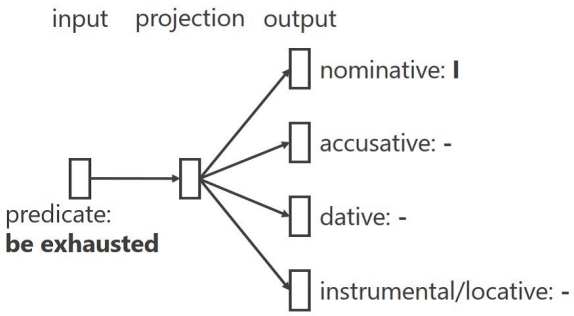


Figure 4.3.: Predicate embedding

with events in the event causality pairs. We define an event with a single predicate or a pair of a predicate and arguments. Argument a of an event is embedded into a vector as v_a by using GloVe [86]. Predicate p of an event is embedded into a vector as v_p by using predicate embedding which is based on case-unit Skip-gram [72–74]. Figure 4.3 shows the model architecture of predicate embedding. The model learns predicate vector representations which are good at predicting its arguments. To get an event embedding for the pair of v_p and v_a, we propose to use RFTM, which was proposed by [106]. RFTM embeds a predicate and its arguments into vector ve as,

$$\mathrm{ve} \quad = \quad \sum_a W_a T(\mathrm{v}_p, \mathrm{v}_a). \tag{4.3}$$

The relation of a predicate and its arguments is computed using a 3D tensor T and matrices W_a. If the event has no arguments, ve is substituted by v_p. RFTM is trained to predict an event sequence; thus, it can represent the meaning of the event in a particular context. As with the distributional hypothesis of words, it assumes that events appearing in similar contexts have similar meanings. Thus, events with similar contexts are embedded in close locations in the distributed representation space.

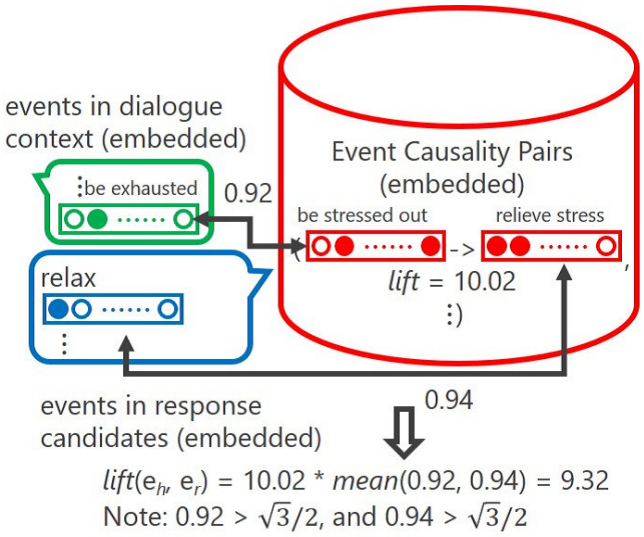


Figure 4.4.: Matching of event causality relations; The $lift$ of the causality (*be exhausted* $\rightarrow$ *relax*) is calculated based on the $lift$ of the causality (*be stressed out* $\rightarrow$ *relieve stress*) that has the highest cosine similarity.

Causality Relation Matching Based on Distributed Event Representation

Figure 4.4 illustrates the process of matching events based on distributed event representation. Given an event pair from a response candidate and a dialogue context, the proposed method finds an event causality pair that has the highest cosine similarity from the pool. $lift_{emb}$ score, strength of the event causality relation, is extended as,

$$lift_{emb}(\mathrm{e}_h, \mathrm{e}_r) = lift(\mathrm{e}_c, \mathrm{e}_e) * mean\Big(sim(\mathrm{ve}_h, \mathrm{ve}_c), sim(\mathrm{ve}_r, \mathrm{ve}_e)\Big). \quad (4.4)$$

e_h is an event in the dialogue context, e_r is an event in the response candidate. e_c and e_e are a cause and an effect event of an event causality pair, respectively. ve_h, ve_r, ve_c and ve_e are vectors of each event. sim is the cosine similarity between the vectors. $mean$ is a process to calculate the mean value. We also calculate the score for the case in which the cause and effect events are exchanged to deal with the inverse case. The method has a possibility to overgeneralize events

such as dealing with *catch a cold* and *wake up* as the same event. To avoid this problem, both *sim* values have a threshold to prevent overgeneralization. If the *sim* value of an event pair is lower than the threshold, the event pair is not used for re-raking. The event pair $\mathrm{e}_c, \mathrm{e}_e$ used as an alternative to $\mathrm{e}_h, \mathrm{e}_r$ is selected to have the highest $mean\Big(sim(\mathrm{ve}_h, \mathrm{ve}_c), sim(\mathrm{ve}_r, \mathrm{ve}_e)\Big)$. Replacing $lift(\mathrm{e}_h, \mathrm{e}_r)$ in Eq. (4.2) with $lift_{emb}(\mathrm{e}_h, \mathrm{e}_r)$, the score using distributed event representation is defined as,

$$score(\mathrm{h}, \mathrm{r}) \quad = \quad \max_{<\mathrm{e}_h, \mathrm{e}_r>} \frac{\log_2 \mathrm{P}(\mathrm{r} \mid \mathrm{h})}{\Big(\log_2 lift_{emb}(\mathrm{e}_h, \mathrm{e}_r)\Big)^{\lambda}}. \tag{4.5}$$

As with *Re-ranking (Pairs)*, in the case where more than one event causality relations are recognized between the dialogue context and the response candidate, the score of the candidate is determined by the relation with the highest $lift_{emb}(\mathrm{e}_h, \mathrm{e}_r)$. We call this model *Re-ranking (RFTM)*.

4.1.3. Re-ranking Utilizing Coherence Model

Although event causality is important to estimate coherency, we expect that various other factors, such as content words and functional words, contribute to response coherency. For example, when a dialogue context is "I am stressed out." and a response is "I want to relieve stress," the response coherency is not high although the event in the response is coherent to the dialogue context. In addition, the event causality pairs used in this study were extracted using a statistical method, and not all of them were established as event causality relations. Thus, we propose a re-ranking method that evaluates the coherency of event pairs and whole responses by utilizing Coherence Model. As with the re-ranking methods using event causality pairs, this re-ranking method extracts events in a dialogue using the event parser (Figure 4.5 ②-1). Next, given the extracted events, the dialogue context, and the response candidates, Coherence Model estimates coherence scores of the response candidates to the dialogue context (Figure 4.5 ②-2; Section 4.1.3). Finally, the response candidates are re-ranked based on the coherence scores (Figure 4.5 ②-3; Section 4.1.3).

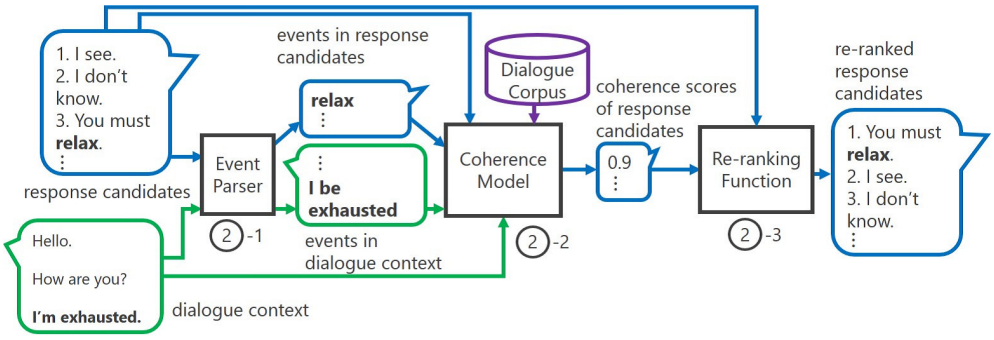


Figure 4.5.: Re-ranking using Coherence Model; The response is selected based on coherency of the events (*I be exhausted* and *relax*) and the whole dialogue.

Coherence Estimation of Dialogue Based on Coherence Model

The re-raking method utilizes Coherence Model [109] to estimate the coherency of response candidates to dialogue contexts. The model detects whether a subsequent sentence to the preceding text is a continuous sentence or a randomly substituted sentence. In this study, Coherence Model is used to estimate the coherency of the response candidates to the dialogue context. Figure 4.6 shows positive/negative examples used for training the model. The positive example is the pair of the dialogue context and the response that have the causality relation. As with *Re-ranking (Pairs)*, the event causality pairs [93, 94] are used for matching of causality relations. The negative example is the reversed order of the positive example, i.e., the response is replaced with an utterance in the dialogue context that has a causality relation with the response. We expect that the method estimates a high coherence score only for response candidates for which both the included events and the overall meaning are coherent to the dialogue context. Figure 4.7 overviews the model architecture. This model converts the dialogue context h and the response candidate r into distributed representations v_h and v_r using BERT [28]. In addition, the event pair e_h and e_r, which is extracted from the dialogue context and the response candidate, is converted into distributed representations ve_h and ve_r using RFTM. Only event pairs with cosine similarity greater than a threshold are used for re-ranking. We expect that

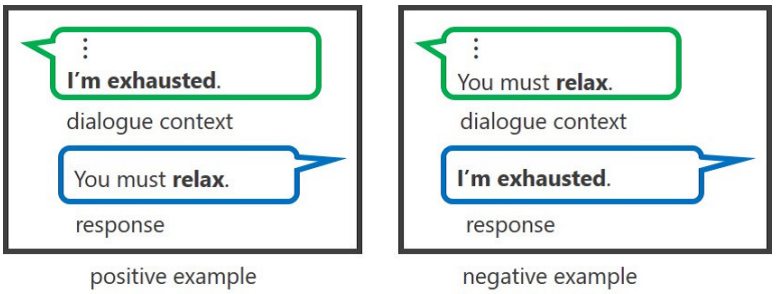


Figure 4.6.: Positive/negative examples used to train Coherence Model; Although the event of the negative example response (*I be exhausted*; cause) is coherent to the event of the dialogue context (*relax*; relax), the response itself is not coherent to the dialogue context.

the model weights sequential events because RFTM learns to increase the cosine similarity between sequential events. Based on the distributed representations, the coherence score *coh* of the response candidate is calculated as,

$$coh(\mathrm{v}_h, \mathrm{v}_r, \mathrm{ve}_h, \mathrm{ve}_r) = \sigma\Big(W\mathrm{v} + b\Big). \tag{4.6}$$

$$\begin{aligned}\mathrm{v} = {} & [\mathrm{v}_h; \mathrm{v}_r; \mathrm{v}_h - \mathrm{v}_r; |\mathrm{v}_h - \mathrm{v}_r|; \mathrm{v}_h * \mathrm{v}_r \\ & ; \mathrm{ve}_h; \mathrm{ve}_r; \mathrm{ve}_h - \mathrm{ve}_r; |\mathrm{ve}_h - \mathrm{ve}_r|; \mathrm{ve}_h * \mathrm{ve}_r].\end{aligned} \tag{4.7}$$

σ is a sigmoid function, W, b are a parameter matrix and a parameter bias, respectively. $[;]$ represents concatenation of vectors, $*$ represents element-wise product. In the case where more than one event causality relations are recognized between the dialogue context and the response candidate, the score of the candidate is determined by the relation with the highest cosine similarity. If the similarity scores of all the causality relations are lower than the threshold, the coherence score is regarded as 0. Multi-layer perceptron (MLP) is used to calculate the coherence score as shown in Figure 4.7. Using the positive example ${x_i}^+$ and the negative example ${x_i}^-$ in the training data, the loss function for learning Eq. (4.6) as Margin Ranking Loss is defined as,

$$Loss({x_i}^+, {x_i}^-) = \max\Big(0, -(f({x_i}^+) - f({x_i}^-)) + 0.5\Big). \tag{4.8}$$

$f({x_i}^+), f({x_i}^-)$ are the coherence scores which the model estimates. Eq. (4.8) is 0 when $f({x_i}^+) - f({x_i}^-) \geq 0.5$. We expect that the model estimates high coher-

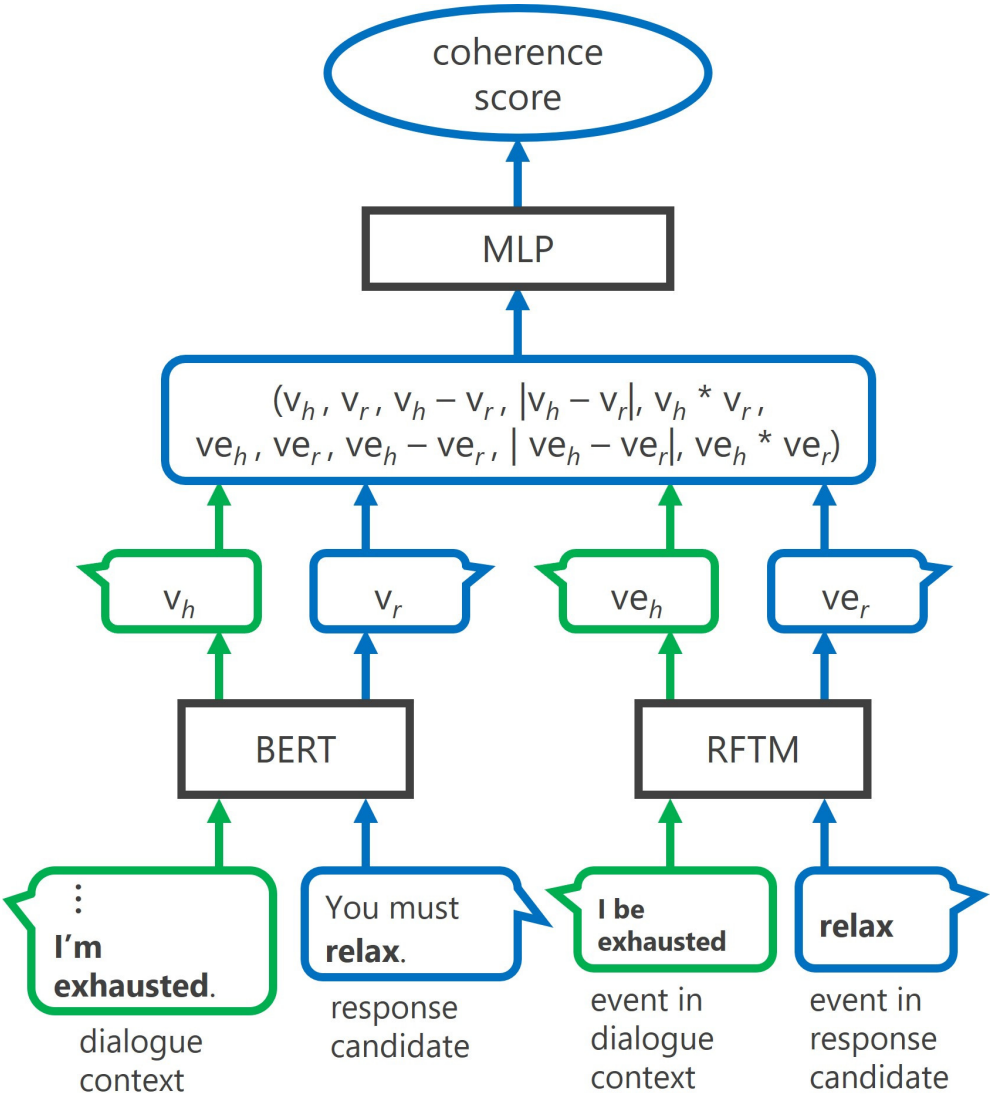


Figure 4.7.: Calculating the coherence score using Coherence Model

ence scores for coherent pairs of dialogue contexts and response candidates and low coherence scores for non-coherent pairs of dialogue contexts and response candidates by training the model with the loss function. If the coherence score ($0 \leq coh \leq 1$) of the response candidate is greater than 0.5, the response candidate is regarded as coherent to the dialogue context. The re-ranking score is calculated as,

$$score(\mathrm{h}, \mathrm{r}) \;=\; \Big(1 - coh(\mathrm{v}_h, \mathrm{v}_r, \mathrm{ve}_h, \mathrm{ve}_r)\Big) \log_2 \mathrm{P}(\mathrm{r} \mid \mathrm{h}). \tag{4.9}$$

As with Eq. (4.2)(4.5), since the value range of $\log_2 \mathrm{P}(\mathrm{r} \mid \mathrm{h})$ is $(-\infty, 0]$, the larger the coherence score, the larger the re-ranking score. We call this model *Re-ranking (Coherence)*.

4.2. Experiments

We evaluate the proposed re-ranking methods for response candidates. We test the hypothesis that the coherency of sequential events improves the coherency of responses to dialogue contexts, resulting in an improvement of dialogue continuity. We conducted automatic and human evaluations to compare responses with and without the re-ranking. In human evaluation, we evaluate the coherency and dialogue continuity of response candidates to dialogue contexts. We used Encoder-Decoder with Attention (EncDec) [8, 67] and Hierarchical Recurrent Encoder-Decoder (HRED) [91, 95] to generate the response candidates. While HRED tries to generate more coherent responses to dialogue context than a simple Encoder-Decoder, the diversity of responses is small due to context constraints. We collected 2,072,893 dialogues from Japanese micro blogs (Twitter) to train and test the response generation models. One utterance in the micro blogs is regarded as one turn, and the number of turns is regarded as a series of utterances. The average dialogue turn was 13.50, and the average utterance length was 22.52 words. We removed emoticons from utterances to reduce vocabulary size and accelerate the training. The dialogue corpus was split into 1,969,626, 51,573, and 51,694 dialogues as training, validation, and test data, respectively. We used the Japanese data from a Wikipedia dump[†] for training GloVe and predicate word

[†]The latest version on November 2nd, 2018.

embeddings of RFTM, and the *Maichichi* newspaper dataset 2017[‡] and the training data of the micro blogs for training RFTM. We used a pre-trained BERT[§] for Coherence Model. The training data of the micro blogs was used to train Coherence Model.

4.2.1. Model Configuration

The hidden unit sizes of GloVe [86], predicate embedding, and RFTM [106] were set to 100. The hidden unit size of BERT of Coherence Model was set to 768, and the number of hidden layers of MLP was set to 1. We used gated recurrent units (GRUs) [25, 26] whose number of layers was 2 and hidden unit size was 256, for the encoder and decoder of the NCMs. The batch size was 100, the dropout probability was 0.1, and the teacher forcing rate was 1.0. We used Adam [52] as the optimizer. The gradient clipping was 50, the learning rate for the encoder and the context RNN of HRED was $1e^{-4}$, and the learning rate for the decoder was $5e^{-4}$. The loss function was inverse token frequency (ITF) loss [77]. We used sentencepiece [53] as the tokenizer, and the vocabulary size was 32,000. The dialogue data has an average dialogue length of 13.50 turns, but the topic shifts between distant utterances, reducing relevance. NCM cannot take too long contexts as input due to RAM size limitation. In this study, the maximum dialogue context for response generation and re-ranking was the past three utterances. These settings were the same in all models. When multiple utterances were input into EncDec, they were combined with a special token </s> indicating the end of a sentence and input as a single utterance. When multiple utterances are input into HRED, each utterance is encoded by Utterance Encoder, and then the entire dialogue context is encoded by Context Encoder [91, 95].

Repetitive suppression [77] was used during generation to prevent the same tokens from being output repeatedly, and length normalization [68] was used to prevent only short responses from being output. λ of Eq. 4.2 and Eq. 4.5 was set to 1.0. We set the threshold, which is for the cosine similarity of the matching of event causality relations based on distributed representation and event pairs of Coherence Model, to 0.9 heuristically.

[‡]http://www.nichigai.co.jp/sales/mainichi/mainichi-data.html
[§]http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT 日本語 Pretrained モデル

Table 4.2.: Diversity of N-best response candidates

	Ave.dist-1	Ave.dist-2
EncDec	**0.55**	**0.69**
HRED	0.46	0.58

4.2.2. Diversity of Response Candidates

We investigated the internal diversity of N-best response candidates generated from each dialogue model. It is expected that the higher diversity is, the more effective re-ranking is. We evaluated diversity on the test data by dist-1, 2 [58]. The beam width was set to 20; it is the same in the following experiments. The result is shown in Table 4.2: Ave.dists are averages of dist computed internal N-best response candidates. The diversity of EncDec is higher than that of HRED.

4.2.3. Automatic Evaluation

Evaluation Metrics

We calculated the ratios of re-ranked response candidates (re-ranked) to estimate how much the re-ranking methods work. We compared the results by referring to BLEU [84], NIST [30], and three vector-based metrics (greedy, average, extrema) [35, 62]. BLEU represents N-gram agreements with the reference. NIST is based on BLEU, but heavily weights less frequent N-grams to focus on content words. greedy, average, and extrema compute the similarity between sentence vectors of a reference and a generated response. The similarity GM in greedy is computed by greedily matching the GloVe word vector ge_w of words in the reference and the generated response, respectively, as,

$$G(s, \hat{s}) = \frac{\sum_{w \in s} \max_{\hat{w} \in \hat{s}} sim(\mathrm{ge}_w, \mathrm{ge}_{\hat{w}})}{|s|} \tag{4.10}$$

$$GM(s, \hat{s}) = \frac{G(s, \hat{s}) + G(\hat{s}, s)}{2}. \tag{4.11}$$

s and $\hat{s}$ are the reference and the generated response, respectively. sim is cosine similarity. The similarity in average is computed based on the cosine similarities

of the sentence vectors of the reference and the generated response. The sentence vector ge_s is computed by averaging the GloVe word vectors as,

$$\mathrm{ge}_s = \frac{\sum_{w \in s} \mathrm{ge}_w}{|s|}. \tag{4.12}$$

As with average, the similarity in extrema is computed based on the cosine similarities of the sentence vectors. The sentence vector is computed by taking the extreme values of the GloVe word vectors in each dimension d as,

$$\mathrm{ge}_{sd} = \begin{cases} \max_{w \in s} \mathrm{ge}_{wd} & \text{if } \mathrm{ge}_{wd} > |\min_{w' \in s} \mathrm{ge}_{w'd}| \\ \min_{w \in s} \mathrm{ge}_{wd} & \text{otherwise} \end{cases}. \tag{4.13}$$

ge_{sd} and ge_{wd} denote the d-dimensional elements of ge_s and ge_w, respectively. We also used dist-n [58], Pointwise Mutual Information (PMI) [78] as evaluation metrics. dist-n and PMI represent response diversity and coherency, respectively. PMI of the response and the dialogue context is computed as,

$$\mathrm{PMI} = \frac{1}{\#wr} \sum_{wr}^{\#wr} \max_{wh} \mathrm{PMI}(wr, wh). \tag{4.14}$$

wr and wh are words in the response and the dialogue context, respectively. The corpus used to compute PMI is identical to the NCM training data.

Comparistion with Automatic Evaluation

Table 4.3 and 4.4 show the comparison results of responses with and without the re-raking on all of the test data. The method names from left to right indicate NCMs and the re-ranking methods. *1-best* indicate the baseline NCMs that do not re-rank response candidates. *Re-ranking (Pairs)*, *Re-ranking (RFTM)*, and *Re-ranking (Coherence)* indicate the re-ranking using only the event causality pairs, the re-ranking using the event causality pairs and RFTM, and the re-ranking using Coherence Model, respectively.

The ratios of the re-ranked response candidates are around 10% on *Re-ranking (Pairs)* and *Re-ranking (Coherence)*, indicating that the effects of the re-ranking methods are limited. The ratio was improved to 30% by the model that generalizes events with distributed representations of RFTM. NIST, dist-2 and PMI are

Table 4.3.: Automatic evaluation results for all the test data (1)

Method		Evaluation					
NCM	re-ranking	re-ranked (%)	BLEU	NIST	greedy	average	extrema
reference	-	-	-	-	-	-	-
EncDec	*1-best*	-	1.19	0.12	0.46	0.56	**0.46**
	Re-ranking (Pairs)	3,384 (9.80)	1.19	0.12	0.46	0.56	**0.46**
	Re-ranking (RFTM)	**11,412(33.06)**	1.38	0.22	0.45	0.55	0.45
	Re-ranking (Coherence)	2,972 (8.61)	1.21	0.16	**0.47**	**0.57**	**0.46**
HRED	*1-best*	-	**1.58**	2.64	0.44	0.56	0.45
	Re-ranking (Pairs)	2,608 (7.56)	1.56	2.62	0.44	0.56	0.45
	Re-ranking (RFTM)	11,247 (32.59)	1.57	**2.73**	0.44	0.56	0.45
	Re-ranking (Coherence)	3,245 (9.40)	1.53	2.61	0.45	0.56	**0.46**

Table 4.4.: Automatic evaluation results for all the test data (2)

Method		Evaluation		
NCM	re-ranking	dist-1	dist-2	PMI
reference	0.09	0.43	2.26	
EncDec	*1-best*	0.06	0.10	1.62
	Re-ranking (Pairs)	0.06	0.11	1.66
	Re-ranking (RFTM)	0.07	0.14	**1.92**
	Re-ranking (Coherence)	0.06	0.11	1.68
HRED	*1-best*	**0.08**	0.19	1.60
	Re-ranking (Pairs)	**0.08**	0.19	1.63
	Re-ranking (RFTM)	**0.08**	**0.20**	1.75
	Re-ranking (Coherence)	**0.08**	0.19	1.64

most improved by *Re-ranking (RFTM)*, indicating that the vocabulary is diverse and related to the dialogue contexts. Note that although the scores of BLEU and NIST are low for all methods, the scores of evaluation methods based on similarity with reference responses tend to be low because the appropriate responses are diverse in non-task oriented dialogues [62].

Automatic Evaluation Results per Re-ranking Method

In Table 4.3 and 4.4, the performances of all of the methods on the majority of the test data are computed for the same responses as before re-ranking. It is difficult to evaluate only from the results of Table 4.3 and 4.4 how much each re-ranking method improves performances. Based on the performances only for the data that each re-ranking method re-ranked responses, we analyzed the characteristics

of the data to be re-ranked by each method and the evaluation metrics that are improved by each method. Although EncDec is potentially better for re-ranking because it has higher diversity within the N-best responses in Table 4.2, HRED has higher performances at 1-best in the automatic evaluation. In this study, we used HRED, in which 1-best without re-ranking is a stronger baseline, and compared it to each proposed re-ranking method.

