# Semantic Content Prediction for Generating Interviewing Dialogues to Elicit Users' Food Preferences

**Jie Zeng**
Graduate School of Science
and Technology, Seikei University
dd196201@cc.seikei.ac.jp

**Tatsuya Sakato** and **Yukiko I. Nakano**
Department of Computer and
Information Science, Seikei University
{sakato, y.nakano}
@st.seikei.ac.jp

## Abstract

Dialogue systems that aim to acquire user models through interactions with users need to have interviewing functionality. In this study, we propose a method to generate interview dialogues to build a dialogue system that acquires user preferences for food. First, we collected 118 text-based dialogues between the interviewer and customer and annotated the communicative function and semantic content of the utterances. Next, using the corpus as training data, we created a classification model for the communicative function of the interviewer's next utterance and a generative model that predicts the semantic content of the utterance based on the dialogue history. By representing semantic content as a sequence of tokens, we evaluated the semantic content prediction model using BLEU. The results demonstrated that the semantic content produced by the proposed method was closer to the ground truth than the semantic content transformed from the output text generated by the retrieval model and GPT-2. Further, we present some examples of dialogue generation by applying model outputs to template-based sentence generation.

## 1 Introduction

Traditionally, dialogue systems have been characterized in terms of whether they are task- or non-task-oriented. In task-oriented dialogue systems, such as an airline ticket reservation system (Hemphill et al., 1990), eliciting specific information from the user, such as the date, time, and destination of the flight, is an important functionality for completing the task. However, in non-task-oriented dialogue systems, the system does not have a clear goal of eliciting information from the user, and the content of the dialogue is free.

In this study, as another type of dialogue system, we focus on interviewing systems, in which the goal is to acquire a user model through a flexible flow of dialogue. Specifically, we propose a method for interviewing a user's preference for food. To generate such dialogues, the system must be able to generate appropriate questions to elicit the user's preferences for food while touching on various topics in the food domain, such as how to eat, how to cook, etc., without limiting the content of the dialogue as a task-oriented dialogue does.

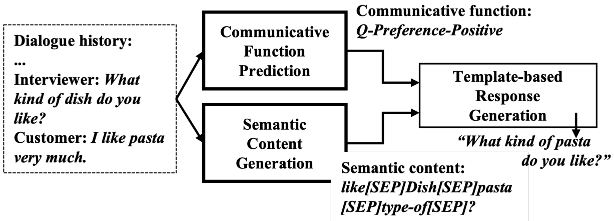
Dialogue history:
...
Interviewer: *What kind of dish do you like?*
Customer: *I like pasta very much.*
Communicative Function Prediction
Semantic Content Generation
Communicative function:
*Q-Preference-Positive*
Template-based Response Generation
*"What kind of pasta do you like?"*
Semantic content:
*like[SEP]Dish[SEP]pasta [SEP]type-of[SEP]?*

Figure 1: Overview of the proposed method: taking dialogue history as input, a model predicts the interviewer's intent (communicative function), another model decides the content of the utterance (semantic content), and the outputs of these models are combined to generate a response. (For details, refer to Section 4)

One possible approach for achieving the requirements discussed above is end-to-end neural network, where dialogue generation is the task of predicting the next utterance using dialogue history as input (Vinyals and Le, 2015; Serban et al., 2016). This method is widely used to generate open-domain dialogues, such as chitchats. However, it requires a large amount of dialogue data to learn the model. Otherwise, less informative and contextually inappropriate utterances are frequently generated. To overcome this drawback, we propose a method that first determines the intention and semantic content of the interviewer's next utterance and then combines these to generate questions from the interviewer.

Figure 1 shows the proposed approach. First, we trained two models. The first is a classification model that takes the dialogue history as input and determines the interviewer's intention for the next utterance. The second is a generator model, which

*Proceedings of the 2nd Workshop on When Creative AI Meets Conversational AI (CAI2), COLING 2022*, pages 48–58
October 12–17, 2022.

also takes the dialogue history as input and outputs the semantic content of the utterance, including the target (e.g., dish or ingredient) mentioned in the utterance and its related information (e.g., taste or how to eat). Next, a template for sentence generation is selected based on these two outputs, and they are applied to the selected template to generate sentences. Compared to learning a model that directly generates a surface expression, the models for predicting the intent and semantic content of an utterance can be learned using a smaller amount of data. Additionally, because the content of an utterance is determined based on the context obtained from the dialogue history, appropriate utterances that are related to the preceding utterances can be generated.

The contributions of this study are as follows:

- Collection of 118 text-based dialogues for interviewing food preferences.
- Proposal of an annotation schema for utterance intention and semantic content of utterances, and creation of a dataset with these annotations.
- Creation of a classification model for utterance intention and a generative model of semantic content of utterances.
- Demonstration of the effectiveness of the proposed method using an automated evaluation method.
- Presentation of examples of dialogues generated by the proposed method, and discussion of the quality of the dialogues.

## 2 Related Work

Task-oriented dialog systems are typically designed to collect information from users. For example, previous studies have proposed an airline ticket reservation system (TIS) (Hemphill et al., 1990), a restaurant reservation system (Henderson et al., 2014), and interview systems to collect information, such as public opinion polls and class evaluation interview systems (Johnston et al., 2013; Stent et al., 2006). In these systems, the purpose of the dialogue is to obtain information to accomplish a predefined task.