The results are shown in Table 4.5, 4.6 and 4.7. In the re-ranking using only the event causality pairs in Table 4.5, all the similarity scores with the references such as BLEU and NIST decreased from the values before re-ranking, while dist-2 and PMI improved. In particular, PMI is slightly higher than that of the references, indicating that the words in the responses are related to the dialogue contexts. In the re-ranking using the event causality pairs and RFTM in Table 4.6, PMI and NIST improved, indicating that the methods selected the low-frequent words related to the dialogue contexts. In the re-ranking using Coherence Model in Table 4.7, although BLEU, NIST, and dist decreased, the vector-based similarity metrics (greedy, average, extrema) improved. PMI is the highest value among the three re-ranking methods, and the difference from the value of the references is larger than the value in Table 4.5. Comparing the 1-best values in Table 4.7 with the values in Table 4.5 and 4.6 shows that the similarity scores and PMI are high before re-ranking. In other words, the re-ranking using Coherence Model can detect dialogues in which the response generation model can generate responses similar to the references compared to the other re-ranking methods. In addition, the method can improve the similarity of the responses in the distributed representation. We expect that the reason for the large improvements of PIM in all of the re-ranking methods is the fact that the event causal pairs are extracted based on the co-occurrence of the PA structures, and that the co-occurrence contributes to the improvement of PMI. Note that since the event causality pairs are also used for training RFTM of Coherence Model, *Re-ranking (Coherence)* also indirectly refers to the event causality pairs. Although the improvement of PMI suggests that the methods selected the responses that include the words related to the dialogue contexts, human evaluation is necessary to evaluate naturalness and dialogue continuity. In the next section, we conduct a human evaluation.

Table 4.5.: Automatic evaluation results; HRED with *Re-ranking (Pairs)*; 2,608 dialogues

Method	Evaluation (±%)							
	BLEU	NIST	greedy	average	extrema	dist-1	dist-2	PMI
reference	-	-	-	-	-	0.25	0.67	2.33
1-best	1.00	2.55	0.49	0.62	0.51	0.22	0.38	1.92
Re-ranking	0.67	2.31	0.49	0.62	0.50	0.21	0.39	2.34
	(-32.69%)	(-9.66%)	(±0%)	(±0%)	(-0.84%)	(-4.55%)	(+1.45%)	(+21.86%)

Table 4.6.: Automatic evaluation results; HRED with *Re-ranking (RFTM)*; 11,247 dialogues

Method	Evaluation (±%)							
	BLEU	NIST	greedy	average	extrema	dist-1	dist-2	PMI
reference	-	-	-	-	-	0.15	0.54	2.32
1-best	0.89	2.19	0.46	0.58	0.46	0.13	0.27	1.62
Re-ranking	0.84	2.46	0.45	0.58	0.47	0.12	0.28	2.10
	(-5.87%)	(+12.28%)	(-0.28%)	(±0%)	(+0.68%)	(-1.30%)	(+3.50%)	(+29.75%)

4.2.4. Human Evaluation

Although the proposed method improved the word coherency of the responses in the automatic evaluation, it is difficult to evaluate the performance of the dialogue system only with the automatic evaluation [62]. We evaluated coherency, naturalness, and dialogue continuity of the responses selected by the proposed methods by comparing the baseline model and the proposed methods on human evaluation. In order to reduce the workload of the workers, dialogues that requires prerequisite knowledge to understand the content, such as dialogues about social games that are generally not well-known, were manually removed from the evaluation. We used crowdsourcing for the human evaluation. Ten crowdworkers compared responses and selected from three options, either one of the two responses or "neither" in the following three subjective criteria. The first one is "which response is more related to the dialogue context (coherency)", which indicates system response coherency to dialogue contexts. The second one is "which response is more grammatically natural (naturalness)", which indicates the grammatical naturalness of system responses. The third criterion is "which

Table 4.7.: Automatic evaluation results; HRED with *Re-ranking (Coherence)*; 3,245 dialogues

Method	Evaluation (±%)							
	BLEU	NIST	greedy	average	extrema	dist-1	dist-2	PMI
reference	-	-	-	-	-	0.20	0.59	2.29
1-best	2.50	4.94	0.53	0.68	0.54	0.15	0.31	2.11
Re-ranking	2.00	4.71	0.55	0.72	0.57	0.13	0.28	2.60
	(-20.16%)	(-4.63%)	(+3.62%)	(+4.95%)	(+4.57%)	(-13.46%)	(-8.82%)	(+23.39%)

response is more attractive to respond to" (dialogue continuity), which indicates how much dialogue continuity system responses have. We were inspired to make these criteria by those of the Alexa Prize [87]. We used 100 dialogues for each evaluation. Note that since we used samples that different responses were selected with and without re-raking, the samples were different for each re-ranking method: the scores between different figures cannot be directly compared.

Human evaluation results are shown in Figure 4.8-4.13. We used the chi-square test. * and ** mean that $p < 0.05$, and $p < 0.01$ for a significant difference in performance. Coherency significantly decreased from 1-best on all of the re-ranking methods. This result indicates that focusing only on event causal relations or sequential events does not necessarily improve subjective coherency in dialogue. However, PMI increases in Table 4.5-4.7, suggesting that word-level coherency improved. These results indicate that word-level coherency is not sufficient for users to recognize a response as coherent, and that the first hypothesis "Improving coherency based on words does not necessarily contribute to greater coherency in human evaluations." in Chap. 1 is valid. Comparing the different re-ranking methods, the responses selected by *Re-ranking (Coherence)* have the highest coherency, indicating that *Re-ranking (Coherence)* has a higher ability to detect response coherency than the other re-ranking methods.

On the other hand, dialogue continuity is significantly improved by *Re-ranking (RFTM)*. Comparing the different re-ranking methods, the responses selected by *Re-ranking (RFTM)* and *Re-ranking (Coherence)* have the highest dialogue continuity, indicating that the improvement of coherency of events and words contributes to the improvement of dialogue continuity.

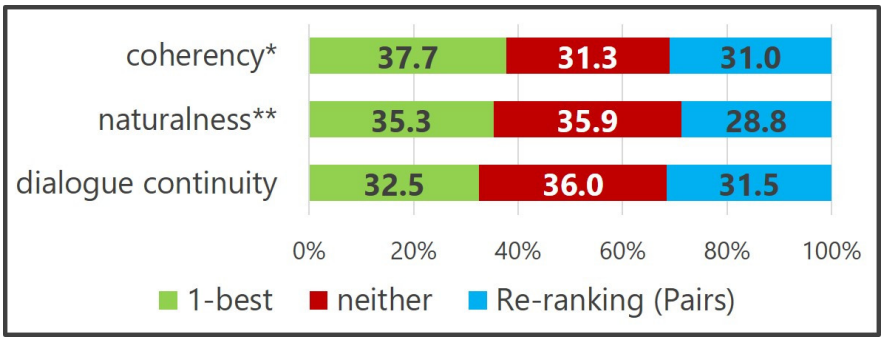


Figure 4.8.: Human evaluation results; *1-best* v.s. *Re-ranking (Pairs)*

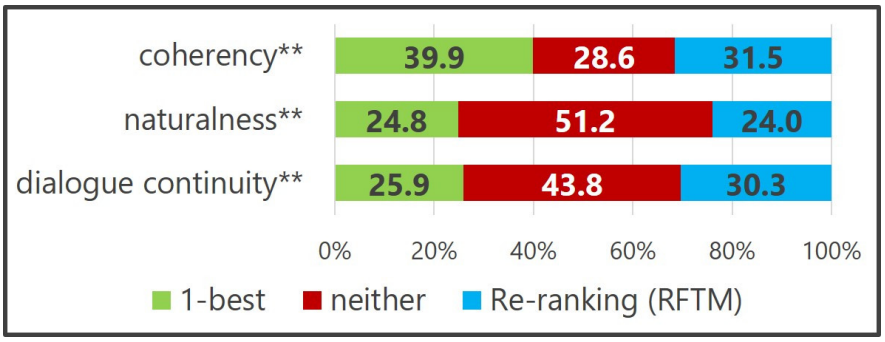


Figure 4.9.: Human evaluation results; *1-best* v.s. *Re-ranking (RFTM)*

Naturalness decreased on all of the re-ranking methods. This is because the Decoder of the neural dialogue model has the role as a language model, and the higher the rank before re-ranking, the higher the likelihood of the response in terms of the language model. Comparing the different re-ranking methods, *Re-ranking (Coherence)* has the highest naturalness. This is because *Re-ranking (Coherence)* deals with the coherency of the entire responses to the dialogue contexts, and detects the plausibility of the generated responses to the contexts.

The human evaluation results revealed that although responses with low naturalness are regarded as also having low coherency, dialogue continuity improves even if coherency and naturalness are low. To analyze these results in detail, we examine the correlations between the evaluation criteria in the next section.

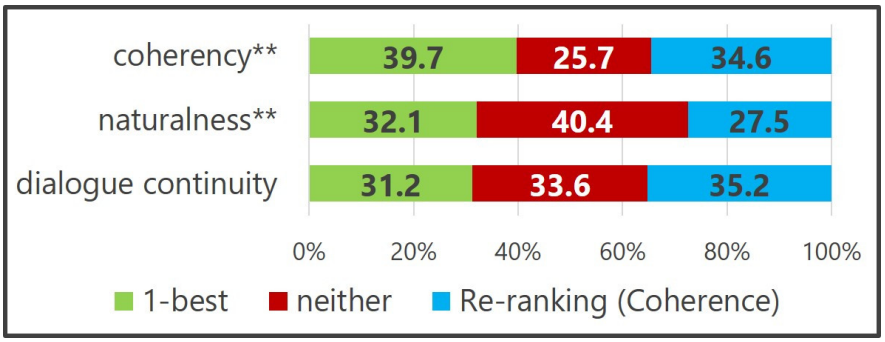


Figure 4.10.: Human evaluation results; *1-best* v.s. *Re-ranking (Coherence)*

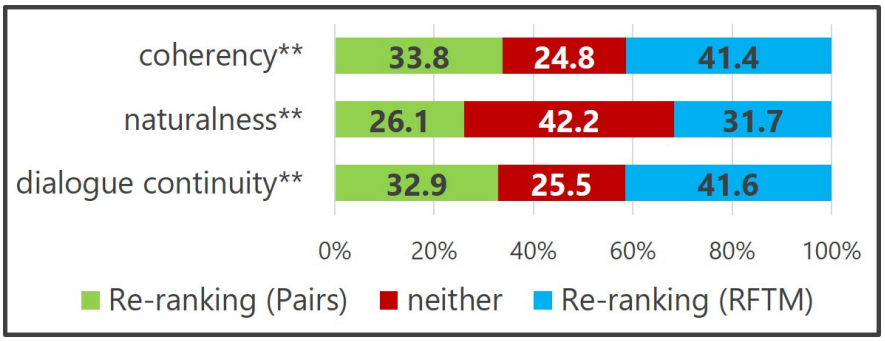


Figure 4.11.: Human evaluation results; *Re-ranking (Pairs)* v.s. *Re-ranking (RFTM)*

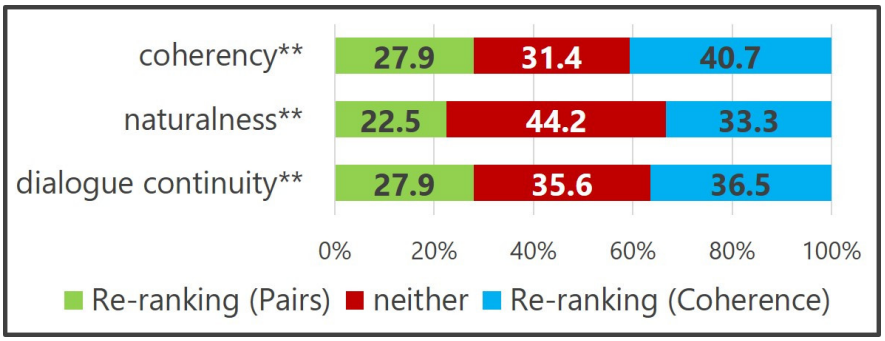


Figure 4.12.: Human evaluation results; *Re-ranking (Pairs)* v.s. *Re-ranking (Coherence)*

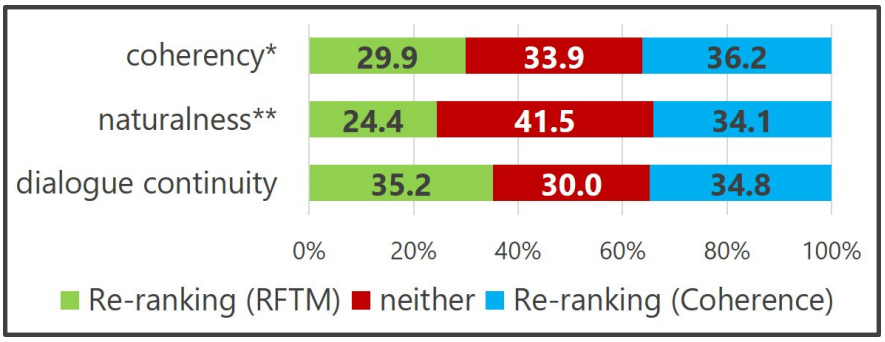


Figure 4.13.: Human evaluation results; *Re-ranking (RFTM)* v.s. *Re-ranking (Coherence)*

4.2.5. Correlation Analysis for Human Evaluation Results

The re-ranking method that focuses on the coherency of sequential events improved dialogue continuity but decreased coherency on the human evaluation. The results seem contradictory. To compute the correlation between coherency, naturalness, and dialogue continuity on the human evaluation, we made confusion matrices of the criteria and computed Cramér's $V (0 \leq V \leq 1)$. We analyzed each 100 dialogues on Figure 4.8-4.10. There are 300 dialogues in total. The Fleiss' Kappa value, which represents the agreement between workers, was 0.20. The confusion matrices and Cramér's V are shows in Table 4.8-4.10. The coefficients of association between coherency and naturalness, and between dialogue continuity and coherency are 0.20, indicating that the contribution of coherency to dialogue continuity is not high in the human evaluation. The coefficient of association between naturalness and dialogue continuity is 0.52, which is also not high. These results support the second hypotheses "Coherency and dialogue continuity in human evaluation have a low correlation." in Chap. 1.

The following dialogue is an example that coherency decreased but dialogue continuity improved by re-ranking in the human evaluation. The texts in square brackets are the event pairs used for re-ranking. The event pairs connected with $\rightarrow$ represent event causality relations.

Dialogue 1:

Table 4.8.: Confusion matrix (coherency-naturalness); $V = 0.20$

Coherency \ Naturalness	1-best	neither	Re-ranking	Total
1-best	518	439	216	1,173
neither	179	473	204	856
Re-ranking	225	363	383	971
Total	922	1,275	803	3000

Table 4.9.: Confusion matrix (naturalness-dialogue continuity); $V = 0.52$

Naturalness \ Dialogue continuity	1-best	neither	Re-ranking	Total
1-best	623	174	125	922
neither	186	831	258	1,275
Re-ranking	87	129	587	803
Total	896	1,134	970	3,000

Table 4.10.: Confusion matrix (dialogue continuity-coherency); $V = 0.20$

Dialogue continuity \ Coherency	1-best	neither	Re-ranking	Total
1-best	484	185	227	896
neither	381	453	300	1,134
Re-ranking	308	218	444	970
Total	1,173	856	971	3,000

User: It's the countryside of Tokyo. lol
System: But I envy you. Is it far from Harajuku?
User: I think I **can go** there in about an hour.
System (*1-best*): I see.
System (*Re-ranking (Pairs)*): Maybe we **can meet**.
[can go → can meet]

In the above example, the information provided (*can meet*) by the system seems to

attract the user's interest more than the simple response of *1-best*. The following dialogue is an example that dialogue continuity decreased but coherency improved by re-ranking. The events in parentheses are the events included in the event causality pairs. The events to the left of the parentheses represent events included in the dialog that were regarded to be identical to the events in the parentheses by RFTM.

Dialogue 2:

> **User: Go to bed**. lol
> **System:** Good morning. You woke up early.
> **User:** Good morning... You are the one who woke up early, usually around 9:00 a.m.
> **System (*1-best*):** I wonder if it's ok if I don't sleep well... lol
> **System (*Re-ranking (RFTM)*):** I **could sleep** well.
> [can sleep (sleep) → go to bed (sleep)]

The response before re-ranking was a question to the user, which helped to continue the dialogue. Although the event in the response after re-ranking is coherent to the dialogue context, the meaning of the response is the opposite. We revealed that improving the coherency of responses does not necessarily contribute to improving dialogue continuity based on the human evaluation and the case analysis. However, as shown below, there are examples where both response coherency and dialogue continuity improved.

Dialogue 3:

> **User:** Good morning!
> **System:** Good morning, I went fishing for flounder, but I was sunk.
> **User:** Good morning. It is **cold**, isn't it?
> **System (*1-best*):** Good morning.
> **System (*Re-ranking (Pairs)*):** Good morning. It' s **getting cooler** in the mornings and evenings.
> [getting cooler → cold]

Table 4.11.: Classification of results of PA structure analysis

	Correct	Wrong	Total
Each	424	176	600
Both	170	130	300

We need to conduct detailed analyses for constraints and conditions that contribute to improving dialogue continuity, including the coherency of events and responses.

4.3. Case Analysis

We verified whether the proposed method improves the event-level coherency of system responses to the dialogue contexts, resulting in improvement of the coherency and the dialogue continuity on the human evaluation. Although the proposed method improved the word-level coherency and the dialogue continuity, coherency decreased on the human evaluation. We analyzed in detail the re-ranking of responses by classifying the results of PA structure analysis and the event pairs used for the re-ranking. In addition, we analyzed the dialogue acts of the generated responses to clarify the correlation between the coherency and dialogue continuity of the responses and the dialogue acts. As with Section 4.2.5, we analyzed 300 dialogues on Figure 4.8-4.10. One worker annotated the results of PA structure analyses, the event pairs used for the re-ranking, and the dialogue acts.

4.3.1. Analysis for Predicate-Argument Structure Parsing Results

We need to verify the accuracy of the PA structure analyzer because the proposed methods utilize events that are automatically extracted using PA structure analysis. We investigated the ratios that the PA structure analyzer extracted the correct events used for the re-ranking. Table 4.11 shows the result. *Correct* on the horizontal axis indicates that the PA structure analyzer made no error for

Table 4.12.: Classification of event pairs

Re-ranking \ Events	Good	Bad (Pairs)	Bad (Over)	Bad (Sequence)	Total
Pairs	61	39	-	-	100
RFTM	5	12	83	-	100
Coherence	52	-	-	48	100

the events used for the re-ranking, while *Wrong* indicates that the analyzer made errors. For example, there are cases in which the wrong predicate (determiner) "scam (*sagi* in Japanese)" is extracted from the utterance "Ohayousagi (combination of good morning (*ohayo* in Japanese) and rabbit (*usagi* in Japanese))," and cases in which the case analysis is wrong, such as "Sakae goes" for the utterance "I'm not sure if I should go to Sakae." *Each* on the vertical axis is the case where two events in an event pair are classified into correct or incorrect separately, and *Both* is the case where two events are classified into correct or incorrect together. Therefore, *Both* is regarded as *Correct* only if both events in the event pair were extracted without error.

The percentage of cases that all events were perfectly extracted by the PA structure analyzer is around 70% for *Each* and around 60% for *Both*, which is not high. Note that *Both* is the evaluation of the analysis results for two PA structures. In addition, the distributed representation of events used in this study may generalize the problems such as case analysis errors.

4.3.2. Analysis for Event Pairs Used for Re-ranking

Since the proposed method is based on the coherency of event pairs, we need to investigate the coherency of the event pairs used for the re-ranking and the tendency of the human evaluation when coherent event pairs are used. We classified and analyzed the coherency of the event pairs used for the re-ranking, and discussed the results of the human evaluation. Table 4.12 shows the results. Each column indicates the coherency of the event pairs used for the re-ranking. *Good* indicates that coherent event pairs were used, and *Bad (Pairs)* indicates that the methods used causality relations of the event causality pairs that are

not appropriate in the target dialogue contexts. *Bad (Over)* indicates that the re-ranking method used causality relations that were overgeneralized by the distributed event representation, and *Bad (Sequence)* indicates that continuous but incoherent event pairs were used. These samples are manually classified.

The ratios that coherent event pairs were used are 61% for *Re-ranking (Pairs)*, 5% for *Re-ranking (RFTM)*, and 52% for *Re-ranking (Coherence)*. Figure 4.14-4.16 shows the results of extracting the cases that coherent event pairs were used for the re-ranking from the dialogue samples of the human evaluation on Figure 4.8-4.10. Since each sample was evaluated by ten workers, the number of samples of Figure 4.14-4.16 are 610, 50, and 520, respectively. In *Re-ranking (RFTM)*, coherency, naturalness, and dialogue continuity improved when coherent event pairs were used, suggesting that coherent event pairs contribute to improving dialogue continuity. On the other hand, in *Re-ranking (Pairs)* and *Re-ranking (Coherence)*, there is no significant difference with and without re-ranking. Table 4.5-4.7 show that the automatic evaluation scores of the 1-best responses re-ranked by *Re-ranking (RFTM)* or Re-ranking (Coherence) are higher than the scores of the 1-best responses re-ranked by *Re-ranking (RFTM)*. In other words, the re-ranking with coherent event pairs is ineffective when it re-ranks 1-best responses with high automatic evaluation scores, but can improve response coherency, naturalness, and dialogue continuity when it re-ranks responses with low automatic evaluation scores. These results support the third hypothesis of Chap. 1, "Improving coherency based on words, rather than coherency in human evaluation, improves dialogue continuity in human evaluations." to some extent. Note that since the ratio of coherent event pairs used by the *Re-ranking (RFTM)* is low (5%), we need to improve the recall of the re-ranking based on coherent event pairs by improving the distributed event representation and the precision of event coherency on the distributed event representation.

The following dialogues are examples that coherent event pairs were used for the re-ranking.

Dialogue 4:

> **User:** I can't believe I **got sick** so early in the new year... I have to get better by tomorrow or the day after tomorrow.

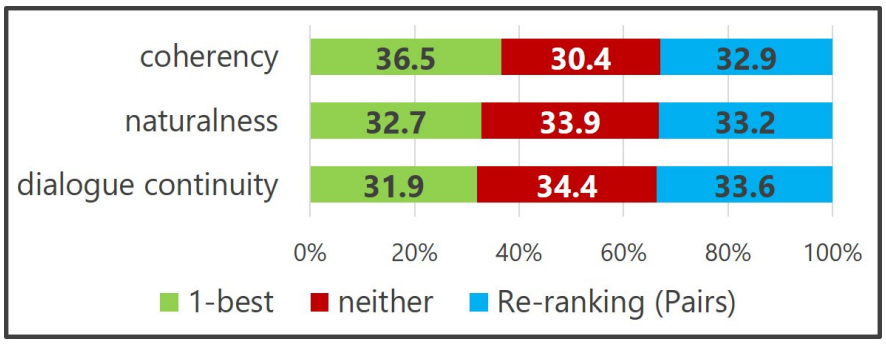


Figure 4.14.: Human evaluation results where coherent event pairs were used; *1-best* v.s. *Re-ranking (Pairs)*

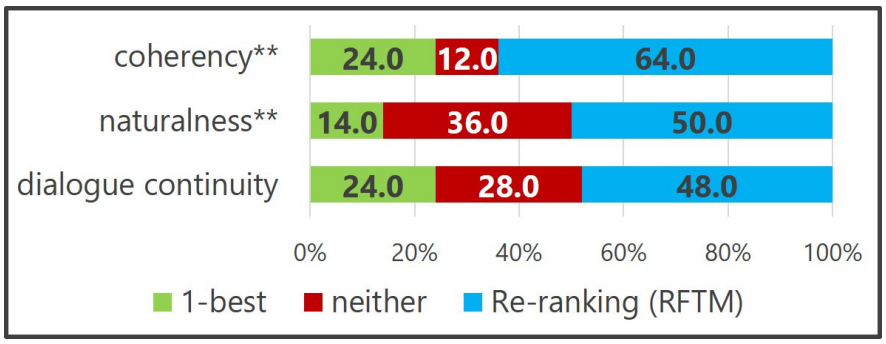


Figure 4.15.: Human evaluation results where coherent event pairs were used; *1-best* v.s. *Re-ranking (RFTM)*

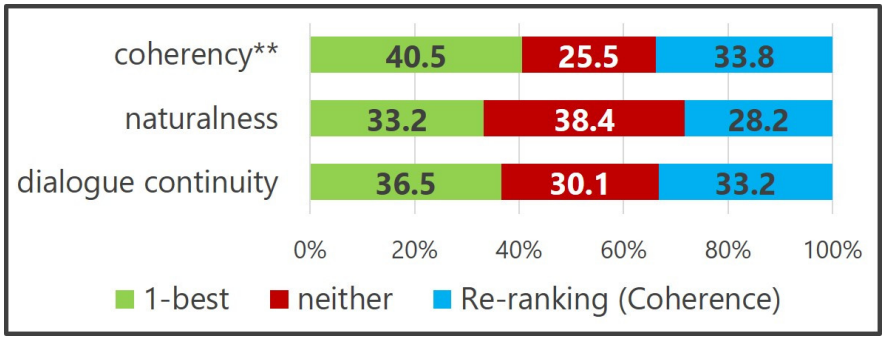


Figure 4.16.: Human evaluation results where coherent event pairs were used; *1-best* v.s. *Re-ranking (Coherence)*

System (*1-best*): Bless you.
System (*Re-ranking (Pairs)*): Are you okay? Please don't **overwork**.
[overwork → get sick]

Dialogue 5:

System: Thank you for following me. I have followed you back.
User: I'm free except for exams, so you can **talk to** me anytime.
System (*1-best*): Yes, by all means, let's be friends.
System (*Re-ranking (RFTM)*): I'm totally fine with it. How should I **call** you?
[talk to (talk to) → call (talk to)]

Dialogue 6:

User: Everyone should **eat** tasty seven-eleven **donuts**.
System (*1-best*): It's tasty.
System (*Re-ranking (Coherence)*): I **want to eat chocolate chip**.
[eat donuts, want to eat chocolate chip]

In the above examples, the coherent responses were selected by the re-ranking using the coherent event pairs. On the other hand, there are examples that incoherent responses were selected as shown in Table 4.12. We found cases where incoherent event pairs were used as shown in Dialogue 7, and cases where responses were unnatural even when coherent event pairs were used as shown in Dialogue 8.

Dialogue 7:

System: Let's go to Korean BBQ.
User: I **want to go**$\cdots$

System (*1-best*): I will go.
System (*Re-ranking (RFTM)*): Let's **go to a** Korean BBQ **restaurant**.
[want to go (go to karaoke) → go to a restaurant (go to hot spring)]

Dialogue 8:

System: Get over your cold. Let's go when you're better.
User: I've had a cold for about three weeks.
System (*1-best*): Bless you.
System (*Re-ranking (Coherence)*): I'm **getting over a cold**.
[have a cold, get over a cold]

The event causality relation in Dialogue 7 is obtained by generalizing *go to hot spring* with *go to a restaurant*, indicating that we need to control overgeneralization. For Dialogue 8, the response became unnatural due to selecting the event representing the passage of time, although the coherency between events improved. One possible solution is selecting a natural response using multi-hop causality such as *have a cold* → *want to get over a cold* → *bless you*.

4.3.3. Analysis for Dialogue Act of Response

Dialogue act is tags that indicate intentions of speakers and functions of responses in dialogues [17, 18]. We reveal the differences in the functions of the responses before and after the re-ranking, and the differences of the human evaluation scores before and after the re-ranking, by analyzing the correlation between the changes in the dialogue acts of the responses and the human evaluation scores. We annotated dialogue acts based on the dialogue act tag set [111] that is the extended tags of ISO-24617-2 [17, 18]. Table 4.13 shows the dialogue act tag set.

Table 4.14 shows the differences in the dialogue acts before and after the re-ranking. The dialogue acts of about half of the responses changed by all of the re-ranking methods. In order to confirm whether the re-ranking methods change particular dialogue acts into other acts, we computed Cramér's V from the

Table 4.13.: Dialogue act tag set

Information seeking (IS)	Seeking information with questions.
Information providing (IP)	Providing new information or information that was asked by the interlocutor.
Commissive (CO)	Proposing or promising something to the interlocutor.
Directive (DI)	Asking the interlocutor to do something.
Auto/allo-feedback (AA)	Expressing understanding to the interlocutor with auto/allo-feedback.
Own/partner comm. man. (CM)	Supplementing or correcting the contents of the utterance of the speaker or the interlocutor.
Discourse structure man. (DS)	Clarifying the content of what the speaker is going to say.
Social obligations man. (SO)	Social utterances such as greetings, self-introductions, appreciations, and apologies.

Table 4.14.: Number of different dialogue acts before and after re-ranking (100 dialogues)

Re-ranking	Number	Cramér's V
Pairs	60	0.30
RFTM	59	0.33
Coherence	51	0.38

confusion matrix of the dialogue acts before and after the re-ranking. Table 4.14 shows all of the coefficients of association were less than 0.4, indicating that the correlations between the dialogue acts before and after the re-ranking are not high, and that there is no strong tendency in the change of dialogue acts by the re-ranking. Table 4.15-4.17 show the results of analyzing whether the

human evaluation scores changed when the dialogue act changed. Note that the total number of each dialogue act is 100 because one dialogue act is assigned to each response, while the total number of each human evaluation metric is 1,000 because ten workers evaluate each response in the human evaluation. The tables show that all of the coefficients of association were less than 0.4, indicating that the correlations are not high. In other words, although the human evaluation scores can change when dialogue acts change as shown in Dialogue 1 and 2 in Section 4.2.5, we did not observe any strong tendency of the change of the human evaluation scores by the change of particular dialogue acts to other acts.

4.4. Conclusion

We proposed the methods for re-ranking N-best response candidates generated by an NCM based on the coherency of sequential events. The methods select response candidates that are coherent to dialogue contexts by focusing on the coherency of PA structures (events). In addition, we also proposed the re-raking method that deals with the coherency of whole dialogues based on Coherence Model. The experimental results show that our methods improve the coherency of words and events on the automatic evaluation metrics such as PMI. On the other hand, the methods improve dialogue continuity but decrease coherency on the human evaluation. The results seem contradictory. We conducted the correlation analysis and the case analysis. The results show that dialogue continuity and coherency do not have a strong correlation on the human evaluation, and that coherency of event pairs seems to contribute to improving dialogue continuity.

Our future work will investigate constraints and conditions that improve dialogue continuity. We will also improve distributed event representation and develop methods that generate responses including coherent events to dialogue contexts.

Table 4.15.: Changes of dialogue acts and human evaluation results by *Re-ranking (Pairs)*; The numbers in the columns represent the number of the response dialogue acts that were rated as 1-best, neither, or Re-ranking in the human evaluation when the dialogue acts were changed to another dialogue act by re-ranking. Note that the total number of each dialogue act is 100 because one dialogue act is assigned to each response, while the total number of each human evaluation metric is 1,000 because ten workers evaluate each response in the human evaluation.

Dialogue act			Coherency ($V = 0.35$)			Naturalness ($V = 0.36$)			Dialogue continuity ($V = 0.32$)		
1-best	Re-ranking	Total	1-best	neither	Re-ranking	1-best	neither	Re-ranking	1-best	neither	Re-ranking
IS	IS	1	2	1	7	1	1	8	1	2	7
	IP	3	18	8	4	9	11	10	11	11	8
	CO	0	0	0	0	0	0	0	0	0	0
	DI	1	0	2	8	3	2	5	5	2	3
	AA	0	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	0	0	0	0	0	0	0	0	0	0
IP	IS	1	7	3	0	10	0	0	6	2	2
	IP	11	35	47	28	36	50	24	40	39	31
	CO	2	4	14	2	10	6	4	9	8	3
	DI	2	7	4	9	4	8	8	10	5	5
	AA	4	24	5	11	23	9	8	19	9	12
	CM	2	3	4	13	2	10	8	5	6	9
	DS	0	0	0	0	0	0	0	0	0	0
	SO	4	11	7	22	3	8	29	15	9	16
CO	IS	0	0	0	0	0	0	0	0	0	0
	IP	0	0	0	0	0	0	0	0	0	0
	CO	0	0	0	0	0	0	0	0	0	0
	DI	1	5	3	2	3	6	1	2	5	3
	AA	0	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	1	0	6	4	0	4	6	2	5	3
DI	IS	0	0	0	0	0	0	0	0	0	0
	IP	5	18	12	20	19	10	21	19	14	17
	CO	0	0	0	0	0	0	0	0	0	0
	DI	7	16	39	15	9	44	17	6	42	22
	AA	0	0	0	0	0	0	0	0	0	0
	CM	3	14	10	6	23	5	2	21	6	3
	DS	0	0	0	0	0	0	0	0	0	0
	SO	1	9	0	1	8	2	0	9	1	0
AA	IS	1	0	5	5	1	4	5	0	3	7
	IP	6	32	5	23	21	20	19	14	16	30
	CO	1	5	1	4	3	6	1	2	4	4
	DI	3	22	1	7	20	3	7	13	2	15
	AA	8	27	34	19	29	36	15	20	44	16
	CM	3	8	12	10	12	10	8	7	13	10
	DS	0	0	0	0	0	0	0	0	0	0
	SO	1	7	3	0	6	4	0	7	2	1
CM	IS	1	3	5	2	3	4	3	4	5	1
	IP	1	2	2	6	3	3	4	2	3	5
	CO	0	0	0	0	0	0	0	0	0	0
	DI	0	0	0	0	0	0	0	0	0	0
	AA	1	6	2	2	7	1	2	7	2	1
	CM	3	8	17	5	8	15	7	6	20	4
	DS	0	0	0	0	0	0	0	0	0	0
	SO	1	2	3	5	0	2	8	2	1	7
DS	IS	0	0	0	0	0	0	0	0	0	0
	IP	0	0	0	0	0	0	0	0	0	0
	CO	0	0	0	0	0	0	0	0	0	0
	DI	0	0	0	0	0	0	0	0	0	0
	AA	0	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	0	0	0	0	0	0	0	0	0	0
SO	IS	0	0	0	0	0	0	0	0	0	0
	IP	7	27	16	27	27	21	22	23	18	29
	CO	0	0	0	0	0	0	0	0	0	0
	DI	2	2	1	17	4	3	13	3	4	13
	AA	1	8	0	2	10	0	0	8	2	0
	CM	1	5	2	3	7	3	0	3	5	2
	DS	0	0	0	0	0	0	0	0	0	0
	SO	10	40	39	21	29	48	23	24	50	26
Total		100	377	313	310	353	359	288	325	360	315

Table 4.16.: Changes of dialogue acts and human evaluation results by *Re-ranking (RFTM)*; The numbers in the columns represent the number of the response dialogue acts that were rated as 1-best, neither, or Re-ranking in the human evaluation when the dialogue acts were changed to another dialogue act by re-ranking. Note that the total number of each dialogue act is 100 because one dialogue act is assigned to each response, while the total number of each human evaluation metric is 1,000 because ten workers evaluate each response in the human evaluation.

Dialogue act			Coherency ($V = 0.38$)			Naturalness ($V = 0.34$)			Dialogue continuity ($V = 0.39$)		
1-best	Re-ranking	Total	1-best	neither	Re-ranking	1-best	neither	Re-ranking	1-best	neither	Re-ranking
IS	IS	3	14	0	16	15	9	6	12	5	13
	IP	2	5	2	13	3	6	11	8	3	9
	CO	0	0	0	0	0	0	0	0	0	0
	DI	0	0	0	0	0	0	0	0	0	0
	AA	1	6	1	3	6	3	1	9	1	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	0	0	0	0	0	0	0	0	0	0
IP	IS	1	9	0	1	5	4	1	6	1	3
	IP	17	48	66	56	17	97	56	16	96	58
	CO	2	19	1	0	16	4	0	15	2	3
	DI	2	7	3	10	4	9	7	6	4	10
	AA	2	4	3	13	3	10	7	2	11	7
	CM	2	4	3	13	3	7	10	1	3	16
	DS	0	0	0	0	0	0	0	0	0	0
	SO	1	1	1	8	1	6	3	2	1	7
CO	IS	0	0	0	0	0	0	0	0	0	0
	IP	1	3	2	5	0	5	5	1	5	4
	CO	0	0	0	0	0	0	0	0	0	0
	DI	0	0	0	0	0	0	0	0	0	0
	AA	0	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	0	0	0	0	0	0	0	0	0	0
DI	IS	0	0	0	0	0	0	0	0	0	0
	IP	3	17	6	7	15	13	2	18	10	2
	CO	2	1	18	1	0	20	0	0	19	1
	DI	3	4	25	1	1	26	3	1	26	3
	AA	0	0	0	0	0	0	0	0	0	0
	CM	1	6	1	3	8	2	0	8	0	2
	DS	0	0	0	0	0	0	0	0	0	0
	SO	0	0	0	0	0	0	0	0	0	0
AA	IS	5	20	9	21	8	19	23	9	12	29
	IP	5	29	11	10	17	21	12	14	17	19
	CO	1	4	1	5	2	3	5	2	2	6
	DI	5	26	5	19	20	19	11	13	13	24
	AA	10	34	48	18	18	65	17	23	62	15
	CM	8	40	14	26	28	33	19	29	26	25
	DS	0	0	0	0	0	0	0	0	0	0
	SO	2	11	1	8	7	8	5	9	3	8
CM	IS	1	3	1	6	6	2	2	3	6	1
	IP	3	13	9	8	4	17	9	8	14	8
	CO	0	0	0	0	0	0	0	0	0	0
	DI	0	0	0	0	0	0	0	0	0	0
	AA	0	0	0	0	0	0	0	0	0	0
	CM	1	4	3	3	2	7	1	2	7	1
	DS	0	0	0	0	0	0	0	0	0	0
	SO	0	0	0	0	0	0	0	0	0	0
DS	IS	0	0	0	0	0	0	0	0	0	0
	IP	0	0	0	0	0	0	0	0	0	0
	CO	0	0	0	0	0	0	0	0	0	0
	DI	0	0	0	0	0	0	0	0	0	0
	AA	0	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	0	0	0	0	0	0	0	0	0	0
SO	IS	0	0	0	0	0	0	0	0	0	0
	IP	5	16	13	21	11	27	12	13	23	14
	CO	0	0	0	0	0	0	0	0	0	0
	DI	2	4	12	4	3	15	2	6	11	3
	AA	0	0	0	0	0	0	0	0	0	0
	CM	2	10	1	9	5	10	5	6	10	4
	DS	0	0	0	0	0	0	0	0	0	0
	SO	7	37	26	7	20	45	5	17	45	8
Total		100	399	286	315	248	512	240	259	438	303

Table 4.17.: Changes of dialogue acts and human evaluation results by *Re-ranking (Coherence)*; The numbers in the columns represent the number of the response dialogue acts that were rated as 1-best, neither, or Re-ranking in the human evaluation when the dialogue acts were changed to another dialogue act by re-ranking. Note that the total number of each dialogue act is 100 because one dialogue act is assigned to each response, while the total number of each human evaluation metric is 1,000 because ten workers evaluate each response in the human evaluation.

Dialogue act			Coherency ($V = 0.25$)			Naturalness ($V = 0.23$)			Dialogue continuity ($V = 0.29$)		
1-best	Re-ranking	Total	1-best	neither	Re-ranking	1-best	neither	Re-ranking	1-best	neither	Re-ranking
IS	IS	5	19	13	18	9	22	19	6	22	22
	IP	2	14	3	3	12	4	4	15	1	4
	CO	1	4	3	3	3	5	2	6	3	1
	DI	0	0	0	0	0	0	0	0	0	0
	AA	0	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	2	10	2	8	9	4	7	15	2	3
IP	IS	2	10	3	7	8	7	5	5	7	8
	IP	25	85	76	89	70	104	76	70	86	94
	CO	3	15	8	7	11	15	4	9	8	13
	DI	4	14	5	21	16	12	12	16	5	19
	AA	1	1	3	6	3	4	3	3	5	2
	CM	1	3	2	5	2	3	5	3	4	3
	DS	0	0	0	0	0	0	0	0	0	0
	SO	2	6	7	7	4	13	3	6	10	4
CO	IS	0	0	0	0	0	0	0	0	0	0
	IP	0	0	0	0	0	0	0	0	0	0
	CO	2	3	10	7	7	8	5	3	13	4
	DI	2	9	8	3	10	8	2	8	8	4
	AA	0	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	0	0	0	0	0	0	0	0	0	0
DI	IS	0	0	0	0	0	0	0	0	0	0
	IP	4	24	5	11	12	14	14	16	13	11
	CO	2	12	2	6	5	7	8	6	5	9
	DI	2	1	3	16	2	8	10	2	2	16
	AA	0	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	3	9	10	11	7	16	7	9	17	4
AA	IS	0	0	0	0	0	0	0	0	0	0
	IP	9	43	22	25	34	37	19	25	29	36
	CO	1	2	3	5	3	2	5	4	0	6
	DI	1	4	0	6	2	5	3	2	2	6
	AA	1	1	3	6	0	6	4	0	3	7
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	1	6	1	3	6	4	0	6	3	1
CM	IS	2	10	3	7	7	10	3	10	4	6
	IP	3	12	11	7	11	12	7	9	13	8
	CO	0	0	0	0	0	0	0	0	0	0
	DI	1	1	5	4	1	3	6	1	3	6
	AA	0	0	0	0	0	0	0	0	0	0
	CM	2	3	7	10	5	11	4	6	8	6
	DS	0	0	0	0	0	0	0	0	0	0
	SO	0	0	0	0	0	0	0	0	0	0
DS	IS	0	0	0	0	0	0	0	0	0	0
	IP	0	0	0	0	0	0	0	0	0	0
	CO	0	0	0	0	0	0	0	0	0	0
	DI	0	0	0	0	0	0	0	0	0	0
	AA	0	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	0	0	0	0	0	0	0	0	0	0
SO	IS	1	10	0	0	10	0	0	7	0	3
	IP	2	6	3	11	6	4	10	6	1	13
	CO	0	0	0	0	0	0	0	0	0	0
	DI	1	3	4	3	4	2	4	1	5	4
	AA	0	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	0	0	0	0	0	0
	DS	0	0	0	0	0	0	0	0	0	0
	SO	12	57	32	31	42	54	24	37	54	29
Total		100	397	257	346	321	404	275	312	336	352

5. Reflective Action Selection Based on Positive-Unlabeled Learning and Causality Detection Model

This chapter investigates the dialogue system to select the reflective actions to the user utterances on the task-oriented dialogue. First, we collect a high-quality corpus consisting of ambiguous requests and reflective actions by devising the collection method. Next, we propose a PU learning method that incorporates event causality knowledge based on the characteristics that the collected corpus is an incompletely labeled dataset. Finally, we conduct detailed analyses of the classification performances of the proposed PU learning method and the mechanism of the PU learning method itself.

5.1. Reflective System Action to Ambiguous User Requests

Existing task-oriented dialogue systems assume that user intentions are clarified and uttered in an explicit manner; however, since users often don't really know what they want to request, their requests are sometimes ambiguous. Taylor [98, 99] categorizes user states in information search into four levels by their clarity (Table 1.1.)

Most existing task-oriented dialogue systems [69, 102] convert explicit user requests (Q3) into machine readable expressions (Q4). Future dialogue systems need to take appropriate actions even in situations such as Q1 and Q2, where

Figure 5.1.: Example of reflective action

the users fail to clearly verbalize their requests [110]. We used crowdsourcing to collect ambiguous user requests and linked them to appropriate system actions. This section describes the data collection.

5.1.1. Collecting Ambiguous Requests and Reflective System Actions

This study deals with a conversation between a user and a dialogue agent on a smartphone application in a domain of tourist information navigation. When a user makes an ambiguous request, the agent responds with reflective actions. In this study, we assume that the users know that the agent can respond with reflective actions to ambiguous requests if they engage with the system though chatting. Figure 5.1 shows an example dialogue between a user and a dialogue agent. This user utterance, "I love this view!", does not request a specific function. The dialogue agent responds with a reflective action, "Should I launch the camera application?" and starts it.

The WOZ method, in which two subjects play user and dialogue agent roles, is widely used to collect dialogue samples [16, 48]. However, even human workers sometimes struggle to respond with reflective actions to ambiguous user requests. In other words, a general WOZ dialogue is inappropriate for collecting such reflective actions. Moreover, these reflective actions must be linked to a system's API functions because the possible agent actions are limited by its applications. In other words, we can qualify the corpus by collecting antecedent ambiguous user

requests to defined possible agent actions. Therefore, we collected request-action pairs by asking crowd-workers to input antecedent ambiguous user requests for the pre-defined agent action categories.

We defined three major functions for the dialogue agent: *spot search*, *restaurant search*, and *application (app) launch*. Table 5.1 shows the defined functions. Each function has its own categories. The actions of the dialogue agent in the corpus are generated by linking them to these 70 categories. The functions and categories are defined heuristically according to Kyoto sightseeing web sites*:

- *Spot search*: a function that seeks specific spots presented to the user as an action, such as "Should I search for an art museum around here?"
- *Restaurant search*: a function that looks for specific restaurants presented to the user as an action, such as "Should I search for shaved ice around here?"
- *App launch*: a function that launches a specific application presented to the user as an action, such as "Should I launch the camera application?"

We used crowdsourcing† to collect a Japanese corpus based on the pre-defined action categories of the dialogue agent. Figure 5.2 shows an example of an instruction and input form for the corpus collection. Since the user requests (utterances) to be collected in our study need to be ambiguous, we avoid such utterances with a clear request, such as, "Search for rest areas around here." Each worker was asked to input user requests for ten different categories.

The statistics of the collected corpus are shown in Table 5.2. The request examples in the corpus are shown in Table 5.3, which shows that we collected ambiguous user requests when the pre-defined action were regarded as reflective. In Chap. 1, we explained that *ambiguous* denotes that users failed to clearly define and verbalize their requests even when they have such potential requests, which can be associated with system actions. To check whether the collected user requests are ambiguous, we automatically filtered them with the following criterion: "The predefined action category names are not included in the user

*https://ja.kyoto.travel/
†https://crowdworks.jp/

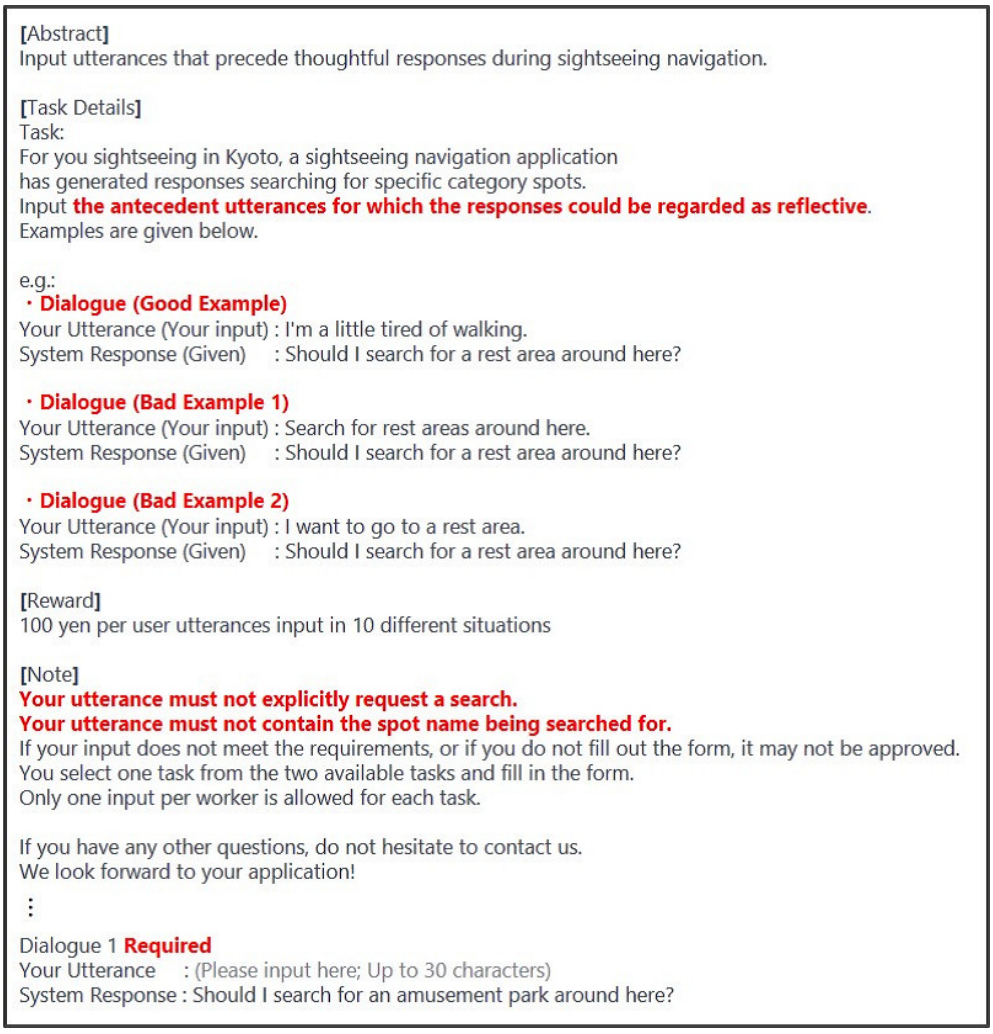


Figure 5.2.: Instruction and input form for corpus collection. Actual form is in Japanese; figure was translated into English.

Table 5.1.: Functions and categories of dialogue agent: # means number of categories.

Function	Category	#
spot search	amusement park, park, sports facility, experience-based facility, souvenir shop, zoo, aquarium, botanical garden, tourist information center, shopping mall, hot spring, temple, shrine, castle, nature/landscape, art museum, history museum, kimono-rental, fall colors, cherry blossom, rickshaw, train station, bus stop, rest area, Wi-Fi spot, quiet place, beautiful place, fun place, wide-open place, nice view place	30
restaurant search	cafe, Japanese tea, shaved ice, Japanese sweets, western-style sweets, curry, traditional Kyoto food, tofu cuisine, bakery, fast food, noodles, Japanese stew, rice bowls or fried food, meat dishes, sushi or fish dishes, flour-based foods, Kyoto cuisine, Chinese, Italian, French, child-friendly restaurant or family restaurant, tea-ceremony dishes, Buddhist vegetarian cuisine, vegetarian restaurant, izakaya or bar, food court, breakfast, cheap restaurant, average-priced restaurant, expensive restaurant	30
app launch	camera, photo, weather, music, transfer navigation, message, phone, alarm, browser, map	10

Table 5.2.: Corpus statistics

Function	Ave. length	# Requests
spot search	13.44 (±4.69)	11,670
restaurant search	14.08 (±4.82)	11,670
app launch	13.08 (±4.65)	3,890
all	13.66 (±4.76)	27,230

Table 5.3.: Examples of user requests in corpus: Texts are translated from Japanese to English. User requests for all pre-defined system actions are available in A.1.

User request (collected by crowdsourcing)	System action (pre-defined)
I'm hot and uncomfortable.	Should I search for a hot spring around here?
I've been eating a lot of Japanese food lately, and I'm getting a bit bored with it.	Should I search for some meat dishes around here?
Nice view.	Should I launch the camera application?

requests." If unambiguous user requests were included, we removed them and re-collected the same number of ambiguous user requests. Thus, the collected user requests include those where the user's intentions themselves are clear, such as "I want to take a nap on the grass" or "Where can I see pandas?" However, the system has to learn the correspondences between these user's intentions and predefined actions. In other words, these user requests are included in the ambiguous requests that we defined. The actual examples are shown in A.1. The collected corpus, which contains 27,230 user requests, was split into data of training:validation:test $= 24,430 : 1,400 : 1,400$. Each data set contains every category in the same proportion.

Table 5.4.: # Added action categories

Function	# Added categories
spot search	8.45 (±7.34)
restaurant search	9.81 (±7.77)
app launch	5.06 (±8.48)
all	8.55 (±7.84)

5.1.2. Multi-class Problem on Ambiguous User Requests

Since the user requests collected in Section 5.1.1 are ambiguous, some of the 69 unannotated actions other than the pre-defined actions can be reflective. Although labeling whether all combinations of user requests and system actions are reflective is costly and impractical, a comprehensive study is necessary to determine actual reflective actions. Thus, we completely annotated all the combinations of 1,400 user requests and system actions in the test data.

We used crowdsourcing for this additional annotation. We gave pairs of user requests and unannotated actions to crowdworkers and asked them to make a binary judgment whether each action was "contextually natural and reflective to the user request." Each pair was judged by three workers; the final decision was made by majority vote.

The number of added action categories that were identified as reflective is shown in Table 5.4. 8.55 different categories on average were additionally identified as reflective. The standard deviation was 7.84; this indicates that the number of added categories varied greatly for each user request. Comparing the number of added categories for each function, *restaurant search* has the highest average at 9.81 and *app launch* has the lowest average at 5.06. The difference is caused by the target range of the functions; *restaurant search* contains the same intention with different slots, while *app launch* covers different types of system roles. The second example in Table 5.3, "I've been eating a lot of Japanese food lately, and I'm getting a bit bored with it," suggests that any type of restaurant other than Japanese is a reflective action in this dialogue context.

Table 5.5 shows the detailed decision ratios of the additional annotation. The

Table 5.5.: Decision ratios of additional annotation: # means number of workers who identified each request and action pair as reflective. Fleiss' kappa value is 0.4191.

#	Ratio (%)
0	70,207 (72.68)
1	14,425 (14.93)
2	6,986 (7.23)
3	4,982 (5.16)
all	96,600

ratios where two or three workers identified each pair of a user request and a system action as reflective are 7.23 and 5.16, which indicates that one worker identified about 60% added action categories as unreflective. The Fleiss' kappa value is 0.4191, and the inter-annotator agreement is moderate.

Figure 5.3 shows a heatmap of the given and added categories. From the top left of both the vertical and horizontal axes, each line indicates one category in the order listed in Table 5.1. The highest value corresponding to the darkest color in Figure 5.3 is 20 because 20 ambiguous user requests are contained for each given action in the test data. Actions related to the same role are annotated in the *spot search* and *restaurant search* functions. One action near the rightmost column is identified as reflective for many contexts. This action category was *browser* in the *app launch* function, which is expressed as "Should I display the information about it?" *Spot search* and *restaurant search* also had one action category annotated as reflective action for many antecedent requests: *tourist information center* and *food court*.

Table 5.6 shows pairs with large values in Figure 5.3. For any combination, both actions can be responses to the given ambiguous requests.

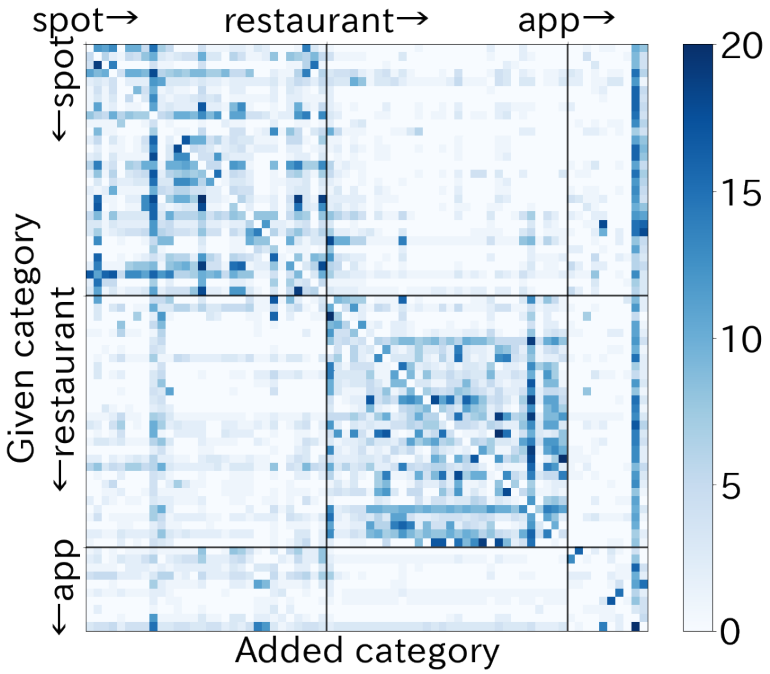


Figure 5.3.: Heat map of given and added categories

5.2. Reflective Action Classification by Positive-Unlabeled Learning and Causality Model

We collected pairs of ambiguous user requests and reflective system action categories in Section 5.1. Using these data, we developed a model that chooses reflective actions to given ambiguous user requests, which it classifies into categories of corresponding actions. Positive/negative (PN) learning is widely used for such classification problems, where the collected ambiguous user requests and the corresponding system action categories are taken as positive examples, and other combinations are taken as negative examples. However, as indicated in Section 5.1.2, several action candidates can be reflective response actions to one ambiguous user request. Since complete annotation to any possible system action is costly, we applied positive/unlabeled (PU) learning to consider the data

Table 5.6.: Frequent pairs of pre-defined and additional categories: User requests in Japanese are translated into English.

Pre-defined category	Added category	Frequency	Example user request
map	*browser*	20	Is XX within walking distance?
fall colors	*nature/ landscape*	20	I'd like a place that feels like autumn.
shaved ice	*cafe*	20	I'm going to get heatstroke.
french	*expensive restaurant*	20	I'm having a luxurious meal today!
Kyoto cuisine	*tea-ceremony dishes*	20	I'd like to try some traditional Japanese food.

properties; one action is annotated as a reflective response to one ambiguous user request, but the labels of other system actions are not explicitly decided. Moreover, some reflective action selections are based on causal relations between a user's request and a system's action. For example, suggesting a *vegetarian restaurant* to a person who is on a diet is convincing because of its causal relation. Thus, in this section, we first describe the problem of reflective action classification by conventional PN learning. Then we introduce PU learning objectives and re-scoring based on a causality detection model. Figure 5.4 overviews the whole process and the corresponding sections.

5.2.1. Classifier

Figure 5.5 overviews the classification model, which classifies ambiguous user requests into reflective action (positive example) categories for the dialogue agent. We made a [CLS] vector (distributed representation) of a user request by Bidirectional Encoder Representations from the Transformers (BERT) [28] and used it as input for a multi-layer perceptron (MLP) with single hidden layer. MLP

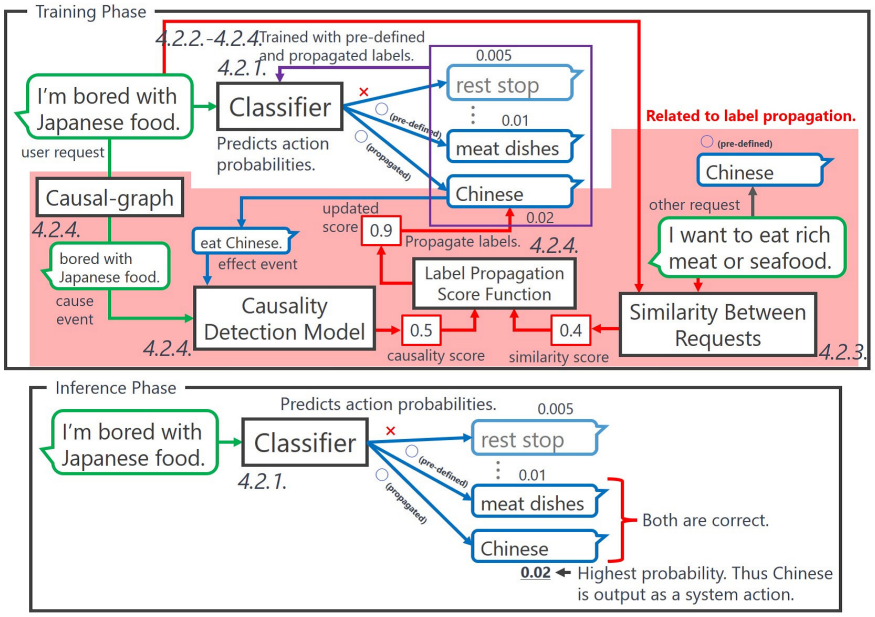


Figure 5.4.: Overview of whole process

calculates the probabilities associated with each action category. If the classifier is used as an actual dialogue agent, the agent selects the action category with the highest probability (confidence) as its action. In Figure 5.5, *camera* category will be selected as the dialogue agent's action because it has the highest probability.

5.2.2. Loss Function in PN Learning

When we build a classifier based on PN learning, the following loss function [20] is used to train the model:

$$\begin{aligned} Loss \;=\; & \sum_{i}^{|U_{train}|} \sum_{j=1}^{|C_{x_i}^{+}|} \sum_{k=1}^{|C_{x_i}^{-}|} L(r_j) R_s(\mathbf{w}_j^{\mathsf{T}}\mathbf{x}_i - \mathbf{w}_k^{\mathsf{T}}\mathbf{x}_i) \\ & + \kappa \sum_{i}^{|U_{train}|} \sum_{j=1}^{|C|} R_s(y_{ij}(\mathbf{w}_j^{\mathsf{T}}\mathbf{x}_i)). \end{aligned} \tag{5.1}$$

U_{train} is the set of user requests in the training data. $C_{x_i}^{+}$ and $C_{x_i}^{-}$ are the sets of the positive example action categories associated with user request x_i and the action categories without any annotation. r_j is the rank predicted by the model

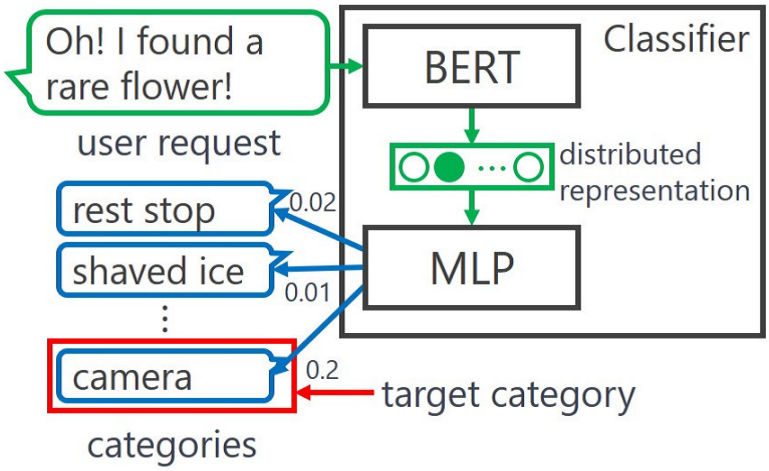


Figure 5.5.: User request classifier

for positive category j, and $L(r_j)$ is the weight function satisfying the following equation:

$$L(r_j) \quad = \quad \sum_{k=1}^{r_j} \frac{1}{k}. \tag{5.2}$$

Eq. (5.2) takes a larger value when the predicted rank is far from the top value. $\mathbf{w}_j$ is the weight vector corresponding to category j. $\mathbf{x}_i$ is the distributed representation corresponding to user request x_i. $R_s(y_{ij}(\mathbf{w}_j^\mathsf{T}\mathbf{x}_i))$ is the ramp loss:

$$R_s(y_{ij}(\mathbf{w}_j^\mathsf{T}\mathbf{x}_i)) \quad = \quad \min(1-m, \max(0, 1-y_{ij}(\mathbf{w}_j^\mathsf{T}\mathbf{x}_i))). \tag{5.3}$$

m is a hyperparameter that determines the classification boundary. Let C be a set of defined categories, where $|C| = 70$ in our dataset. y_{ij} is 1 if category j is a positive example for user request x_i and -1 if it is not annotated. κ is a hyperparameter that represents the weight of the second term.

5.2.3. Loss Function in PU Learning

Although the loss function of PN learning treats all combinations of unlabeled user requests and system action categories as negative examples, about 10% of

these combinations must be treated as positive examples in our corpus, as investigated in Section 5.1.2. To consider the data properties, we applied PU learning [32], which is an effective method for problems that are difficult to annotate completely, such as object recognition in images with various objects [47].

We used a PU learning method proposed by Cevikalp et al. [20], which is based on label propagation [21, 116]. It propagates the labels of annotated samples to unlabeled samples using the distance between the samples on a distributed representation space. The original method [20] propagated labels from the nearest neighbor samples on the distributed representation space. The method calculates similarity score s_{ij} of the propagated labels (categories) as follows:

$$s_{ij} = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)}{\bar{d}} \cdot \frac{|C|}{|C|-1}\right). \tag{5.4}$$

$\mathbf{x}_j$ is the vector of the distributed representations of the nearest neighbor user request whose category j is a positive example. $d(\mathbf{x}_i, \mathbf{x}_j)$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\bar{d}$ is the mean of all the distances. The value range of s_{ij} is $0 \leq s_{ij} \leq 1$. It takes larger values when the Euclidean distance between two distributed representations becomes smaller. We call this method: *PU-nearest*.

However, since the original method is sensitive to outliers, we propose a method that uses the mean vectors of the user requests with the same category. It propagates labels based on the distances between these mean vectors. We update similarity score s_{ij} in Eq. (5.4):

$$s_{ij} = \exp\left(-\frac{d(\mathbf{x}_i, \bar{\mathbf{x}_j})}{\bar{d}} \cdot \frac{|C|}{|C|-1}\right). \tag{5.5}$$

$\bar{\mathbf{x}_j}$ is the mean vector of the distributed representations of the user requests whose category j is a positive example. We call this method *PU-mean*. The proposed method scales similarity score s_{ij} to a range of $-1 \leq s_{ij} \leq 1$ using the following formula:

$$\tilde{s_{ij}} = -1 + \frac{2(s_{ij} - \min(s))}{\max(s) - \min(s)}. \tag{5.6}$$

s is the set of all similarity scores. If scaled score $\tilde{s_{ij}}$ is $0 \leq \tilde{s_{ij}} \leq 1$, we add category j to $C^+_{x_i}$ and let $\tilde{s_{ij}}$ be the weight of category j as a positive category.

If $\tilde{s_{ij}}$ is $-1 \leq \tilde{s_{ij}} < 0$, category j is assigned a negative label, and the weight is set to $-\tilde{s_{ij}}$. Using similarity score $\tilde{s_{ij}}$, we update Eq. (5.1):

$$
\begin{aligned}
&Loss = \\
&\sum_{i}^{|U_{train}|} \sum_{j=1}^{|C_{x_i}^{+}|} \sum_{k=1}^{|C_{x_i}^{-}|} \tilde{s_{ij}} \tilde{s_{ik}} L(r_j) R_s(\mathbf{w}_j^{\mathsf{T}} \mathbf{x}_i - \mathbf{w}_k^{\mathsf{T}} \mathbf{x}_i) \\
&+\kappa \sum_{i}^{|U_{train}|} \sum_{j=1}^{|C|} \tilde{s_{ij}} R_s(y_{ij}(\mathbf{w}_j^{\mathsf{T}} \mathbf{x}_i)). \qquad (5.7)
\end{aligned}
$$

In Eq. (5.7), $\tilde{s_{ij}}$ is a weight representing the contribution of the propagated category to the loss function. Similarity score $\tilde{s_{ij}}$ of the annotated samples is set to 1.

5.2.4. Adjusting Similarity Scores with Causality Score

The label propagation method in Section 5.2.3 propagates labels based only on the similarity between user requests with different action categories. We expect to achieve more accurate label propagation by taking into account what actions users typically choose when they make certain requests. To distill such knowledge, we introduce a causality detection model that regards user requests as causes and action categories as effects. For example, given a pair *bored with Japanese food* → *eat meat dishes*, the model regards the former as a cause and the latter as an effect. We utilized causality scores computed by the causality detection model as a new feature in label propagation.

Figure 5.6 overviews the causality detection model that outputs a causality score. First, BERT converts a cause and an effect and concatenates them with a [SEP] token to a real-valued vector. Then MLP calculates a real-valued causality score. A larger causality score indicates a stronger causal relation between the cause and the effect of the input. In general, causality represents the relationship between single events. If all the text in a user request is input as a cause, the causality score cannot be calculated precisely because multiple events are included in requests. Thus we use causal-graph [117] to analyze the predicate-argument structures contained in user requests. The analyzed predicate-argument structures are input to BERT as the basic units of events. A causal-graph is a tool that extracts causalities in sentences based on such linguistic features as "because."

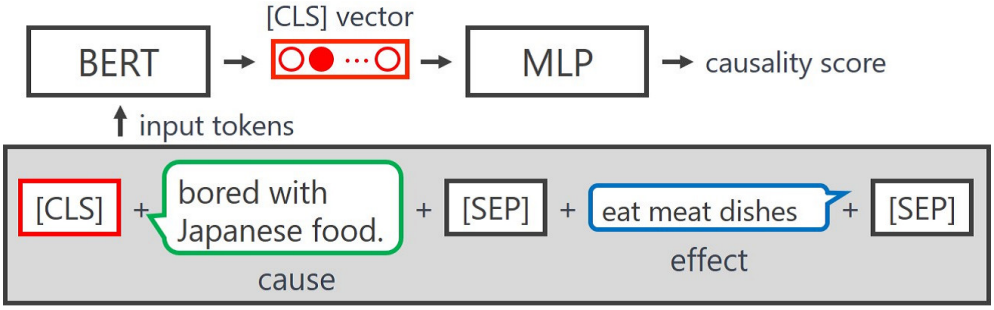


Figure 5.6.: Causality detection model

We used the causal-graph to extract events that will have causal relations. As preprocessing, the causal-graph extracts the predicate-argument structures from the given sentences. We identified the events in user requests by inputting them into the causal-graph to extract the events in them. In our study, the events in user requests denote the predicate-argument structures extracted from user requests. The analysis results of the causal-graph includes the surface representations of predicate-argument structures that have causal relations. Our proposed method concatenates these surface representations and inputs them to BERT as sentences representing events. On the other hand, to identify the events of the action categories that are input as effects, we manually defined the predicate-argument structures in which the user is the subject, such as *eat meat dishes* when the category is *meat dishes*. The causality detection model is inspired by the Next Sentence Prediction of BERT pre-training [28].

We trained the causality detection model using the margin ranking loss in the following equation, which treats pairs of user requests and labeled categories as positive examples ${cx_i}^+$ and pairs of user requests and unlabeled categories as negative examples ${cx_i}^-$:

$$Loss({cx_i}^+, {cx_i}^-) \quad = \quad \max\Big(0, -({c_i}^+ - {c_i}^-) + 0.5\Big). \tag{5.8}$$

${c_i}^+$ and ${c_i}^-$ are the causality scores output by the model for positive and negative examples. Eq. (5.8) is 0 for ${c_i}^+ - {c_i}^- \geq 0.5$.

We updated similarity score $\tilde{s_{ij}}$ in Eq. (5.7) using the trained causality detection model. First, it calculated c_{ij}, which are the causality scores for all the

pairs of the events in user requests x_i and unlabeled categories j. When multiple events are included in a user request, the event with the highest causality score is treated as user request x_i. Next the causality scores are scaled to a range of $-1 \leq c_{ij} \leq 1$ using Eq. (5.6). Then we updated the similarity score:

$$\hat{s_{ij}} \quad = \quad \max(\min(\tilde{s_{ij}} + c_{ij} - \gamma, 1), -1). \tag{5.9}$$

γ is a hyperparameter (margin) for adjusting the effect of the causality score. The value range of γ is $0.0 \leq \gamma \leq 1.0$. We expect that the model propagates labels based not only on similarities between user requests but also on causality relations between user requests and system actions by introducing this equation. Updated similarity score $\hat{s_{ij}}$ is introduced to Eq. (5.7) to train the classification model. We call this method *PU-causal*. We defined *PU-margin* as the similarity score updated in Eq. (5.9) in which causality score c_{ij} is removed. In other words, *PU-margin* is *PU-nearest* or *PU-mean* with a simple margin adjustment for the similarity score.

5.3. Experiments

We evaluated the models described in Section 5.2, which classified the user requests into corresponding action categories.

5.3.1. Model Configuration

We implemented our models using PyTorch [85] and the Japanese BERT model [92], which was pre-trained on Wikipedia articles.

We used Adam [52] to optimize the model parameters and set the learning rate to 1e−5. For m in Eq. (5.3) and κ in Eq. (5.1), we set $m = -0.8, \kappa = 5$ based on the literature [20]. We used the distributed representations output by BERT as vector $\mathbf{x}_i$ in the label propagation. We pre-trained the model by PN learning before we applied PU learning. Similarity score s_{ij} of *PU-nearest* was also scaled by Eq. (5.6) as with *PU-mean*. We used the training data of the collected corpus in Section 5.1.1 as the training, validation, and test data for the causality detection model. The causalities extracted from the user requests by the

causal-graph were also used for training. The accuracy of the trained causality detection model is 91.70. We respectively set γ in Eq. (5.9) to 0.8, 0.6, 0.4, and 1.0 for *PU-nearest-margin*, *PU-nearest-causal*, *PU-mean-margin*, and *PU-mean-causal*. The hyperparameters of each model used in the experiments were determined by the validation data.

5.3.2. Evaluation Metrics

Accuracy, Recall@5 (R@5), and Mean Reciprocal Rank (MRR) were used as evaluation metrics. R@5 counts the ratio of the test samples that have at least one correct answer category in their top five. MRR ($0 < MRR \leq 1$) is calculated as follows:

$$MRR = \frac{1}{|U_{test}|} \sum_{i}^{|U_{test}|} \frac{1}{r_{x_i}}. \tag{5.10}$$

r_{x_i} denotes the rank output by the classification model for the correct answer category corresponding to user request x_i. U_{test} is the set of user requests included in the test data. If multiple action categories correspond to user request x_i as correct categories, the highest rank among the correct categories is regarded as the rank of the correct categories output by the model. All the metrics are calculated based on these ranks of correct categories. For all the metrics, a higher value means a better performance for the classification model. We calculated the performance of each model by averaging a hundred trials. For the test data, the correct action categories were annotated completely, as shown in Section 5.1.2; thus, multi-label scores were calculated for each model.

5.3.3. Reflective Action Classification Performance

We compared the model performances trained with the methods described in Section 5.2 on the test data. The experimental results are shown in Table 5.7. *PN* is the scores of the PN learning method (Section 5.2.2) and *PU* is the scores of the PU learning methods (Section 5.2.3). *Nearest* denotes the label propagation just considering the nearest neighbor samples in the distributed representation space. *Mean* denotes the proposed label propagation using the mean vector of each category. *Causal* denotes the updates of the similarity scores with the causality

Table 5.7.: Classification results as averages of 100 trials. We conducted paired T-test between PN and PU learning methods. † means $p < 0.05$. †† means $p < 0.01$.

Model	Accuracy (%)	R@5 (%)	MRR
PN	89.04 (±0.58)	98.10 (±0.27)	0.9304 (±0.0035)
PU-nearest	88.40 (±0.77)	97.88 (±0.28)	0.9254 (±0.0049)
PU-nearest-margin	††89.61 (±0.66)	††98.21 (±0.23)	††0.9341 (±0.0039)
PU-nearest-causal	††89.76 (±0.67)	††98.25 (±0.24)	††0.9344 (±0.0042)
PU-mean	††89.28 (±0.72)	97.87 (±0.27)	0.9305 (±0.0047)
PU-mean-margin	††89.99 (±0.61)	††98.26 (±0.22)	††0.9359 (±0.0036)
PU-mean-causal	††90.05 (±0.53)	††98.27 (±0.22)	††0.9366 (±0.0032)

scores. *Margin* denotes the updates of the similarity scores only with γ in Eq. (5.9). For each model, a paired t-test was used for a significance test in the performance from the baseline (PN). † and †† mean that $p < 0.05$, and $p < 0.01$ for a significant improvement in performance.

Each system achieved more than 88 points for accuracy and 97 points for R@5. The proposed methods (*PU-nearest-margin*, *PU-nearest-causal*, *PU-mean*, *PU-mean-margin* and *PU-mean-causal*) achieved significant improvement over the baseline method (*PN*); not even the existing PU-based method (*PU-nearest*) saw such a level of improvement. The improvements on R@5 were small, suggesting that most of the correct samples are already included in the top five, even in the baseline. We calculated the ratio of the "positive categories predicted by the PU learning model in the first place that are included in the positive categories predicted by the PN learning model in the second through fifth places" when the following conditions were satisfied: "the PN learning model does not predict any positive category in the top place," "the PN learning model predicts some positive categories in the second through fifth places," and "the PU learning model predicts a positive category in the first place." Figure 5.7 visualizes this investigation. The percentage is 97.35 (±2.85)%, which supports our hypothesis for R@5: "most of the correct samples are already included in the top five, even

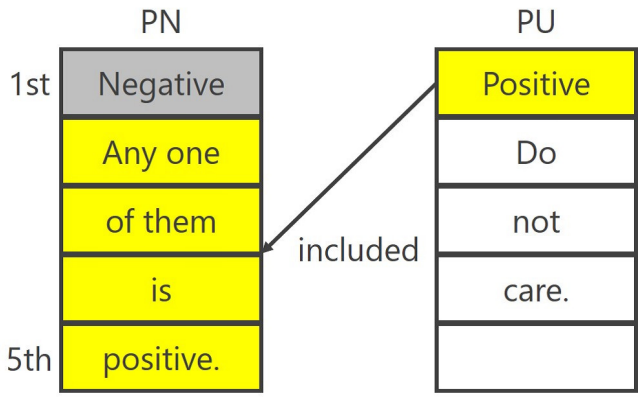


Figure 5.7.: Visualization of investigation for R@5

Table 5.8.: T-test results between PU learning methods. † means $p < 0.05$. †† means $p < 0.01$.

Model	v.s.	Accuracy	R@5	MRR
nearest-margin	*nearest*	††	††	††
nearest-causal	*nearest*	††	††	††
	nearest-margin	†		
mean	*nearest*	††		††
mean-margin	*mean*	††	††	††
	nearest-margin	††	†	††
mean-causal	*mean*	††	††	††
	mean-margin			†
	nearest-causal	††		††

in the baseline."

Detailed Analysis of Classification

Table 5.8 shows the paired T-test results between the PU learning methods. Note that these T-test results only show significant differences between two different

Table 5.9.: Frequent misclassification

Rank	Pre-defined category	# Misclassifications
1	*browser*	7.17 (±1.17)
2	*average-priced restaurant*	6.18 (±1.28)
3	*transfer navigation*	4.15 (±0.70)
4	*tea-ceremony dishes*	4.35 (±1.18)
5	*phone*	4.15 (±0.70)

PU learning methods, not for the ordinal associations among all of them. The *PU-mean* performance is significantly better than the *PU-nearest* on accuracy and MRR, and the *PU-margin* and *PU-causal* performances are significantly better than the *PU-nearest* or the *PU-mean*. In addition, the performance is significantly better on accuracy or MRR when the causality scores are also used in *PU-causal* rather than updating the similarity scores using only the margin.

Table 5.9 shows the frequency of misclassification for each action category. The number of misclassifications is calculated as the average of all the *PU-mean-causal* trials. The results show that the most difficult category was *browser*, which is a common action category for any user request.

Figure 5.8 shows the accuracy for each test sample divided by the number of action categories added in Section 5.1.2. Each horizontal tick represents the data whose number of additional action categories is greater than or equal to the number indicated on the tick and less than the tick on the right. We set the ticks so that the amount of data in each tick is approximately the same. For all the models, the more additional action categories were added, the higher the accuracy became. This result agrees with our intuition that as more action categories are regarded as reflective, appropriate responding becomes easier.

5.3.4. Label Propagation Performance

We evaluated the performance of label propagation itself on the test data to verify its effect in PU learning. Table 5.10 shows the results. In this evaluation, the label propagation method propagates the predefined action categories to the

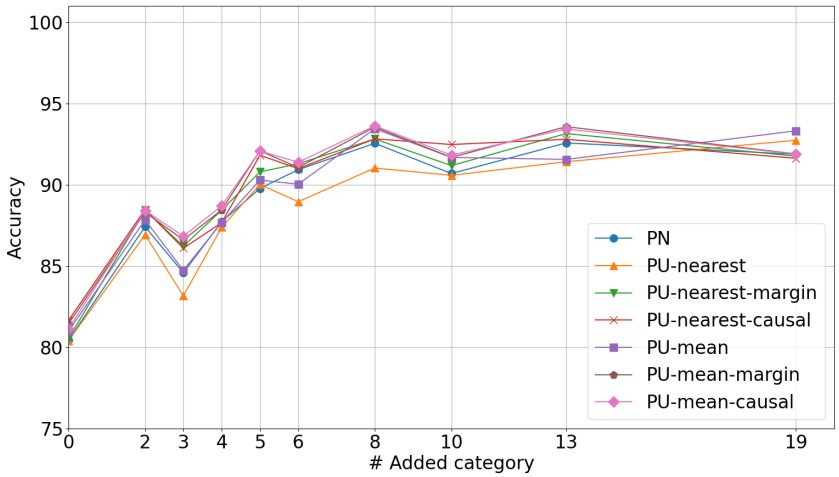


Figure 5.8.: Accuracy for each # added category

Table 5.10.: Label propagation performance

Model	Pre. (%)	Rec. (%)	F1
nearest	67.56 (±2.11)	7.78 (±1.05)	0.1392 (±0.0164)
nearest-margin	88.46 (±1.48)	0.13 (±0.02)	0.0026 (±0.0004)
nearest-causal	90.66 (±1.39)	3.02 (±0.22)	0.0584 (±0.0042)
mean	62.19 (±4.96)	13.51 (±2.31)	0.2200 (±0.0274)
mean-margin	95.78 (±1.34)	0.75 (±0.27)	0.0148 (±0.0053)
mean-causal	98.03 (±0.93)	0.61 (±0.18)	0.0121 (±0.0035)

user requests in the test data. We investigated whether the propagated action categories are included in the added action categories. The label propagation's precision denotes the percentage of the propagated action categories which are included in the added action categories. The label propagation's recall denotes the percentage of the added action categories which are included in the propagated action categories. Comparing Tables 5.7 and 5.10, in general, the higher the precision of the label propagation is, the higher the model's performance. The methods using causality scores have higher precision than methods using only

Table 5.11.: Similarity scores of label propagation

Model	True-positive	False-positive
nearest	267.80 (±41.41)	96.16 (±31.24)
nearest-margin	8.44 (±0.48)	1.17 (±0.06)
nearest-causal	99.13 (±8.28)	6.43 (±1.26)
mean	327.19 (±79.34)	132.41 (±60.75)
mean-margin	34.08 (±17.64)	2.30 (±2.15)
mean-causal	123.51 (±26.11)	11.20 (±4.86)

similarity scores between user requests. The highest precision was about 98%. Although we conclude that label propagation added reflective action categories as positive examples with high precision, there is still room to improve their recalls.

Detailed Analysis of Label Propagation

We investigated the effect of margin γ on the label propagation. Figure 5.9 shows the precision-recall curve for the validation data with the margin changed from 0 to 1.0 by 0.1. The higher the recall is, the smaller γ is. The curve shows that if we decrease γ to increase the recall, the precision decreases significantly. Increasing the recall while maintaining the precision is difficult in any method. Considering the results of the reflective action classification, we conclude that even if the recall increases, the decrease of precision leads to a degradation of label propagation and thus decreases the classification accuracy.

We investigated why *PU-mean-causal* showed the highest classification performance for reflective actions in Table 5.7, even though it showed a low label propagation recall in Table 5.10. Table 5.11 shows the similarity scores of true-positive and false-positive in the label propagations. *PU-mean-causal* had a larger ratio between the true-positive and false-positive similarity scores than *PU-mean*. In addition, comparing *PU-mean-causal* with *PU-mean*, the true-positive similarity scores decreased more gradually than the recall in Table 5.10. These results indicate that *PU-mean-causal* has a true-positive effect, which is larger than the values shown in the recall of the label propagation, and a false-positive effect,

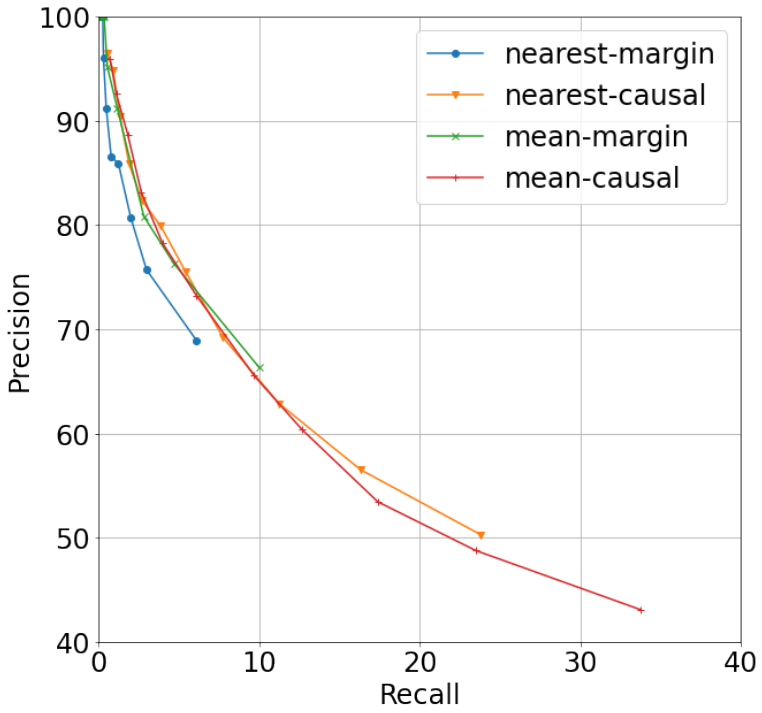


Figure 5.9.: Precision-recall curve for varying margin γ on validation data: Results are averages of ten trials. The higher the recall, the smaller γ is.

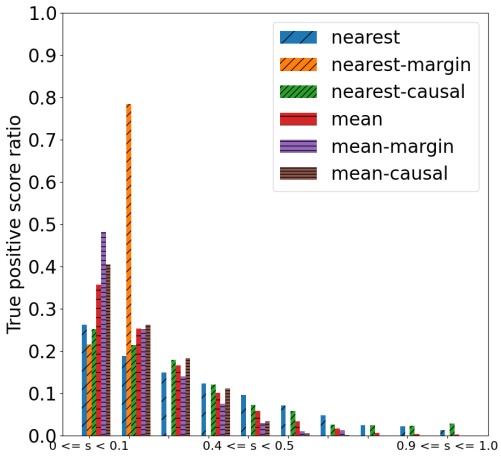


Figure 5.10.: True positive similarity score ratio: Each x scale represents a range of similarity scores of 0.1.

which is smaller than *PU-mean*, on the loss function. Therefore, *PU-mean-causal* showed the highest performance in the action classification experiment in Table 5.7.

The distributions of the similarity scores for true positive and false positive label propagation are shown in Figs. 5.10 and 5.11. In both distributions, *PU-mean* tends to estimate the scores lower than *PU-nearest*. In other words, *PU-mean* has less negative impact on the loss function of Eq. (5.7) in cases of incorrect label propagation. Therefore, *PU-mean* outperformed *PU-nearest* in Tables 5.7 and 5.8.

Table 5.12 shows examples where the label propagation failed. *Nearest request* is the nearest neighbor of *original request* among the requests labeled as *propagated category* as a positive example. Comparing *nearest request* and *original request* in Table 5.12, label propagation was incorrect when the sentence intentions are completely different or when two requests contain similar words. The sentence intentions are altered by negative forms or other factors. Table 5.13 shows examples of incorrect label propagations of *PU-mean-margin*, which are not included in those of *PU-mean-causal*. As in Table 5.12, label propagation

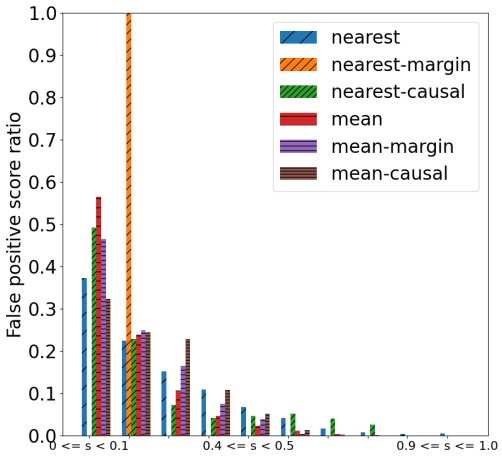


Figure 5.11.: False positive similarity score ratio: Each x scale represents a range of similarity scores of 0.1.

Table 5.12.: Examples of incorrect label propagations of *PU-mean-causal*

Original request	Pre-defined category	Nearest request	Propagated category
I want to eat rich meat and seafood.	*Chinese*	I can't eat meat.	*vegetarian restaurant*
I wonder if it'll be hot again tomorrow.	*weather*	It's hot today.	*shaved ice*
I'll make some plans for tomorrow morning.	*breakfast*	I wonder if I can wake up early enough tomorrow morning.	*alarm*

is incorrect when similar words are included in both requests. According to Table 5.10, *PU-mean-causal* makes fewer such mistakes because it uses causality features, resulting in higher precision of label propagation.

Table 5.14 shows the error ratios in the label propagation between the functions.

Table 5.13.: Examples of incorrect label propagations of *PU-mean-margin* that are not included in incorrect label propagations of *PU-mean-causal*

Original request	Pre-defined category	Nearest request	Propagated category
Look at those walking penguins!	*camera*	The kids want to see penguins.	*aquarium*
I'd like to do something other than shopping.	*beautiful place*	I want to go shopping.	*shopping mall*
I wonder which souvenir would be better.	*message*	What are the most popular souvenirs?	*Japanese sweets*

Table 5.14.: Ratios of false positive in label propagation of *PU-mean-causal*

spot search	spot search	3.83 (±14.57)
	restaurant search	0.00 (±0.00)
	app launch	0.00 (±0.00)
restaurant search	spot search	0.00 (±0.00)
	restaurant search	19.90 (±30.87)
	app launch	0.78 (±4.58)
app launch	spot search	1.25 (±10.28)
	restaurant search	70.45 (±36.72)
	app launch	0.78 (±4.58)

More than 70% of the label propagation errors happened between *app launch* and *restaurant search*. This is because many trials include the incorrect label propagation of the second example in Table 5.12.

5.4. Conclusion

We collected a dialogue corpus that bridges ambiguous user requests to reflective system actions while focusing on system action functions (API calls). We asked crowd-workers to input antecedent user requests for which pre-defined dialogue agent actions can be regarded as reflective. We also constructed test data as a multi-class classification problem, assuming cases in which multiple action candidates are qualified as reflective for ambiguous user requests. With the collected corpus, we developed classifiers that sort ambiguous user requests into the corresponding categories of reflective system actions. Our proposed PU learning method achieved high accuracy on the test data, even when the model was trained on incomplete training data as in multi-class classification tasks. In addition, we revealed that the performance of PU learning improved with causality scores, which represent whether the user requests and the action categories are connected as causalities.

Our future work will investigate model architecture to improve classification performance, especially the performance of label propagation. We will also investigate the features of user requests that are difficult to classify.

6. Reflective Action Selection for Domestic Robot Utilizing Multimodal Features of Ambiguous User Requests and Surrounding Situations

This chapter investigates the dialogue system that takes reflective actions for ambiguous user requests using multimodal information. First, we construct a multimodal corpus including images that represent situations surrounding the user based on the corpus collection method in Chap. 5. In addition, we assign descriptive labels as events that describe the surrounding situations of the user. Next, we train baseline models consisting of existing pre-trained models using the constructed dataset. Finally, we prove the validity of utilizing multimodal information to learn reflective action selection with a small dataset by comparing various baseline models.

6.1. Task Definition and Dataset Collection

Taylor [98, 99] categorized user states in information search into four levels by their clarity as shown in Table 1.1. Most existing task-oriented dialogue systems [69, 102] or robots [7] convert explicit user requests (Q3) into machine readable expressions (Q4). Future dialogue systems need to proactively take appropriate actions even in situations such as Q1 and Q2, where the users fail to clearly verbalize their requests [110]. We used crowdsourcing to collect user situations

Figure 6.1.: Example of reflective interaction: Robot bringing a banana

in which users need proactive support by the robot.

This study deals with a task where an interactive robot helps a user in a typical living room and kitchen. When a user makes an ambiguous request or monologue, the robot responds with reflective actions based on the surrounding situation as context. Figure 6.1 shows an example interaction between a user and a robot. The user's utterance, "I want breakfast," does not represent a call for robot action. The robot selects a reflective action, *brings a banana (to the user)*, confirms the request with the user by asking "Would you like a banana?," and finally brings the banana to her since there is one at the dining table. The following sections describe the detailed definitions of the task and the data construction methods.

6.1.1. Task Definition: Reflective Action Selection

This work's final goal is to output reflective actions when the robot hears ambiguous utterances, based on the surrounding observable situations. Thus, we defined our dataset as a triplet of ambiguous user utterance, an image that assumes the robot's first-person viewpoint representing the situation uttered by the user, and the robot's corresponding reflective action. When an ambiguous request is input, the robot also considers the situational elements as inputs to output a reflective

Table 6.1.: Action categories of home mobile manipulator

bring a banana, bring a charging cable, bring a cup, bring ketchup, bring a package, bring a plastic bottle, bring the remote control, bring the smartphone, bring a snack, bring a tissue box, put the charging cable away, put the cup away, put the ketchup away, put the toy car away, put the plastic bottle away, put the remote control away, put the smartphone away, put the snack away, put the tissue box away, throw the trash away, bring a can opener, bring a cooking sheet, bring a glass, bring a grater, bring some kitchen paper, bring a lemon, bring some olive oil, bring a potato, bring some Saran wrap, bring a water bottle, put the can opener away, put the cooking sheet away, put the glass away, put the grater away, put the kitchen paper away, put the plastic bottle in the refrigerator, put the Saran wrap away, put the Tupperware in the microwave, put the Tupperware in the refrigerator, put the water bottle away

action category from pre-defined action categories. We used Stretch RE1 [*,†] as a robot that helps users. Stretch RE1 is a home mobile manipulator equipped with a camera and a robotic arm. Its robotic arm's load capacity is 1.5 kg. We defined the robot action categories based on the capabilities of Stretch RE1. Table 6.1 lists the 40 pre-defined action categories. Almost all of the action categories are either *bring* or *put away* as shown in Table 6.1. For instance, when the utterance and the situation in Figure 6.1 are input, the robot is regarded as reflective if it selects action category *bring a banana* from the 40 categories.

6.1.2. Collecting Ambiguous User Requests and Reflective Actions of a Robot

The Wizard-of-Oz (WOZ) method [16, 48], in which two subjects play the roles of a user and an interactive robot, is widely used to collect dialogue samples. However, even humans sometimes struggle to respond with reflective actions to

[*]https://hello-robot.com/

[†]https://spectrum.ieee.org/hello-robots-stretch-mobile-manipulator

ambiguous user requests. In other words, a general WOZ dialogue is inappropriate for collecting such interactions that contain reflective actions. Therefore, we collected a corpus consisting of situation-action pairs by asking crowd-workers to input antecedent ambiguous user requests and indoor situations which the pre-defined robot-action categories can be regarded as reflective. Crowd-workers input the detailed situations of the users in addition to their utterances. For instance, the worker inputs a detailed situation: "The user looked for another glass and said, 'There isn't another one' when he is talking about drinking and getting ready for it." This corresponds to a pre-defined action *bring a glass*. The workers were asked not to input clear requests that include both a predicate and an object of a robot action such as "Please bring another glass" to make sure that the collected situations do not include clear user requests. As shown in Figure 6.1, we presented videos of the robot performing defined actions to facilitate worker understanding of the robot actions. We used crowdsourcing[‡] to collect a Japanese corpus based on the robot-action categories defined in Table 6.1.

6.1.3. Annotating Multimodal Features

When the robot receives an ambiguous user request, it often has problems in selecting an appropriate reflective action based solely on such utterances. For example, the corpus collected in Section 6.1.2 includes the following utterance: "There isn't another one." The user made the utterance because he wanted another glass while handling a glass and an alcohol bottle. The robot is unable to select the appropriate action *bring a glass* based solely on the user utterance. If the robot could also recognize images of situations associated with its collected user utterances, it could understand that the user has a glass while standing in the kitchen, where there could be another glass the user might need. We collected these situations occurring in a living room or kitchen that correspond to the user utterances.

In this study, we face a difficulty to collect a large amount of data because we use images that assume a robot's first-person viewpoints in certain environments. Abstracting the dataset is critical for effectively using such a small amount of data as training data [112]. Feature extraction methods, with pre-trained models

[‡]https://crowdworks.jp/

trained on large-scale data, have been widely used in recent years; however, more focused information is required to understand visual situations. To investigate this assumption, we designed several features as the descriptions of images and manually annotated them. We clipped the last frames of the videos as the representative images and labeled descriptive features such as objects or user poses. In the following, we describe the details of the annotation.

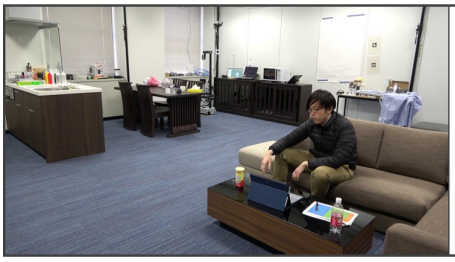
Figure 6.2 shows examples of the collected user utterances, the images associated with them, and the descriptive features obtained from the images. *Uttr* and *action* denote the user utterances and the corresponding robot actions. *Viewpoint* denotes the perspective number of the camera from which the image was taken. There are three such viewpoints. *Position* describes the location of the user in the room, such as on the sofa or in the kitchen. *Pose* describes the user's posture, such as sitting or standing. *Has* describes the objects being held by the user. *Coffee table* describes objects on a small table in the living room. Similarly, we also defined this feature for *kitchen* and *dining table*. Although these features were annotated manually, we expect that it will be possible to automatically extract them using such machine learning models as image recognition in the future. All of the images contain a coffee table, a dining table, and a kitchen. Figure 6.2 shows that we collected and annotated the images and the descriptive features that correspond to the user utterances in which the pre-defined robot actions can be regarded as reflective. Table 6.2 shows the statistics of the collected corpus. We gathered only 400 samples because collecting the explained data requires a great cost. The available dataset is extremely small compared to a typical dialogue corpus [16].

6.2. Baseline Reflective Action Classifier Using Multimodal Features

Using the dataset described in Section 6.1, we built a model that selects reflective actions to the given ambiguous user utterances and corresponding situations. A text-based model, which just processes user utterances, can be developed if we use the corpus collected in Section 6.1.2. However, as indicated in Section 6.1.3, the model will probably often misclassify actions due to its ambiguity.

uttr : ``Thank you for the meal."
action : put the ketchup away
viewpoint : 2
position : sofa
pose : sitting
has : (empty)
coffee table: ketchup, laptop, etc.
dining table: olive oil, kitchen paper, etc.
kitchen : plastic bottle

(a)

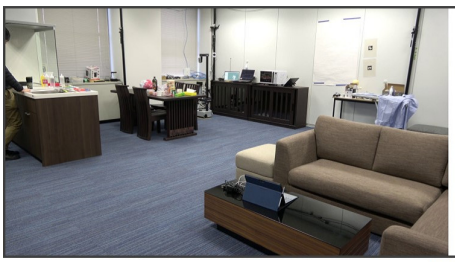


uttr : `` I finished eating. ''
action : put the snack away
viewpoint : 0
position : sofa
pose : sitting
has : (empty)
coffee table: snack, laptop, plastic bottle
dining table: cup, charging cable, etc.
kitchen : ketchup, olive oil, etc.

(b)

uttr : `` There isn't another one. ''
action : bring a glass
viewpoint : 1
position : kitchen
pose : standing
has : glass, alcohol bottle
coffee table: (empty)
dining table: snack, cup, etc.
kitchen : (empty)

(c)



uttr : ``I'm off tomorrow."
action : put the water bottle away
viewpoint : 0
position : kitchen
pose : standing
has : (empty)
coffee table: charging cable, laptop
dining table: olive oil, cup, etc.
kitchen : water bottle, plastic bottle

(d)

Figure 6.2.: Examples of collected interactions. Texts were translated into English.

We also can use recent pre-trained models as feature extractors from both the utterance and the image [63,97] to utilize multimodal features. However, we need

Table 6.2.: Statistics of collected corpus. # denotes the number of objects being held by the user or placed on the coffee table, on the dining table, or in the kitchen.

# interactions	400
Ave. utterance length	11.59 (±4.94)
Ave. # has	0.14 (±0.37)
Ave. # coffee table	1.40 (±0.91)
Ave. # dining table	4.32 (±1.95)
Ave. # kitchen	1.47 (±1.66)

to abstract information that can be extracted from the multimodal input when the model is trained with a small amount of training data for selecting robot action categories [112]. We investigated whether the classification accuracy could be improved by a baseline classifier that utilizes our descriptive features such as objects or user poses.

Figure 6.3 illustrates the overall architecture of the baseline multimodal classifier, which classifies ambiguous requests into forty reflective action categories by visual features. In the following, we describe the details of the processes for each feature and category prediction. We made a [CLS] vector (distributed representation of the whole of a sentence) of a user request by a pre-trained model, called RoBERTa [63], to use the context feature on which the request was made. We also made five [CLS] vectors (*position*, *has*, *coffee table*, *dining table*, and *kitchen*) because they are text-based features extracted from the images. The objects of each feature are concatenated with a [SEP] token and input into RoBERTa. For the viewpoint, a one-dimensional vector was retrieved from an embedding layer. These are descriptive features obtained from the images. In addition, we used EfficientNet-B0 [97], which is a widely used pre-trained model for images, to obtain a feature vector that represents the image itself. One of the baselines is a model that employs only user utterance features by RoBERTa and pre-trained image features by EfficientNet (*uttr+image*). Finally, the feature vectors obtained by these processes are all concatenated and input into a multi-layer perceptron

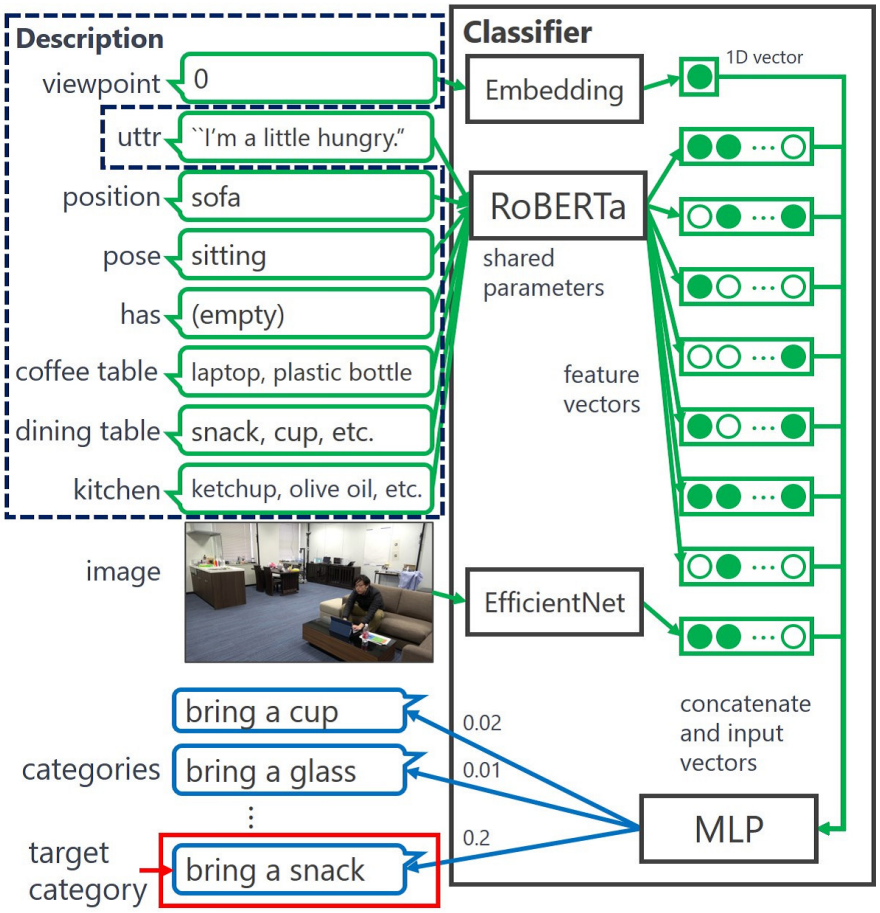


Figure 6.3.: Feature inputs for baseline classifier

(MLP) with single hidden layer to compute the probabilities corresponding to each category.

6.3. Experiments

We evaluated the models described in Section 6.2, which classified the given situations into corresponding action categories. Specifically, we evaluated whether the model could select reflective actions when the descriptive features were input. In the experiments, we compared a model that only convolutes the images with

a model that also utilizes the annotated descriptive features. In addition, the descriptive features should be automatically recognized to mount the baseline classifier on the actual robot. We constructed classifiers that automatically recognize the labels (user utterances, user poses, and object labels) and compared them with the baseline classifier that utilizes the manually annotated features.

6.3.1. Experimental Settings

We implemented our models using PyTorch [85] and the Japanese RoBERTa model [49], which was pre-trained on Japanese Wikipedia and the Japanese portion of CC-100. We converted the collected images, which were clipped from the videos assuming the robot's first-person viewpoint, to pre-trained image features with a pre-trained EfficientNet-B0 of PyTorch. The parameters of RoBERTa and EfficientNet were updated by the model training. When we build classifiers, a hinge loss function [20] is used to train the model. We used Adam [52] to optimize the model parameters and set the learning rate to 1e−5.

Accuracy, Recall@5 (R@5), and Mean Reciprocal Rank (MRR) were used as evaluation metrics. The hyperparameters of each model used in the experiments were determined by the validation data, which are different from the test data. We calculated the performance of each model by 5-fold cross-validation. Thus, the train data includes 240 interactions, and the valid and test data include 80 interactions in each trial. The experiments were run ten times for each bit of split data.

6.3.2. Reflective Action Classification Performance

The experimental results are shown in Table 6.3. *Uttr* denotes that the model uses only the user utterances from Figure 6.3. *Uttr+img* denotes that the model uses the image features output from EfficientNet in addition to the user utterances, which is the baseline without the descriptive features. *Uttr+img+desc* denotes that the model utilizes the annotated descriptive features obtained from the images in addition to the features used by *uttr+img*. A paired t-test was used for a significance test between the performances of *uttr+img+desc* and *uttr+img*. † and †† mean that $p < 0.05$ and $p < 0.01$, respectively, for a significant im-

Table 6.3.: Classification results as averages of 50 trials. We conducted paired T-test. †† means $p < 0.01$.

Model	Accuracy (%)	R@5 (%)	MRR
uttr	††27.02	††53.85	††0.4054
uttr+img	††27.23	††54.50	††0.4064
uttr+img+desc	63.58	87.12	0.7417

provement in performance of a paired t-test. *uttr+img+desc*, which utilizes the descriptive features obtained from the images, achieved significant improvement over the other baseline classifier *uttr+img*. These results suggest that extracting these descriptive features is an effective way to improve the accuracy of selecting reflective actions for the ambiguous requests. In other words, a robot can take reflective actions if it carefully looks at the surrounding situation of users.

6.3.3. Validity of Inputting Descriptive Features

Table 6.4 shows ablation studies of the descriptive features. *User* indicates the features of the user, including *position* and *pose* in Figure 6.3. *Object* indicates the features of the things in the images, including *has*, *coffee table*, *dining table*, and *kitchen* in Figure 6.3. A paired t-test was used for a significance test between the performances of each ablation and *uttr+img+desc*. Table 6.4 shows that performance decreases the most when *object* is removed, indicating that *object* is the most critical feature obtained from the images for selecting reflective actions that correspond to ambiguous requests and their situations. The performance slightly decreases or remains unchanged by removing the descriptive features other than *object*. Although these features do not seem critical, the t-test result between *w/o object* and *uttr+img* showed significant differences at $p < 0.01$, indicating that these features are also useful when *object* is unavailable. Using *viewpoint* or *user* is a better alternative for improving the classification accuracy with lower cost because annotating *object* is more costly than annotating *viewpoint* or *user*.

Figure 6.4 shows, except for (d), examples where *uttr+img* selected wrong ac-

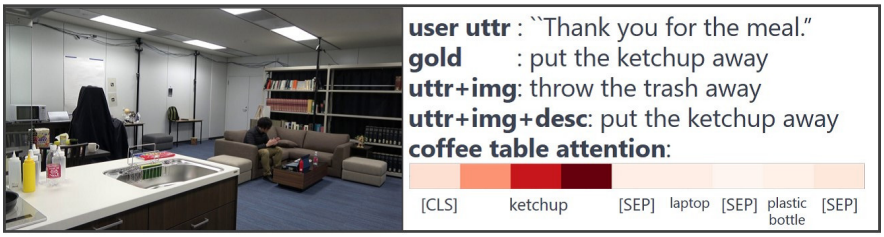
Table 6.4.: Classification results of ablation study. We conducted paired T-test. † means $p < 0.05$, and †† means $p < 0.01$.

Model	Accuracy (%)	R@5 (%)	MRR
uttr+img	27.23	54.50	0.4064
uttr+img+desc	63.58	87.12	0.7417
w/o viewpoint	††60.08	†85.62	††0.7132
w/o user	64.12	86.75	0.7424
w/o object	††31.38	††63.28	††0.4680
w/o image	††61.17	†85.75	††0.7231

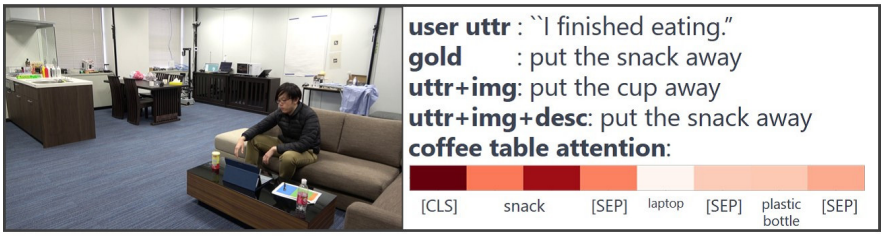
tions, although *uttr+img+desc* selected appropriate reflective ones. *Gold* denotes reflective actions associated with user utterances. *uttr+img* and *uttr+img+desc* denote actions selected by *uttr+img* and *uttr+img+desc*, respectively. We visualized the attention of the objects associated with the reflective actions among those involving *object*, which is the most critical descriptive feature. In the cases shown, *uttr+img+desc* selected appropriate actions in situations where *uttr+img* selected unrelated actions because *uttr+img+desc* focused attention on the objects associated with reflective actions. We conclude that *uttr+img+desc*, which utilizes descriptive features, successfully learned how to utilize them from a small amount of training data. However, as shown in (d) of Figure 6.4, *uttr+img+desc* occasionally fails to focus attention on objects that are effective for selecting appropriate reflective actions. We need further investigations to find the model architecture that can understand the user situations more precisely.

6.3.4. Automatic Feature Recognition

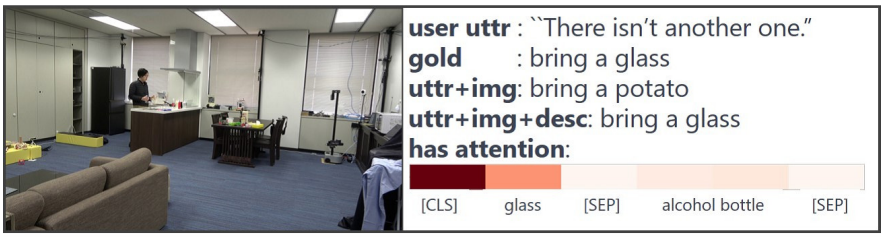
The baseline classifier of Section 6.2 assumes that the user utterances and the descriptive features are recognized with 100% accuracy. If we mount the classifier on the actual robot, the user utterances and the descriptive features should be automatically recognized. We investigated the performances of the classifier when



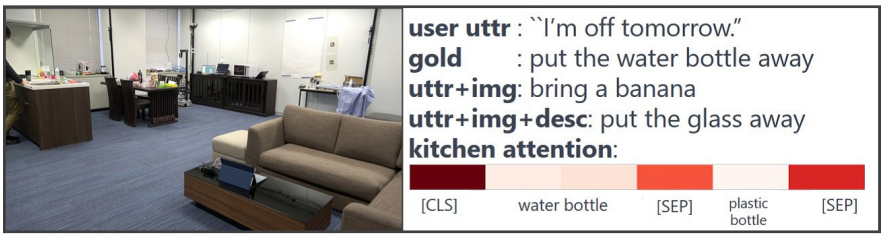
(a)



(b)



(c)



(d)

Figure 6.4.: Cases where multimodal classifier utilizes visual features: Darker colors denote strong attention. Texts and tokens were translated into English. *uttr+img+desc* focuses attention on the ketchup, snack, and glass, which are important objects in (a), (b), and (c), respectively. However, it fails to focus attention on the water bottle in (d).

Table 6.5.: Automatic feature recognition performances

Feature	WER	MER	WIL
uttr	27.46	25.75	32.8097
Feature	Accuracy	-	-
position	98.25	-	-
pose	98.00	-	-
Feature	Precision	Recall	F1 (%)
coffee table	91.36	78.55	84.46
dining table	86.02	87.54	86.76
kitchen	91.54	72.49	80.72

these features are automatically recognized. Google Speech-to-Text API[§] was used for automatic speech recognition (ASR) of the user utterances. EfficientNet+MLP was used to recognize the descriptive features because the pre-trained object detection model [88] does not include the object classes for this study's settings. The training and evaluation of EfficientNet+MLP were conducted with 5-fold cross-validation of the same split as that in Section 6.3.1. We assumed that all of the classes of the descriptive features are known classes. Table 6.5 shows the performances of the automatic feature recognition. *Viewpoint* is not included because it can be determined based on the position of the robot itself. *Has* is not included because the number of interactions that include the *has* objects is too small to learn recognition as shown in Table 6.2. ASR accuracy was evaluated with word error rate (WER), match error rate (MER), and word information lost (WIL) [76]. ASR accuracy is moderate. This is a result of distant speech recognition, which assumes that the robot is standing farther than 2 meters from the user. All of the descriptive features are recognized with high accuracy, especially *position, pose* with almost 100% accuracy because the class number of *position, pose* is limited to three. Figure 6.5 shows the scatter plot between the number of occurrences in the dataset and not-recognized rates for objects: *coffee table*, *dining table*, and *kitchen*. The number of object classes is 28. The correla-

[§]https://cloud.google.com/speech-to-text

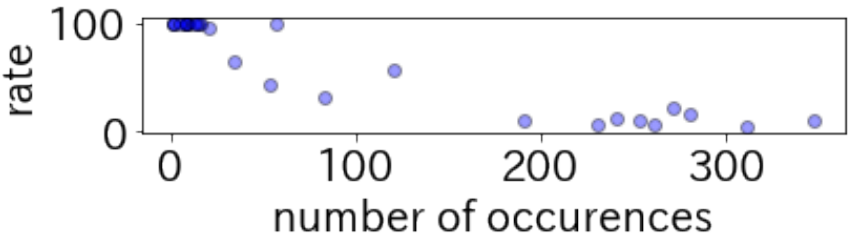


Figure 6.5.: Scatter plot between number of occurrences and not-recognized rates for objects. The correlation is −0.93.

tion is −0.93, indicating that the number of occurrences and the not-recognized rates have a strong negative correlation. For example, the number of occurrences of bread, potato, omelet rice, remote controller, trash, Japanese white radish, lemon, and can opener are less than 10. The recognized rates of these objects are 0%.

Table 6.6 shows the performances of the classifiers that use automatically recognized features. "^ (hat)" denotes that the user utterances or the descriptive features are automatically recognized. *Has* was input as *empty* to the models for all of the interactions because the recognition model could not learn the feature recognition. In all cases that use the recognition model, the performances significantly decreased compared to *uttr+img+desc*, especially in the case where the descriptive features were automatically recognized. Figure 6.6 shows examples where *Uttr+img+desc* selected the reflective actions but *uttr+img+desc^* could not recognize the important objects and selected wrong actions. In both examples, *remote control, trash*, which are important objects of the reflective actions, were not recognized. These objects have a low number of occurrences in the dataset. The percentage that *uttr+img+desc^* could not recognize the objects of the actions and selected wrong actions, such as the examples of Figure 6.6, is 67.54% of all errors of *uttr+img+desc^*. We need to develop a model that can recognize rare objects in order to mount the action selection model on the actual robot.

Comparing the performance of *uttr^+img+desc^* and *uttr^+img* or *uttr^+img*, we found that performance significantly improved for R@5 and MRR but that

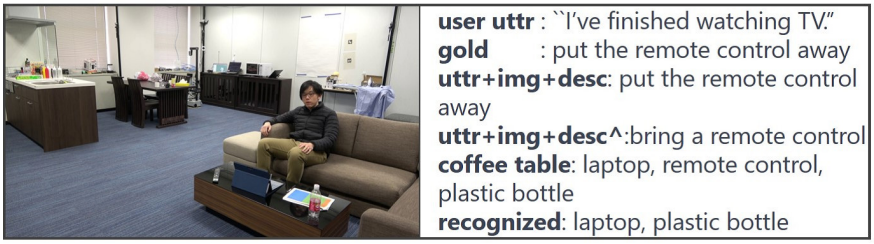
Table 6.6.: Classification results with recognized features. We conducted paired T-test. † means $p < 0.05$, and †† means $p < 0.01$.

Model	Accuracy (%)	R@5 (%)	MRR
uttr	27.02	53.85	0.4054
*uttr^*	22.10	43.80	0.3391
uttr+img	27.23	54.50	0.4064
uttr^+img	21.30	44.53	0.3369
uttr+img+desc	63.58	87.12	0.7417
uttr^+img+desc	††57.33	††83.93	††0.6921
*uttr+img+desc^*	††30.92	††61.80	††0.4577
*uttr^+img+desc^*	††22.20	††51.12	††0.3709

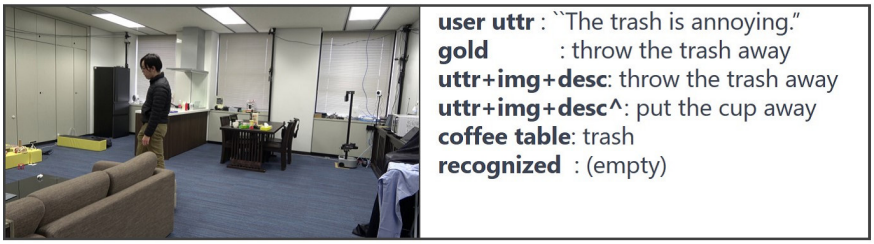
accuracy remained unchanged. In other words, just mounting the current recognition model of descriptive features on robots does not improve the accuracy of action selection. The results reinforce our awareness that we need to develop a model that can recognize descriptive features with high accuracy.

6.3.5. Variations of Baseline Classifiers Using Descriptive Features

Table 6.7 shows the performances when *uttr+img+desc* was trained with a different training method or a different way to process the descriptive features. *Uttr+img+desc (freeze)* denotes that the parameters of pre-trained RoBERTa and EfficientNet were frozen during the model training. This model is used to confirm the importance of fine-tuning. *Uttr+img+desc (word)* denotes that the model converts the descriptive features, which are word features, to feature vectors using a simple word embedding layer. This model is used to discuss the input method for the descriptive features. The word embedding layer was initialized with Japanese fastText vectors [13, 39]. The mean vector of word embeddings is treated as the descriptive feature vector when multiple words are input as a descriptive feature. *uttr+img+desc (sequence)* denotes that the descriptive features



(a)



(b)

Figure 6.6.: Examples where important objects for action selection were not recognized. Texts were translated into English.

Table 6.7.: Classification results of variations. We conducted paired T-test. †† means $p < 0.01$.

Model	Accuracy (%)	R@5 (%)	MRR
uttr+img	27.23	54.50	0.4064
uttr+img+desc (freeze)	††3.50	††57.78	††0.2559
uttr+img+desc (word)	††26.15	††54.28	††0.4019
uttr+img+desc (sequence)	††56.22	††83.15	††0.6829
uttr+img+desc (concat)	††49.50	††81.00	††0.6356
uttr+img+desc	63.58	87.12	0.7417

were input as a word sequence such as "The user has a glass and an alcohol bottle." to RoBERTa, instead of concatenation with [SEP] tokens. *Uttr+img+desc (concat)* denotes that *has, coffee table, dining table, kitchen* were input as the objects included in a picture to the model. Both models are understood to vary

Table 6.8.: Ratios of predicated robot action categories

category	ratio (%)
bring	49.28
put away	46.70
others	4.03

in the explicitness of the descriptive features.

The accuracy of *uttr+img+desc (freeze)* is almost the same value as the chance rate (1/40 = 2.5%), although the R@5 is around 60. This result means that the parameters of the pre-trained models should be fine-tuned to learn the reflective action selection. We confirmed that the performances of *uttr+img+desc (word)* is significantly lower than *uttr+img+desc* using a paired t-test ($p < 0.01$), indicating that we need to convert descriptive features to feature vectors using RoBERTa, although they are word features. The paired t-test results between the performances of *uttr+img+desc (sequence)* and *uttr+img+desc* show that the former is significantly lower than the latter. This result is the same for the paired t-test between the performances of *uttr+img+desc (concat)* and *uttr+img+desc*. These decreases in performance demonstrate the importance of inputting the descriptive features in an explicit way, although *uttr+img+desc (sequence)* and *uttr+img+desc (concat)* input the described features at different levels of explicitness.

6.3.6. Error Analysis for Model Improvement

Table 6.8 shows the ratio of the rough action categories that *uttr+img+desc* predicted. Comparing Table 6.8 and Table 6.1, we see that *uttr+img+desc* predicted action categories with a ratio similar to the true ratio, and that it did not ignore action categories with low frequency.

Figure 6.7 shows the histogram for the error rates of the interactions per action category in *uttr+img+desc*. The error rates vary significantly among the action categories, indicating that the model tends to select wrong actions for specific user situations. The red bins of Figure 6.7 represent action categories that have

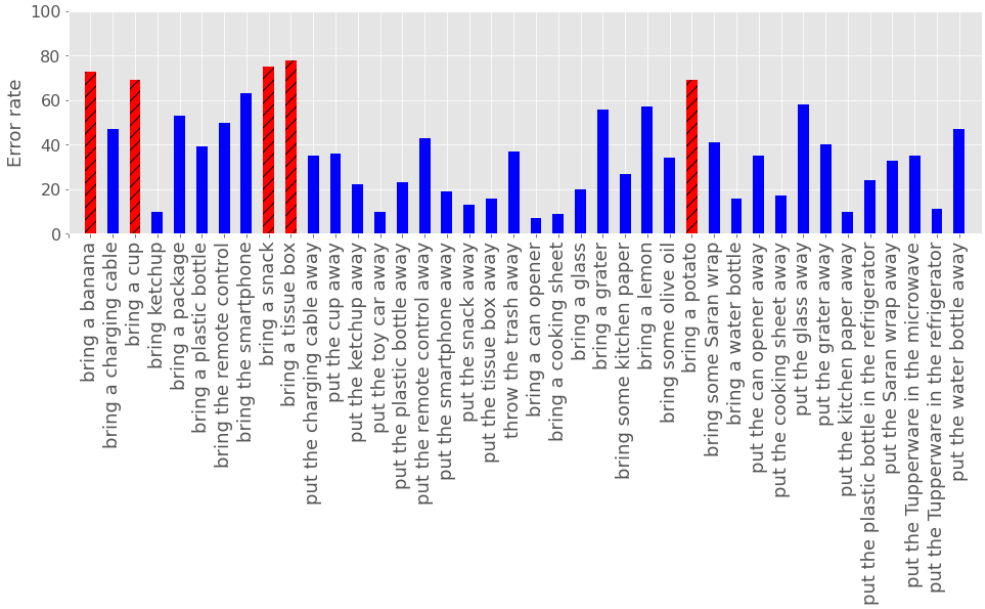


Figure 6.7.: Histogram for the error rates of the interactions per action category: Red bins represent categories that have the top-five error rates.

the top-five error rates. All of the top-five categories are *bring* actions, indicating that selecting correct *bring* actions is more difficult than selecting correct *put away* actions because the target objects of the actions are placed near the user, as shown in Figure 6.4 (a) (b), when *put away* actions can be regarded as reflective. In fact, we summarized the classification results for each rough action category and found that the *bring* actions have lower precision and recall than the *put away* actions, as shown in Table 6.9.

Figure 6.8 shows the example interactions of *bring a tissue box*, which is the most misclassified action category. In Figure 6.8 (a), the model selected the wrong action *put the tissue box away*, which includes the correct target object but the wrong predicate, for the user utterance "My nose is bleeding." The percentage of cases in which the objects of the selected actions are correct but the verbs are wrong is 9.68% of all errors. Resolving these cases is expected to contribute to improving classification accuracy. One possible solution is to apply causal relations derived using causal inference [15, 117], such as "My nose is bleeding." $\rightarrow$ *need to wipe the face* $\rightarrow$ *bring a tissue box*. In Figure 6.8 (b), the model

Table 6.9.: Classification results per robot action category

category	Precision (%)	Recall (%)	F1 (%)
bring	56.16	55.35	55.75
put away	70.02	72.67	71.32
others	79.50	64.00	70.91

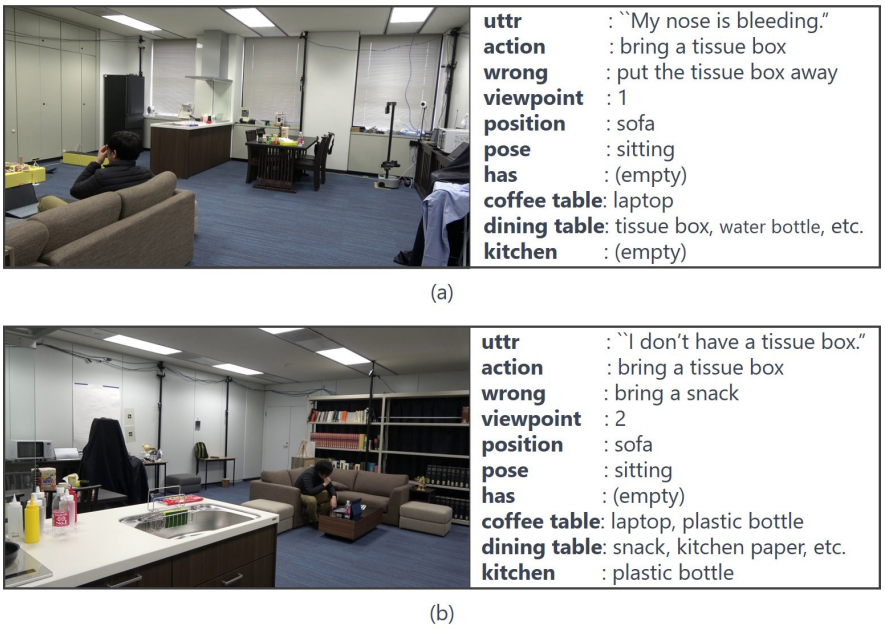


Figure 6.8.: Examples of frequently misclassified interactions. Texts were translated into English.

selected the unrelated action *bring a snack* even when the user utterance included "tissue box," indicating that the model failed to select actions with simple word matching. Some of the collected user utterances include the target objects for the robot actions. The percentage of cases in which the objects of the selected actions were included in the user utterances but wrong actions were selected is 22.99% of all errors. We expect that resolving these easier misclassifications will lead to the improvement of classification accuracy.

6.4. Conclusion

This study focused on constructing a dataset for training action selection by a robot that understands the surrounding situations and takes reflective actions when the user's request is ambiguous. We used crowdsourcing to collect user situations in which the pre-defined robot actions can be regarded as reflective. In addition, we recorded videos in a living room or kitchen corresponding to the collected situations. The developed classifier selects appropriate reflective robot actions by understanding the user situations from a robot's first-person viewpoint. Our experimental results show that using descriptive image features significantly improved classification accuracy, even if only a small volume of data is available as training data.

Our future work will investigate model architecture that can recognize descriptive features with high accuracy. Using the developed model, we will build a robot that can understand situations and select reflective actions based solely on user utterances and first-person images. In addition, we will investigate a causal inference system to connect the user situations to the robot actions for developing a model that can understand the user situations more precisely.

7. Conclusions

This dissertation tackled the problems of realizing the dialogue systems that generate reflective responses and actions. Since dialogue systems are traditionally categorized into non-task-oriented or task-oriented, we studied the generation of reflective responses and actions for both non-task-oriented dialogue systems and task-oriented dialogue systems. First, we discussed the detailed definition of the reflective responses and actions of the dialogue systems. The reflective responses of non-task-oriented dialogue systems are responses with high dialogue continuity that the users want to continue dialogue with the systems. The reflective actions of task-oriented dialogue systems are actions that satisfy users' potential requests even if the users do not explicitly verbalize their requests. In order to focus on the generation of reflective responses and actions, the studies of this dissertation share the task definition to generate a single reflective response or action based on a user utterance, dialogue contexts, or situations surrounding the user. There were a lot of problems with the generation of reflective responses and actions even if the task definition does not require multi-turn dialogue management. This dissertation proposed the methods utilizing user's events that are included in the user utterances and the situations surrounding the user to generate reflective responses and actions. The problems and the proposed methods of this dissertation are summarized below.

First, we tackled the dull response problem that the existing non-task-oriented dialogue systems tend to generate non-reflective responses with low dialogue continuity. Response generation models using neural networks tend to generate statistically appropriate but simple responses that apply to any utterance. We proposed the method to select responses with high dialogue continuity based on coherency between the user utterances and the system responses. The proposed method treats PA structures included in user utterances and systems responses

as events, and re-ranks response candidates based on event causality relations between the events. We also proposed the method that deals with coherency between the entire user utterances and system responses. The experimental results show that the proposed re-ranking method improves dialogue continuity of the system responses, but decreases coherency. The results seem contradictory against the concept of the proposed method. The various analyses of the experimental results suggest that the coherency of PA structures between the user utterances and the system responses, rather than the coherency of the entire responses, contributes to the improvement of dialogue continuity of the responses.

Second, we tackled the problem that the existing task-oriented dialogue systems cannot select the reflective actions for ambiguous user requests. Although a corpus is necessary to develop the dialogue system, there was no corpus consisting of ambiguous user requests and reflective system actions. The conventional collection method is inappropriate for this task definition. In this study, we could collect the high-quality corpus by asking the crowd-workers to input the antecedent user utterances for the predefined reflective system actions. Although multiple system actions can be regarded as reflective when the user requests are ambiguous, complete annotation is impractical because it is extremely expensive. We improved the action selection accuracy for the ambiguous requests by training the model with the incomplete labeled dataset using PU learning. We incorporated the causality knowledge between events which human concierges utilize into the PU learning, and verified the validity of the proposed method.

Finally, we tackled the problem that text-based dialogue systems do not utilize multimodal information to select reflective actions as humans do. We collected the multimodal corpus because there is no corpus including the reflective actions as well as the text-based corpus. We recorded the videos representing the situations that the user verbalizes the ambiguous requests based on the texts representing the user utterances and the surrounding situations collected with the collection method for the text-based corpus including the reflective actions. Constructing a large multimodal corpus is difficult because it is expensive to collect. We proposed labeling the images clipped from the videos with events that describe the surrounding situations of the user as the descriptive features in order to efficiently train the model with the small dataset. Experimental results show that training

the baseline model with the descriptive features is effective in improving the selection accuracy for the reflective actions compared to the models trained with only the user utterances and the images. In other words, we proved the validity of using the events that describes the surrounding situations of the user to select the reflective actions.

Dialogue systems proposed in Chap. 5 and Chap. 6 take reflective actions on specific tasks such as sightseeing navigation or life-support. Although these systems seem task-dependent, the corpus collection method and the architectures of the action selection models are task-independent methods that can be applied to other tasks such as cooking assistanceance or car-navigation. Actually, although dialogue systems of Chap. 5 and Chap. 6 deal with different tasks, the corpus collection methods and the model architectures are similar, proving that these methods can be applied to a wide range of tasks.

All the proposed methods in this dissertation significantly contribute to solving the problems to generate reflective responses and actions on non-task-oriented dialogue or task-oriented dialogue. We expect that future dialogue systems need to satisfy the users with content that they do not explicitly verbalize without distinguishing between non-task-oriented dialogue and task-oriented dialogue. We believe that the proposed methods in this dissertation provide clues for the development of dialogue systems that generate reflective responses.

7.1. Remaining Problems and Future Directions

In this dissertation, we investigated the methods to generate reflective system responses and actions on non-task-oriented dialogue or task-oriented dialogue. There are remaining problems although all the proposed methods contribute to generating reflective responses and actions. We discuss the remaining problems for each research theme and for the whole of this dissertation below.

First, we discuss the problems of the method that re-ranks response candidates generated by a non-task-oriented response generation model based on coherency of events. According to the human evaluation, we confirmed that our re-ranking method selects the reflective responses with high dialogue continuity. However, since the proposed method is a re-ranking method, the re-ranking does not work

if there is no response candidate that has an event causality relation with user utterances. The system has to generate responses that have causality relations with the user utterances during the response generation process to solve this problem. Lu et al. [65, 66] proposed methods to generate system responses that include specific words. We believe that these methods provide a solution to this problem. In addition, there is still room to improve the accuracy of detecting causality relations between events as shown in Section 4.3.2. We need to discuss what kind of causality relations lead to improve dialogue continuity. We believe that a possible solution is to filter the causality relations used for re-ranking and response generation. For example, the system can change causality relations used for response generation based on user profiles. When a user said "I'm stressed out," the system can suggest "Let's play a TV game" to an introverted user based on a causality relation "be stressed out" → "play a TV game," and can suggest "Let's go jogging" to an extroverted user based on a causality relation "be stressed out " → "go jogging." Causality Detection Model proposed in Chap. 5 can be used to detect useful causality relations based on user profiles. Note that training of Causality Detection Model requires pairs of user utterances and system responses that have causality relations. These pairs are expensive to collect. Although all causality relations used in this study are one-hop relations in which a cause and an effect are linked one-to-one, QA systems [41] and commonsense reasoning systems [15] use multi-hop causality relations to derive events that are not directly connected to cause events. Non-task-oriented systems can also utilize multi-hop causality relations to generate responses. For example, even if a causality relation "be stressed out" → "play a TV game" is not directly derived, the system can use two causality relations "be stressed out" → "want to relieve stress" and "want to relieve stress" → "play a TV game" as multi-hop causality relations to generate a response "Let's play a TV game" to the user utterance "I'm stressed out."

Second, we discuss the problems of the classifier to select the reflective actions to ambiguous user requests on text-based dialogue. Our classifier selected the reflective action categories for ambiguous user requests with a high accuracy of about 90% as shown in Section 5.3.3. We expect that our proposed classifier will be used as a high-performance baseline in future work related to task-oriented dialogue systems that deal with ambiguous user requests. However, we are concerned

about whether users might continue to use our system (which takes incorrect actions once every ten times) when it is used as an actual dialogue system. We address this problem by presenting two ideas other than improving the classification accuracy. Since the R@5 of the proposed classifier is almost 100%, we propose a method that presents actions with the top 5 probabilities to users and asks them to select one of the actions [83]. In addition, if a dialogue system cannot select one action category with a high probability through a single-turn dialogue, it can clarify a user request through a multi-turn dialogue [48]. This study did not address clarification of the ambiguous user requests through such multi-turn dialogues. We need to collect a new corpus and build a new model to solve this problem in the future. The label propagation method proposed in this study propagates action categories with high precision and improves classification accuracy (Sections 5.3.3 and 5.3.4). These results indicate that PU learning based on label propagation is a practical method to address the impractical task of annotating all the combinations of ambiguous user requests and reflective system actions. On the other hand, our proposed label propagation method suffers from a trade-off between precision and recall (Section 5.3.4). We must investigate a method to improve the precision of label propagation without decreasing its recall. To address this problem, the label propagation method needs to more precisely understand the causality relations between the user requests and the system actions or the semantic similarities between the user requests. We expect that raising the accuracy that detects the causalities between the user requests and the system actions will improve the label propagation's performance because the label propagation method based on causality can decrease the false-positive effect on loss functions while suppressing the decrease of the true-positive effect (Table 5.11).

Third, we discuss the problem of the model that uses the events that represent the surrounding situations of the user in order to select the reflective system actions in multimodal dialogue. The developed baseline model assumes the ideal situation in which the user utterances and the descriptive features representing the surrounding situation are recognized with 100% accuracy. When the dialogue system is installed in an actual robot, the system must automatically recognize the user utterances and the surrounding situations. Although we developed the

model that automatically recognizes the user utterances and the surrounding situations, the selection accuracy of the reflective actions decreases when the model automatically recognizes the features. In particular, when the descriptive features were automatically recognized, the selection accuracy dramatically decreased compared to the baseline model that assumes the ideal situation: The model can select the reflective actions with only about 30%. We do not expect that the users will continue to use the nosy dialogue system that often takes inappropriate reflective actions. In other words, precision is more important than recall for the selection of the reflective actions. We need to develop a model that does nothing if a reflective action cannot be selected with a high confidence. In addition, we need to improve the accuracy of the automatic recognition models for the descriptive features. Although this study utilized the events of the surrounding situations of the user, it did not utilize the causality relations between events of the user utterances and the system actions. As shown in Section 6.3.6, the system occasionally cannot select appropriate reflective actions based solely on the surrounding situations of the user. We need to investigate methods to select reflective actions by integrating causality relations between the events in the user utterances and the surrounding situations and the events in the system action to solve this problems. As with responses on non-task-oriented dialogue, appropriate causality relations and reflective actions depend on user profiles and situations. For example, when a user said "I'm hungry," there are various appropriate causality relations depending on the user profile and timeframes such as "be hungry" $\rightarrow$ "have a banana" or "be hungry" $\rightarrow$ "have a hamburger."

Finally, we discuss the remaining problem of the whole of this dissertation. Although this dissertation addressed the generation of the reflective responses and actions of dialogue systems independently for non-task-oriented dialogue and task-oriented dialogue, as described in Chap. 1, we expect that future dialogue systems need to satisfy the users without distinguishing between non-task-oriented dialogue and task-oriented dialogue. In other words, we need to develop a dialogue system that generates reflective responses and actions for users in both non-task-oriented dialogue and task-oriented dialogue by integrating both dialogue systems that generate the reflective responses and actions. We need to investigate methods to train the dialogue system with a mixed corpus

that consists of non-task-oriented dialogue and task-oriented dialogue [64, 114] to achieve this goal. Then, we discuss the technical problems to train a dialogue system using the mixed corpus. Figure 1.1 of Chap. 1 showed the three steps to develop dialogue systems that generate reflective responses on non-task-oriented and task-oriented dialogues. The proposed systems of this dissertation can be regarded as the first step systems that generate reflective responses on non-task-oriented or task-oriented dialogues. However, in order to realize dialogue systems that integrate non-task-oriented and task-oriented dialogues, we need to develop a module that determines whether to respond to user utterances with non-task-oriented or task-oriented responses. This is the second step to develop dialogue systems that generate reflective responses. In addition, a dialogue management module, which manages multi-turn dialogues with users, is necessary to realize the third step of dialogue systems that generate reflective responses on multi-turn dialogues. In other words, although this dissertation proposed the core methods to generate reflective responses to user utterances, peripheral modules should be developed to realize dialogue systems that work in the real world. Furthermore, in order to demonstrate the applicability of the proposed methods for generating reflective responses or actions to various real-world tasks, we need to develop dialogue systems with the proposed methods applied to other tasks, such as cooking assistance or car navigation, including the peripheral modules.

Acknowledgements

First of all, I would like to express my gratitude to Professor Satoshi Nakamura for his careful reviews and fruitful discussions on my research. He gave me much important advice to make my research life meaningful. I would thank Professor Taro Watanabe for agreeing to join the member of my doctoral committee and giving insightful comments on my dissertation. I wish to thank Associate Professor Katsuhito Sudoh for his kind and various comments on my studies. I would like to express my gratitude to Affiliate Professor Koichiro Yoshino for his enthusiastic research guidance on my research. I wish to thank Professor Giuseppe Riccardi for his kind advice during my study abroad.

I would thank Mrs. Manami Matsuda, secretary of AHC Laboratory, for her help in various procedures such as paperwork. I would like to express my sincere gratitude to the members of AHC Laboratory and SIS Laboratory for their support. I would like to thank Professor Sadao Kurohashi and Dr. Tomohide Shibata of Kurohashi Laboratory in Kyoto University who provided the event causality pairs. Finally, I would like to express my gratitude to my parents and other families for giving me the opportunity to study at NAIST, and to my friends for their support.

References

[1] Amazon.com: Echo Smart Speakers &; Displays: Amazon Devices &; Accessories: Smart Speakers, Smart Displays &; More. https://www.amazon.com/smart-home-devices/b?ie=UTF8&node=9818047011. (Accessed on March 17, 2023).

[2] Event definition and meaning | Collins English Dictionary. https://www.collinsdictionary.com/dictionary/english/event. (Accessed on March 17, 2023).

[3] Proactive definition and meaning | Collins English Dictionary. https://www.collinsdictionary.com/dictionary/english/proactive. (Accessed on March 17, 2023).

[4] Reactive definition and meaning | Collins English Dictionary. https://www.collinsdictionary.com/dictionary/english/reactive. (Accessed on March 17, 2023).

[5] Reflective definition and meaning | Collins English Dictionary. https://www.collinsdictionary.com/dictionary/english/reflective. (Accessed on March 17, 2023).

[6] メイ&; タクミ公式ウェブサイト | 国立大学法人名古屋工業大学. https://mei.web.nitech.ac.jp/. (Accessed on March 17, 2023).

[7] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng

Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do As I Can and Not As I Say: Grounding Language in Robotic Affordances. In *arXiv preprint arXiv:2204.01691*, 2022.

[8] Dzmitry Bahdanau, Kyunghyunand Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[9] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[10] Ankur Bapna, Gokhan Tür, Dilek Hakkani-Tür, and Larry Heck. Sequential Dialogue Context Modeling for Spoken Language Understanding. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 103–114, Saarbrücken, Germany, August 2017. Association for Computational Linguistics.

[11] Dasha Bogdanova and Jennifer Foster. This is how we do it: Answer Reranking for Open-Domain How Questions with Paragraph Vectors and Minimal Feature Engineering. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1290–1295, 2016.

[12] Dan Bohus and Alexander I Rudnicky. RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda. In *Proceeedings of the 8th European Conference on Speech Communication and Technology*, 2003.

[13] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[14] Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. In *International Conference on Learning Representations*, 2017.

[15] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy, July 2019. Association for Computational Linguistics.

[16] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[17] Harry Bunt, Jan Alexandersson, Jean Carletta, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Kiyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, Claudia Soria, and David Traum. Towards an ISO Standard for Dialogue Act Annotation. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta, May 2010. European Language Resources Association.

[18] Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David Traum. ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 430–437, Istanbul, Turkey, May 2012. European Language Resources Association.

[19] Alessandra Cervone, Evgeny Stepanov, and Giuseppe Riccardi. Coherence Models for Dialogue. In *Proceedings of INTERSPEECH 2018*, 2018.

[20] Hakan Cevikalp, Burak Benligiray, and Omer Nezih Gerek. Semi-supervised robust deep neural networks for multi-label image classification. *Pattern Recognition*, 100:107164, 2020.

[21] Hakan Cevikalp, Jakob Verbeek, Frédéric Jurie, and Alexander Klaser. Semi-supervised dimensionality reduction using pairwise equivalence constraints. In *3rd International Conference on Computer Vision Theory and Applications*, pages 489–496, 2008.

[22] Nathanael Chambers and Dan Jurafsky. Unsupervised Learning of Narrative Event Chains. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 789–797, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[23] Nathanael Chambers and Dan Jurafsky. Unsupervised Learning of Narrative Schemas and their Participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 602–610, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[24] Yuya Chiba and Ryuichiro Higashinaka. Dialogue Situation Recognition for Everyday Conversation Using Multimodal Information. In *Proceedings of INTERSPEECH 2021*, pages 241–245, 2021.

[25] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[26] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence

Modeling. In *Proceedings of the 28th Conference Neural Information Processing Systems, Deep Learning and Representation Learning Workshop*, 2014.

[27] David Cohen and Ian Lane. A Simulation-based Framework for Spoken Language Understanding and Action Selection in Situated Interaction. In *NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data*, pages 33–36, Montréal, Canada, June 2012. Association for Computational Linguistics.

[28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 18th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.

[29] Antonio Di Marco and Roberto Navigli. Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction. *Computational Linguistics*, 39(3):709–754, 2013.

[30] George Doddington. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the 2nd International Conference on Human Language Technology Research*, pages 138–145, 2002.

[31] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 207–219, Saarbrücken, Germany, August 2017. Association for Computational Linguistics.

[32] Charles Elkan and Keith Noto. Learning Classifiers from Only Positive and Unlabeled Data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 213–220, New York, NY, USA, 2008. Association for Computing Machinery.

[33] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. Key-Value Retrieval Networks for Task-Oriented Dialogue. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany, August 2017. Association for Computational Linguistics.

[34] Motoyasu Fujita, Rafal Rzepka, and Kenji Araki. Evaluation of Utterances Based on Causal Knowledge Retrieved from Blogs. In *Proceedings of the 14th IASTED International Conference Artificial Intelligence and Soft Computing*, pages 294–299, 2011.

[35] Forgues Gabriel, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. Bootstrapping Dialog Systems with Word Embeddings. In *Proceedings of the NIPS 2014 workshop on Modern Machine Learning and Natural Language Processing*, 2014.

[36] Ping Ganbin Zhou, Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. Mechanism-Aware Neural Machine for Dialogue Response Generation. In *Proceedings of the 31st Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, pages 3400–3406, 2017.

[37] Felix Gervits, Antonio Roque, Gordon Briggs, Matthias Scheutz, and Matthew Marge. How Should Agents Ask Questions For Situated Learning? An Annotated Dialogue Corpus. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 353–359, Singapore and Online, July 2021. Association for Computational Linguistics.

[38] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[39] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2018.

[40] Xiaoxiao Guo, Tim Klinger, C. Rosenbaum, J. P. Bigus, Murray Campbell, B. Kawas, Kartik Talamadupula, G. Tesauro, and S. Singh. Learning to Query, Reason, and Answer Questions On Ambiguous Texts. In *International Conference on Learning Representations*, 2017.

[41] Chikara Hashimoto, Kentaro Torisawa, Julien Kloetzer, Motoki Sano, Istvan Varga, Jong-Hoon Ohk, and Yutaka Kidawara. Toward Future Scenario Generation: Extracting Event Causality Exploiting Semantic Relation, Context, and Association Features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 987–997, 2014.

[42] Ben Hixon, Rebecca J. Passonneau, and Susan L. Epstein. Semantic Specificity in Spoken Dialogue Requests. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 257–260, Seoul, South Korea, July 2012. Association for Computational Linguistics.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[44] Stephen James, Michael Bloesch, and Andrew J. Davison. Task-Embedded Control Networks for Few-Shot Imitation Learning. In *Conference on Robot Learning*, 2018.

[45] Peter Jansen, Mihai Surdeanu, and Peter Clark. Discourse Complements Lexical Semantics for Non-factoid Answer Reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 977–986, 2014.

[46] Mihir Kale and Abhinav Rastogi. Template Guided Text Generation for Task-Oriented Dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6505–6520, Online, November 2020. Association for Computational Linguistics.

[47] A. Kanehira and T. Harada. Multi-label Ranking from Positive and Unlabeled Data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5138–5146, 2016.

[48] Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y-Lan Boureau, and Jason Weston. Recommendation as a Communication Game: Self-Supervised Bot-Play for Goal-oriented Dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1951–1961, Hong Kong, China, November 2019. Association for Computational Linguistics.

[49] Daisuke Kawahara. Kawahara Lab. RoBERTa (https://huggingface.co/nlp-waseda/roberta-base-japanese). 2021.

[50] Daisuke Kawahara and Sadao Kurohashi. A Fully-Lexicalized Probabilistic Model for Japanese Syntactic and Case Structure Analysis. In *Proceedings of Human Language Technology Conference - North American Chapter of the Association for Computational Linguistics Annual Meeting*, pages 176–183, 2006.

[51] Evgeny Kim and Roman Klinger. Who Feels What and Why? Annotation of a Literature Corpus with Semantic Roles of Emotions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1345–1359, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[52] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[53] Taku Kudo and John Richardson. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[54] Hung Le, Chinnadhurai Sankar, Seungwhan Moon, Ahmad Beirami, Alborz Geramifard, and Satwik Kottur. DVD: A Diagnostic Dataset for Multi-step Reasoning in Video Grounded Dialogue. In *Proceedings of the 59th Annual*

Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5651–5665, Online, August 2021. Association for Computational Linguistics.

[55] Kyung-Soon Lee, Kyo Kageura, and Key-Sun Choi. Implicit Ambiguity Resolution Using Incremental Clustering in Korean-to-English Cross-Language Information Retrieval. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

[56] Nyoungwoo Lee, Suwon Shin, Jaegul Choo, Ho-Jin Choi, and Sung-Hyon Myaeng. Constructing Multi-Modal Dialogue Dataset by Replacing Text with Semantically Relevant Images. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 897–906, Online, August 2021. Association for Computational Linguistics.

[57] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

[58] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, 2016.

[59] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards Deep Conversational Recommendations. In *Advances in Neural Information Processing Systems*, volume 31, pages 9725–9735. Curran Associates, Inc., 2018.

[60] Zujie Liang, Huang Hu, Can Xu, Chongyang Tao, Xiubo Geng, Yining Chen, Fan Liang, and Daxin Jiang. Maria: A Visual Experience Powered Conversational Agent. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

Conference on Natural Language Processing, pages 5596–5611, Online, August 2021. Association for Computational Linguistics.

[61] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proceedings of the 13th European Conference on Computer Vision*, pages 740–755, Cham, 2014. Springer International Publishing.

[62] Chia-Wei Liu, Ryan Lowe, Iulian V. Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How NOT to Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[63] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In *arXiv*, 2019.

[64] Zeming Liu, Jun Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, and Hua Wu. Where to Go for the Holidays: Towards Mixed-Type Dialogs for Clarification of User Goals. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 1024–1034, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[65] Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States, July 2022. Association for Computational Linguistics.

[66] Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. NeuroLogic Decoding: (Un)supervised Neural Text

Generation with Predicate Logic Constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299, Online, June 2021. Association for Computational Linguistics.

[67] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-Based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[68] Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap Between Human and Machine Translation. In *arXiv:1609.08144*, 2016.

[69] Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1468–1478, 2018.

[70] Matthew Marge, Carol Espy-Wilson, Nigel G. Ward, Abeer Alwan, Yoav Artzi, Mohit Bansal, Gil Blankenship, Joyce Chai, Hal Daumé, Debadeepta Dey, Mary Harper, Thomas Howard, Casey Kennington, Ivana Kruijff-Korbayová, Dinesh Manocha, Cynthia Matuszek, Ross Mead, Raymond Mooney, Roger K. Moore, Mari Ostendorf, Heather Pon-Barry, Alexander I. Rudnicky, Matthias Scheutz, Robert St. Amant, Tong Sun, Stefanie Tellex, David Traum, and Zhou Yu. Spoken language interaction with robots: Recommendations for future research. *Computer Speech & Language*, 71:101255, 2022.

[71] Hongyuan Mei, Mohit Bansal, and Matthew R Walter. Coherent Dialogue with Attention-Based Language Models. In *Proceedings of the 31st Associ-*

ation for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, 2017.

[72] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Deany. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the 1st International Conference on Learning Representations*, 2013.

[73] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2, pages 3111–3119, 2013.

[74] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 12th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

[75] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854, Florence, Italy, July 2019. Association for Computational Linguistics.

[76] Andrew Morris, Viktoria Maier, and Phil Green. From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In *Proceedings of INTERSPEECH 2004*, 10 2004.

[77] Ryo Nakamura, Katsuhito Sudoh, Koichiro Yoshino, and Satoshi Nakamura. Another Diversity-Promoting Objective Function for Neural Dialogue Generation. In *Proceedings of the 33rd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, Workshop on Reasoning and Learning for Human-Machine Dialogues*, 2019.

[78] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic Evaluation of Topic Coherence. In *Proceedings of the 11th Annual*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 100–108, 2010.

[79] Jong-Hoon Oh, Kentaro Torisawa, Chikara Hashimoto, Ryu Iida, Masahiro Tanaka, and Julien Kloetzer. A Semi-supervised Learning Approach to Why-Question Answering. In *Proceedings of the 30th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, pages 3022–3029, 2016.

[80] Jong-Hoon Oh, Kentaro Torisawa, Chikara Hashimoto, Motoki Sano, Stijn De Saeger, and Kiyonori Ohtake. Why-Question Answering Using Intra- and Inter-Sentential Causal Relations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1733–1743, 2013.

[81] Jong-Hoon Oh, Kentaro Torisawa, Canasai Kruengkrai, Ryu Iida, and Julien Kloetzer. Multi-Column Convolutional Neural Networks with Causality-Attention for Why-Question Answering. In *Proceedings of the 10th Association for Computing Machinery International Conference on Web Search and Data Mining*, pages 415–424, 2017.

[82] Junki Ohmura and Maxine Eskenazi. Context-Aware Dialog Re-ranking for Task-Oriented Dialog Systems. In *Proceedings of IEEE Spoken Language Technology Workshop*, 2018.

[83] Yi-Cheng Pan, Hung-Yi Lee, and Lin-Shan Lee. Interactive Spoken Document Retrieval With Suggested Key Terms Ranked by a Markov Decision Process. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):632–645, 2012.

[84] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[85] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein,

Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[86] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[87] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. Conversational AI: The Science Behind the Alexa Prize. In *arXiv:1801.03604*, 2018.

[88] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society.

[89] David Rumelhart, Geoffrey Hinton, and Ronald Williams. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536, 1986.

[90] Ryohei Sasano and Sadao Kurohashi. A Discriminative Approach to Japanese Zero Anaphora Resolution with Large-Scale Lexicalized Case Frames. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 758–766, 2011.

[91] Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Proceedings of the 30th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, 2016.

[92] Tomohide Shibata, Daisuke Kawahara, and Kurohashi Sadao. Kurohashi Lab. BERT (http://nlp.ist.i.kyoto-u.ac.jp/?ku_bert_japanese). 2019.

[93] Tomohide Shibata, Shotaro Kohama, and Sadao Kurohashi. A Large Scale Database of Strongly-Related Events in Japanese. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, 2014.

[94] Tomohide Shibata and Sadao Kurohashi. Acquiring Strongly-Related Events Using Predicate-Argument Co-occurring Statist ics and Case Frames. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 1028–1036, 2011.

[95] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob G. Simonsen, and Jian-Yun Nie. A Hierarchical Recurrent Encoder-Decoder For Generative Context-Aware Query Suggestion. In *Proceedings of the 24th Association for Computing Machinery International Conference on Information Knowledge and Management*, 2015.

[96] Ilya Sutskever, Oriol Vinyals, and V. Le Quoc. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 28th Conference Neural Information Processing Systems*, volume 2, pages 3104–3112, 2014.

[97] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6105–6114. PMLR, 09–15 Jun 2019.

[98] Robert S. Taylor. The process of asking questions. *American Documentation*, pages 391–396, 1962.

[99] Robert S. Taylor. Question-Negotiation and Information Seeking in Libraries. *College & Research Libraries*, 29(3):178–194, 1968.

[100] Alberto Testoni and Raffaella Bernardi. Looking for Confirmations: An Effective and Human-Like Visual Dialogue Strategy. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9330–9338, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[101] Geoffrey Towell and Ellen M. Voorhees. Disambiguating Highly Ambiguous Words. *Computational Linguistics*, 24(1):125–145, 1998.

[102] Andrea Vanzo, Emanuele Bastianelli, and Oliver Lemon. Hierarchical multi-task natural language understanding for cross-domain conversational ai: Hermit nlu. In *Proceedings of the 20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 254–263, Stockholm, Sweden, September 2019. Association for Computational Linguistics.

[103] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017.

[104] Oriol Vinyals and Quoc V. Le. A Neural Conversational Model. In *Proceedings of the 32nd International Conference on Machine Learning, Deep Learning Workshop*, 2015.

[105] Yu Wang and Eugene Agichtein. Query Ambiguity Revisited: Clickthrough Measures for Distinguishing Informational and Ambiguous Queries. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 361–364, Los Angeles, California, June 2010. Association for Computational Linguistics.

[106] Noah Weber, Niranjan Balasubramanian, and Nathanael Chambers. Event Representations with Tensor-Based Compositions. In *Proceedings of the 32nd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, 2018.

[107] Jean-Baptiste Weibel, Timothy Patten, and Markus Vincze. Addressing the Sim2Real Gap in Robotic 3D Object Classification. *IEEE Robotics and Automation Letters*, 5(2):407–413, 2020.

[108] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Annual*

Meeting of the Association for Computational Linguistics, pages 808–819, Florence, Italy, July 2019. Association for Computational Linguistics.

[109] Peng Xu, Hamidreza Saghir, Jin Sung Kang, Teng Long, Avishek Joey Bose, Yanshuai Cao, and Jackie Chi Kit Cheung. A Cross-Domain Transferable Neural Coherence Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 678–687, 2019.

[110] Koichiro Yoshino, Yu Suzuki, and Satoshi Nakamura. Information Navigation System with Discovering User Interests. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 356–359, Saarbrücken, Germany, August 2017. Association for Computational Linguistics.

[111] Koichiro Yoshino, Hiroki Tanaka, Kyoshiro Sugiyama, Makoto Kondo, and Satoshi Nakamura. Japanese Dialogue Corpus of Information Navigation and Attentive Listening Annotated with Extended ISO-24617-2 Dialogue Act Tags. In *Proceedings of the 13th International Conference on Language Resources and Evalu ation*, 2018.

[112] Koichiro Yoshino, Kohei Wakimoto, Yuta Nishimura, and Satoshi Nakamura. Caption Generation of Robot Behaviors based on Unsupervised Learning of Action Segments. In *Conversational Dialogue Systems for the Next Decade*, pages 227–241, 2021.

[113] Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174, 2010.

[114] Zhou Yu, Alan W. Black, and Alexander I. Rudnicky. Learning Conversational Systems That Interleave Task and Non-Task Content. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, page 4214–4220. AAAI Press, 2017.

[115] Xiaoxue Zang, Lijuan Liu, Maria Wang, Yang Song, Hao Zhang, and Jindong Chen. PhotoChat: A Human-Human Dialogue Dataset With Photo

Sharing Behavior For Joint Image-Text Modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 6142–6152, Online, August 2021. Association for Computational Linguistics.

[116] Denny Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning from Labeled and Unlabeled Data on a Directed Graph. In *Proceedings of the 22nd international conference on Machine learning*, page 1036. ACM Press, August 2005.

[117] 清丸 寛一, 植田 暢大, 児玉 貴志, 田中 佑, 岸本 裕大, 田中 リベカ, 河原 大輔, and 黒橋 禎夫. 因果関係グラフ: 構造的言語処理に基づくイベントの原因・結果・解決策の集約. In **言語処理学会 第 *26* 回年次大会 発表論文集**, pages 1125–1128, 2020.

[118] 徳久 良子 and 寺嶌 立太. 非課題遂行対話における発話の特徴とその分析. **人工知能学会論文誌**, 22(4):425–435, 2007.

A. Appendix

A.1. Additional Examples of User Requests on Text-based Dialogue

Table A.1-A.3 show examples of user requests for all pre-defined system actions on Chap. 5.

A.2. Additional Examples of Interactions on Multimodal Dialogue

Figure A.1-A.10 show examples of interactions for all pre-defined robot actions on Chap. 6.

Table A.1.: User requests for all pre-defined system actions of spot search: Texts were translated from Japanese to English.

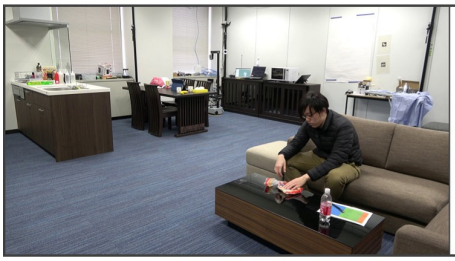
User request (collected by crowdsourcing)	System action (pre-defined)
Is there a place for family fun for a day?	Should I search for an amusement park around here?
I want to take a nap on the grass.	Should I search for a park around here?
I want to get some exercise.	Should I search for a sports facility around here?
I'd like to do something more interactive.	Should I search for an experience-based facility around here?
I want a Kyoto-style key chain.	Should I search for a souvenir shop around here?
Where can I see pandas?	Should I search for a zoo around here?
I haven't seen any penguins lately.	Should I search for an aquarium around here?
I want to relax in nature.	Should I search for a botanical garden around here?
I don't know where to go.	Should I search for a tourist information center around here?
It's suddenly getting cold. I need a jacket.	Should I search for a shopping mall around here?
I'm feeling sweaty and uncomfortable.	Should I search for a hot spring around here?
I'm interested in historical places.	Should I search for a temple around here?
I ' m not having a very good year.	Should I search for a shrine around here?
I'd like to see some famous buildings.	Should I search for a castle around here?
I'm feeling a bit depressed.	Should I search for some nature or landscape around here?
Since it's autumn, it's nice to experience art.	Should I search for an art museum around here?
Is there a tourist spot where I can also learn something?	Should I search for an history museum around here?
I'd love to walk around wearing a kimono.	Should I search for a kimono-rental shop around here?
I'd like to see some autumn colors.	Should I search for some fall colors around here?
I want to experience spring.	Should I search for cherry blossoms around here?
I want to take an interesting ride.	Should I search for a rickshaw around here?
It would be faster to go by train.	Should I search for a train station around here?
Walking is too time-consuming.	Should I search for a bus stop around here?
I'd like to sit down and relax.	Should I search for a rest area around here?
I'm having trouble getting good reception for my phone.	Should I search for a WiFi spot around here?
I want to relax.	Should I search for a quiet place around here?
I'd like to take a picture to remember this day.	Should I search for a beautiful place around here?
I'm looking for some places where my children can play.	Should I search for a fun place around here?
I want to feel free.	Should I search for a wide-open place around here?
I want to see a night view.	Should I search for a place with a nice view around here?

Table A.2.: User requests for all pre-defined system actions of restaurant search: Texts are translated from Japanese to English.

User request (collected by crowdsourcing)	System action (pre-defined)
I'm thirsty.	Should I search for a cafe around here?
I bought some delicious Japanese sweets!	Should I search for Japanese tea around here?
I'm too hot.	Should I search for shaved ice around here?
I'm bored with cake.	Should I search for Japanese sweets around here?
I feel like having an afternoon snack.	Should I search for western-style sweets around here?
I want something spicy!	Should I search for curry around here?
I'd like to have some home-cooking.	Should I search for traditional Kyoto food around here?
I want to eat something healthy.	Should I search for tofu cuisine around here?
I want to buy some breakfast for tomorrow.	Should I search for a bakery around here?
I think it's time for a snack.	Should I search for fast food around here?
I'm not really in the mood for rice.	Should I search for noodles around here?
It's cold today, so I'd like to eat something that will warm me up.	Should I search for Japanese stew around here?
I want to eat a rather heavy meal.	Should I search for rice bowls or fried food around here?
I've been eating a lot of Japanese food lately, and I'm getting a bit bored by it.	Should I search for some meat dishes around here?
I think I've been eating too much meat lately.	Should I search for sushi or fish dishes around here?
Let's go out for a meal together.	Should I search for flour-based foods around here?
I want to eat some typical Kyoto food.	Should I search for Kyoto cuisine around here?
My daughter wants to eat fried rice.	Should I search for Chinese food around here?
I'm not in the mood for Japanese or Chinese food today.	Should I search for Italian food around here?
I want to celebrate today.	Should I search for French food around here?
The kids are hungry and whiny.	Should I search for a child-friendly restaurant or family restaurant around here?
I want a quiet restaurant.	Should I search for tea-ceremony dishes around here?
I'm on a diet.	Should I search for Buddhist vegetarian cuisine around here?
I hear the vegetables are delicious around here.	Should I search for a vegetarian restaurant around here?
I want to go drinking in Kyoto!	Should I search for an izakaya or bar around here?
I want to eat so many things, and it's hard to decide.	Should I search for a food court around here?
When I travel, I get hungry in the morning.	Should I search for breakfast around here?
I don't have much money right now.	Should I search for an cheap restaurant around here?
I'd like a reasonably priced restaurant.	Should I search for an average-priced restaurant around here?
I'd like to have a luxurious meal.	Should I search for an expensive restaurant around here?

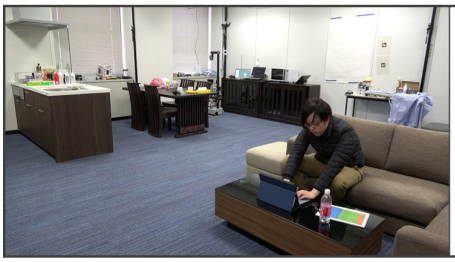
Table A.3.: User requests for all pre-defined system actions of app search: Texts are translated from Japanese to English.

User request (collected by crowdsourcing)	System action (pre-defined)
Nice view.	Should I launch the camera application?
What did I photograph today?	Should I launch the photo application?
I hope it's nice tomorrow.	Should I launch the weather application?
I want to get excited.	Should I launch the music application?
I'm worried about catching the next train.	Should I launch the transfer navigation application?
I have to tell my friends my hotel room number.	Should I launch the message application?
I wonder if XX is back yet.	Should I call XX?
The appointment is at XX.	Should I set your alarm clock?
I wonder what events are going on at XX right now.	Should I display the information about it?
How do we get to XX?	Should I search for a route to it?



uttr : ``I wonder if it' s enough."
action : bring a banana
viewpoint : 0
position : sofa
pose : sitting
has : (empty)
coffee table: bread, plastic bottle.
dining table: banana, cup, etc.
kitchen : olive oil, kitchen paper, etc.

(a)



uttr : ``I'm about to run out of charge."
action : bring a charging cable
viewpoint : 0
position : sofa
pose : sitting
has : (empty)
coffee table: laptop, plastic bottle
dining table: cup, charging cable, etc.
kitchen : ketchup, olive oil, etc.

(b)

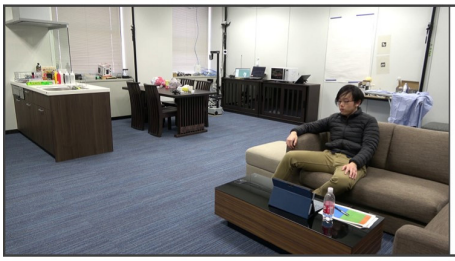
uttr : ``I don't have a cup."
action : bring a cup
viewpoint : 1
position : sofa
pose : sitting
has : (empty)
coffee table: laptop, plastic bottle
dining table: charging cable, etc.
kitchen : (empty)

(c)

uttr : ``It tastes bland."
action : bring ketchup
viewpoint : 2
position : sofa
pose : sitting
has : (empty)
coffee table: omelet, laptop
dining table: olive oil, cup, etc.
kitchen : water bottle, plastic bottle

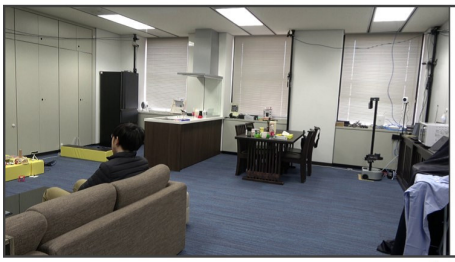
(d)

Figure A.1.: Interactions for all pre-defined robot actions (1). Texts were translated into English.



uttr : ``Where is my package?"
action : bring a package
viewpoint : 0
position : sofa
pose : sitting
has : (empty)
coffee table: laptop, plastic bottle, etc.
dining table: charging cable, cup, etc.
kitchen : ketchup, Saran wrap, etc.

(a)



uttr : ``My throat is getting dry."
action : bring a plastic bottle
viewpoint : 1
position : sofa
pose : sitting
has : (empty)
coffee table: (empty)
dining table: plastic bottle, cup, etc.
kitchen : (empty)

(b)

uttr : ``Where is the remote control?"
action : bring the remote control
viewpoint : 1
position : sofa
pose : sitting
has : (empty)
coffee table: (empty)
dining table: cup, Saran wrap, etc.
kitchen : (empty)

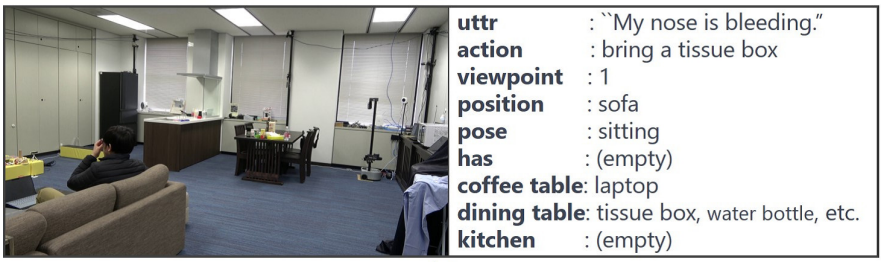
(c)

uttr : ``Where did I put my phone?"
action : bring the smartphone
viewpoint : 2
position : sofa
pose : sitting
has : (empty)
coffee table: laptop, plastic bottle
dining table: smartphone, snack, etc.
kitchen : plastic bottle

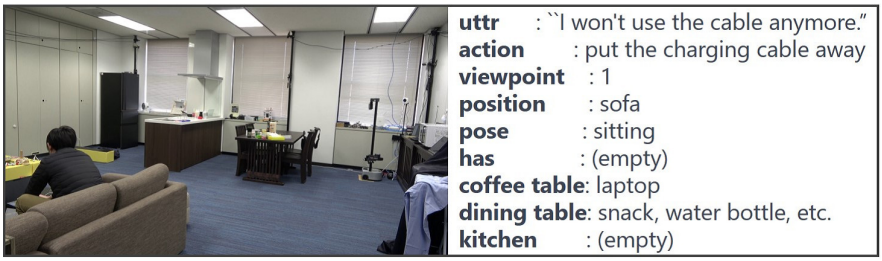
(d)

Figure A.2.: Interactions for all pre-defined robot actions (2). Texts were translated into English.

(a)



(b)



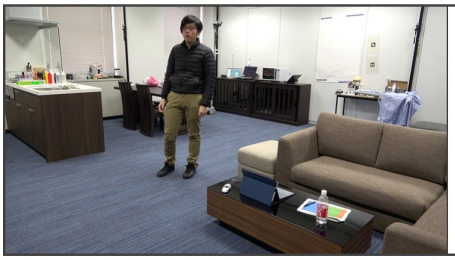
(c)

(d)

Figure A.3.: Interactions for all pre-defined robot actions (3). Texts were translated into English.

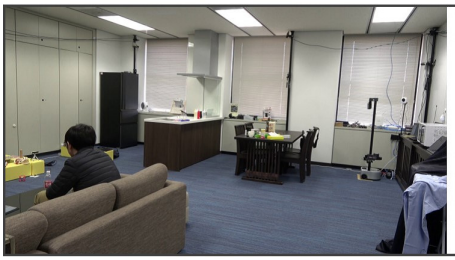
uttr : ``Thank you for the meal.''
action : put the ketchup away
viewpoint : 2
position : sofa
pose : sitting
has : (empty)
coffee table: ketchup, laptop, etc.
dining table: olive oil, kitchen paper, etc.
kitchen : plastic bottle

(a)



uttr : ``The kids have gone to bed.''
action : put the toy car away
viewpoint : 0
position : sofa
pose : standing
has : (empty)
coffee table: toy car, laptop, etc.
dining table: (empty)
kitchen : ketchup, olive oil, etc.

(b)



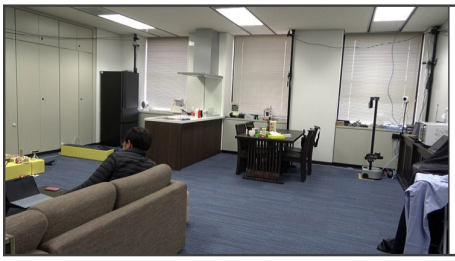
uttr : ``It's empty.''
action : put the plastic bottle away
viewpoint : 1
position : sofa
pose : sitting
has : (empty)
coffee table: plastic bottle
dining table: snack, cup, etc.
kitchen : (empty)

(c)

uttr : ``I've finished watching TV.''
action : put the remote control away
viewpoint : 0
position : sofa
pose : sitting
has : (empty)
coffee table: laptop, remote control, etc.
dining table: snack, cup, etc.
kitchen : ketchup, water bottle, etc.

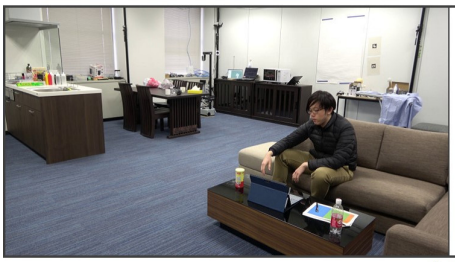
(d)

Figure A.4.: Interactions for all pre-defined robot actions (4). Texts were translated into English.



uttr : ``I'll clear the table.''
action : put the smartphone away
viewpoint : 1
position : sofa
pose : sitting
has : (empty)
coffee table: smartphone, laptop, etc.
dining table: cup, dining table, etc.
kitchen : (empty)

(a)

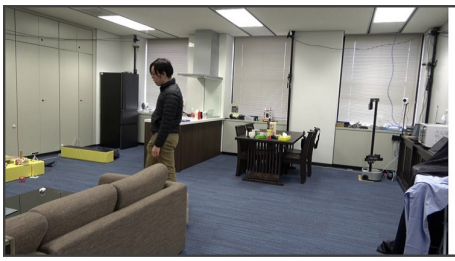


uttr : ``I finished eating.''
action : put the snack away
viewpoint : 0
position : sofa
pose : sitting
has : (empty)
coffee table: snack, laptop, plastic bottle
dining table: cup, charging cable, etc.
kitchen : ketchup, olive oil, etc.

(b)

uttr: ``I don't need the tissue box anymore.''
action : put the tissue box away
viewpoint : 1
position : sofa
pose : sitting
has : (empty)
coffee table: laptop, tissue box
dining table: cup, water bottle, etc.
kitchen : (empty)

(c)



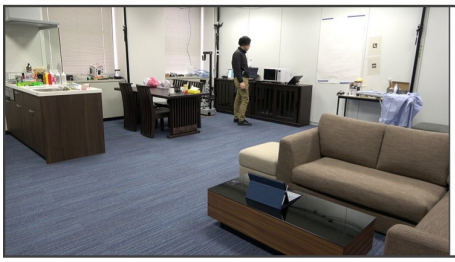
uttr : ``The trash is annoying.''
action : throw the trash away
viewpoint : 1
position : sofa
pose : standing
has : (empty)
coffee table: trash
dining table: cup, water bottle, etc.
kitchen : (empty)

(d)

Figure A.5.: Interactions for all pre-defined robot actions (5). Texts were translated into English.

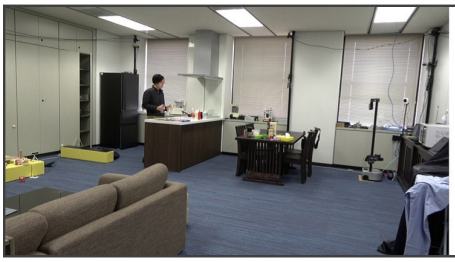
uttr : ``I forgot a can opener.''
action : bring a can opener
viewpoint : 2
position : kitchen
pose : standing
has : (empty)
coffee table: laptop
dining table: olive oil, kitchen paper, etc.
kitchen : plastic bottle

(a)



uttr : ``Let's start preheating the oven.''
action : bring a cooking sheet
viewpoint : 0
position : dining table
pose : standing
has : (empty)
coffee table: laptop
dining table: cup, snack, tissue box
kitchen : ketchup, olive oil, etc.

(b)



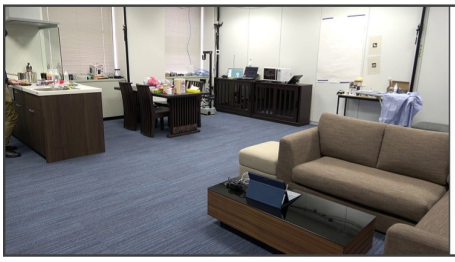
uttr : ``There isn't another one.''
action : bring a glass
viewpoint : 1
position : kitchen
pose : standing
has : glass, alcohol bottle
coffee table: (empty)
dining table: snack, cup, etc.
kitchen : (empty)

(c)

uttr : ``Let's grill Pacific saury.''
action : bring a grater
viewpoint : 2
position : kitchen
pose : standing
has : (empty)
coffee table: laptop
dining table: olive oil, kitchen paper
kitchen : plastic bottle

(d)

Figure A.6.: Interactions for all pre-defined robot actions (6). Texts were translated into English.



uttr : `` I'll use paper to absorb the oil."
action : bring some kitchen paper
viewpoint : 0
position : kitchen
pose : standing
has : (empty)
coffee table: laptop, etc.
dining table: kitchen paper, snack, etc.
kitchen : (empty)

(a)

uttr : ``I forgot a lemon."
action : bring a lemon
viewpoint : 2
position : kitchen
pose : standing
has : (empty)
coffee table: laptop
dining table: olive oil, snack, etc.
kitchen : plastic bottle

(b)

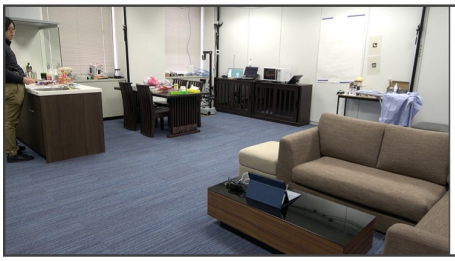
uttr : ``I need some olive oil first."
action : bring some olive oil
viewpoint : 1
position : kitchen
pose : standing
has : (empty)
coffee table: (empty)
dining table: glass, lemon, cup, etc.
kitchen : (empty)

(c)

uttr : ``There isn't another potato."
action : bring a potato
viewpoint : 2
position : kitchen
pose : standing
has : (empty)
coffee table: laptop
dining table: (empty)
kitchen : (empty)

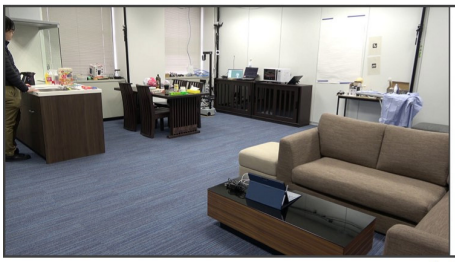
(d)

Figure A.7.: Interactions for all pre-defined robot actions (7). Texts were translated into English.



uttr : ``Where is Saran wrap?"
action : bring some Saran wrap
viewpoint : 0
position : kitchen
pose : standing
has : (empty)
coffee table: laptop
dining table: Saran wrap, cup, etc.
kitchen : plastic bottle, etc.

(a)



uttr: ``I need to fill my water bottle with tea."
action : bring a water bottle
viewpoint : 0
position : kitchen
pose : standing
has : (empty)
coffee table: charging cable, laptop
dining table: water bottle, cup, etc.
kitchen : plastic bottle, etc.

(b)

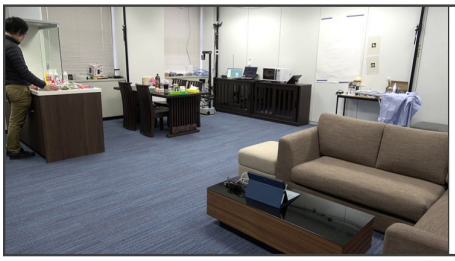
uttr: ``The can don't need a can opener."
action : put the can opener away
viewpoint : 1
position : kitchen
pose : standing
has : (empty)
coffee table: (empty)
dining table: cup, snack, etc.
kitchen : (empty)

(c)

uttr : ``I thought it was kitchen paper."
action : put the cooking sheet away
viewpoint : 2
position : kitchen
pose : standing
has : (empty)
coffee table: laptop
dining table: (empty)
kitchen : (empty)

(d)

Figure A.8.: Interactions for all pre-defined robot actions (8). Texts were translated into English.



uttr : ``I'm done washing the dishes.''
action : put the glass away
viewpoint : 0
position : kitchen
pose : standing
has : (empty)
coffee table: laptop, charging cable
dining table: olive oil, cup, etc.
kitchen : glass

(a)

uttr : ``That's all.''
action : put the grater away
viewpoint : 2
position : kitchen
pose : standing
has : (empty)
coffee table: laptop
dining table: olive oil, kitchen paper
kitchen : grater, plastic bottle

(b)

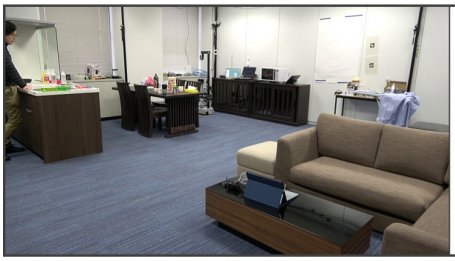
uttr : ``I don't use it now.''
action : put the kitchen paper away
viewpoint : 2
position : kitchen
pose : standing
has : (empty)
coffee table: laptop
dining table: olive oil
kitchen : kitchen paper

(c)

uttr : ``I'll drink the rest later.''
action: put the plastic bottle in the refrigerator
viewpoint : 1
position : kitchen
pose : standing
has : (empty)
coffee table: (empty)
dining table: cooking sheet, cup, etc.
kitchen : plastic bottle

(d)

Figure A.9.: Interactions for all pre-defined robot actions (9). Texts were translated into English.

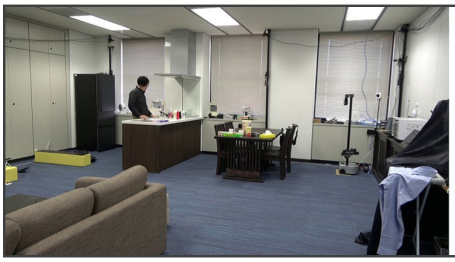


uttr : ``I have to clean up what I've used.''
action : put the Saran wrap away
viewpoint : 0
position : kitchen
pose : standing
has : (empty)
coffee table: laptop, charging cable
dining table: olive oil, cup, etc.
kitchen : Saran wrap, plastic bottle

(a)

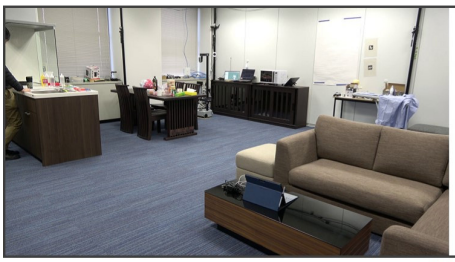
uttr: ``I'd be happy with warm rice in the Tupperware.''
action: put the Tupperware in the microwave
viewpoint : 0
position : kitchen
pose : standing
has : Tupperware
coffee table: laptop, charging cable
dining table: olive oil, can, etc.
kitchen : (empty)

(b)



uttr : ``Okay, now we're ready.''
action: put the Tupperware in the refrigerator
viewpoint : 1
position : kitchen
pose : standing
has : (empty)
coffee table: (empty)
dining table: snack, cup, etc.
kitchen : Tupperware

(c)



uttr : ``I'm off tomorrow.''
action : put the water bottle away
viewpoint : 0
position : kitchen
pose : standing
has : (empty)
coffee table: charging cable, laptop
dining table: olive oil, cup, etc.
kitchen : water bottle, plastic bottle

(d)

Figure A.10.: Interactions for all pre-defined robot actions (10). Texts were translated into English.

Publication List

Referred Journals

[1] **Shohei Tanaka**, Konosuke Yamasaki, Akishige Yuguchi, Seiya Kawano, Satoshi Nakamura, Koichiro Yoshino. “Do As I Demand, Not As I Say: A Dataset for Developing Reflective Life-Support Robot” IEEE Access, Submitted
[2] **Shohei Tanaka**, Koichiro Yoshino, Katsuhito Sudoh, Satoshi Nakamura. “Reflective Action Selection Based on Positive-Unlabeled Learning and Causality Detection Model” Computer Speech and Language (CSL), Volume 78, March 2023
[3] **田中翔平**，吉野幸一郎，須藤克仁，中村哲．“雑談対話応答における連続する事態の一貫性と対話継続性の関係” 自然言語処理 (TNLP)，pp. 26-59，28 巻 1 号，3 月，2021 年

Referred Conferences

[4] **Shohei Tanaka**, Koichiro Yoshino, Katsuhito Sudoh, Satoshi Nakamura. “ARTA: Collection and Classification of Ambiguous Requests and Thoughtful Actions” The 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pp. 77-88, Singapore (Online), July 2021, Oral
[5] **Shohei Tanaka**, Koichiro Yoshino, Katsuhito Sudoh, Satoshi Nakamura. “Conversational Response Re-ranking Based on Event Causality and Role Factored Tensor Event Embedding” The 1st Workshop NLP for Conversational AI ACL 2019 Workshop (ConvAI), pp. 51-59, Florence, Italy, August 2019, Oral & Poster, **Best Paper Award**

Unreferred Conferences

[6] **田中翔平**，山崎康之介，湯口彰重，河野誠也，中村哲，吉野幸一郎．“観測した周囲の状況を曖昧な発話に統合した対話ロボットによる気の利いた行動選択” 言語処理学会第 29 回年次大会 (ANLP)，pp. 1377-1382, 沖縄，3 月，2023 年，口頭発表

[7] **田中翔平**，湯口彰重，河野誠也，中村哲，吉野幸一郎．“気の利いた家庭内ロボット開発のための曖昧なユーザ要求と周囲の状況の収集” 情報処理学会 第 253 回自然言語処理研究会 (SIGNL)，京都 (オンライン) ，9 月，2022 年，口頭発表

[8] **田中翔平**，吉野幸一郎，須藤克仁，中村哲．“曖昧なユーザ要求に対する因果関係知識を用いた気の利いたシステム行動の選択” 言語処理学会第 28 回年次大会 (ANLP)，pp. 1084-1089, 静岡 (オンライン)，3 月，2022 年，口頭発表

[9] **田中翔平**，吉野幸一郎，須藤克仁，中村哲．“曖昧な要求と気の利いた応答を含む対話コーパスの収集と分類” 言語処理学会第 27 回年次大会 (ANLP)，pp. 359-364, 福岡 (オンライン)，3 月，2021 年，ポスター

[10] **田中翔平**，吉野幸一郎，須藤克仁，中村哲．“対話エージェントの機能に着目した気の利いた応答を含むコーパスの収集” 人工知能学会 第 90 回言語・音声理解と対話処理研究会 (SIGSLUD)，東京 (オンライン)，11 月，2020 年，ポスター，**若手萌芽賞**

[11] **田中翔平**，吉野幸一郎，須藤克仁，中村哲．“連続する事態の一貫性に基づく雑談対話応答のリランキングにおける事例分析” 言語処理学会第 26 回年次大会 (ANLP)，pp. 1316-1319, 茨城 (オンライン)，3 月，2020 年，ポスター

[12] **田中翔平**，吉野幸一郎，須藤克仁，中村哲．“事態の一貫性推定に基づく雑談対話応答選択モデル” 人工知能学会 第 87 回言語・音声理解と対話処理研究会 (SIGSLUD)，東京，12 月，2019 年，ポスター

[13] **田中翔平**，吉野幸一郎，須藤克仁，中村哲．“因果関係と事態分散表現を用いた雑談対話応答のリランキングにおける傾向分析” 情報処理学会 第 241 回自然言語処理研究会 (SIGNL)，北海道，8 月，2019 年，口頭発表

[14] **田中翔平**，吉野幸一郎，須藤克仁，中村哲．“因果関係を用いた雑談対話応答のリランキングの評価” 言語処理学会第 25 回年次大会 (ANLP)，pp. 1026-1029, 名古屋，3 月，2019 年，口頭発表

[15] **田中翔平**，吉野幸一郎，須藤克仁，中村哲．“因果関係を用いた雑談対話応答のリランキング” 人工知能学会 第 84 回言語・音声理解と対話処理研究会 (SIGSLUD) 東京，11 月，2018 年，ポスター