Meanwhile, chitchat does not have a clear goal as a task-oriented dialogue does, but this type of dialogue has the potential to elicit a variety of information from the user. For example, the system asks follow-up questions such as "Please tell me more about the *keyword*" by using a keyword from the user's preceding utterance. To improve such interviewing functionality, relevant topics and questions should be selected and the dialogue strategies should be modified. To address these issues, we propose a method to determine the target object and semantic content of the system response based on the dialogue context.

Previous studies on dialogue generation have proposed different techniques to generate task- and non-task-oriented dialogue. Early studies on generating open-domain chitchat proposed DNN-based techniques to generate system responses by exploiting the data-driven approach (Sordoni et al., 2015a; Vinyals and Le, 2015; Serban et al., 2016). Recent studies have proposed incorporating useful information (that is relevant to the domain) and responses into the model, thus improving the quality of generated responses (Li et al., 2018). Some studies have exploited word-based information, such as nouns extracted from the user's preceding utterances and a set of keywords predicted to be used in the response (Serban et al., 2017; Xu et al., 2021). Other studies have used knowledge ontologies, including commonsense (Wu et al., 2020; Zhang et al., 2020; Moon et al., 2019; Galetzka et al., 2021). However, these end-to-end methods, in which training models directly generate system responses, require a large amount of training data, and our corpus was not sufficiently large for this approach.

In traditional task-oriented dialogue systems, the information required to achieve the dialogue goals is limited to the task domain. Therefore, the internal state of the system is defined as a slot–value pair, and the system generates responses through the following modules: a) understanding the user's utterance, b) determining the system action (e.g., the intention and the slot–value as the utterance content) based on the internal state, and c) generating a response sentence from the system action. The action of the system is determined by rule-based, statistical-based (Young et al., 2010), deep learning (Chen et al., 2019) and reinforcement learning approaches (Sankar and Ravi, 2019).

In this study, we exploited the approach described above, which represents the interviewer's utterance as structured semantic content composed of the intent of the utterance, the objects mentioned in the utterance, and their attributes and values. We created a machine learning model to predict these types of information and generate responses based on the determined actions.

## 3 Data Collection and Dataset Making

This study aims to generate interview dialogues that elicit information about users' food preferences. For this purpose, we collected role-play conversations between an interviewer and a customer and constructed a corpus from the collected conversations.

### 3.1 Interview Dialogue Collection

Subject pairs were created with participants recruited by crowdsourcing. One subject was assigned the role of an interviewer and the other, the role of a customer. They conducted a text-based chat session in Japanese on the web. After typing an utterance and pressing the send button, the message was added to the chat screen. They were also instructed to take turns sending the messages.

The participants playing as interviewers were requested to engage in conversations to elicit food preferences from customers. The participants playing as customers were asked to indicate their food preferences. We allowed the customers to respond to their real preferences or to pretend to be someone else.

After the dialogue, each participant answered a questionnaire. The interviewers were asked to describe the client's food preferences obtained from the conversation, and the dishes they would like to recommend to the customer. The customers were asked to describe the food preferences they expressed in the dialogue. They were also asked to describe the dishes they would like the interviewer to recommend to them.

To create a dialogue model capable of generating responses that considered the interviewer's dialogue strategy and dialogue history, we requested the participants to input at least 20 turns from each party and 40 turns in total. This was a task completion requirement.

### 3.2 Annotation

Structured semantic labels were assigned to classify the interviewees' utterances and understand their semantic content. Following the idea of structured semantic labels discussed in the Dialogue Act annotation (Bunt et al., 2012), we represented each utterance as a combination of communicative function and semantic content.

More specifically, a dialog consists of messages sent by the user in the chat, and one message may include multiple sentences. We annotated each sentence in interviewer's message. To annotate sentences in the interviewer's message in our corpus collected in Section 3.1, we first defined labels for communicative function and semantic content.

**Communicative Function:**

We defined 32 labels for the communicative functions based on those for SWBD-DAMSL (Jurafsky, 1997) and Meguro et al. (2014). We used SWBD-DAMSL to label backward utterances, including understanding, answer, and agreement (Appendix A). For self-disclosure (SD) and questions (Q), we used labels defined in the Meguro et al. (2014) as references and added new labels such as preferences, experiences, and habits. For the preference labels, we added the polarity: positive, negative, and neutral.

**Semantic Content:**

The semantic content expresses the meaning of a sentence, whereas the communicative function specifies the intention of a sentence, as discussed above. In our corpus, many of the interviewer's questions referred to the name of the dish and its ingredients, tastes, recipes, and how to eat. Based on this observation, we defined semantic content as a combination of utterance objects (e.g., dishes and ingredients) and their attributes (e.g., tastes and cooking methods).

Figure 2 shows the structure of the semantic content and list of values for <*verb*>, <*ObjectType*>, and <*ObjectAttribute*>. Two examples of semantic content were assigned to an interviewer sentence.

In Example A *"I ate hot curry"* in Figure 2, the verb is *"eat"* and its object is *"hot curry"*. The object is the first argument (*argument_1*) of the *verb:eat*, and the relationship between this verb and the object is expressed as a verb frame.

**verb frame:**

***<verb>:*** We defined five verbs that are frequently used in conversations regarding food. They consider direct objects as arguments. We also defined negative forms for them by adding "!". For example, the negative form for "like" is "!like." In addition to these 10 verbs, "think" and "other" were added, and 12 verbs were defined in total.

**object-features:**

We defined four types of features for an object. These are *ObjectType*, *ObjectName*, *ObjectAttribute*, and *AttributeValue*. These are

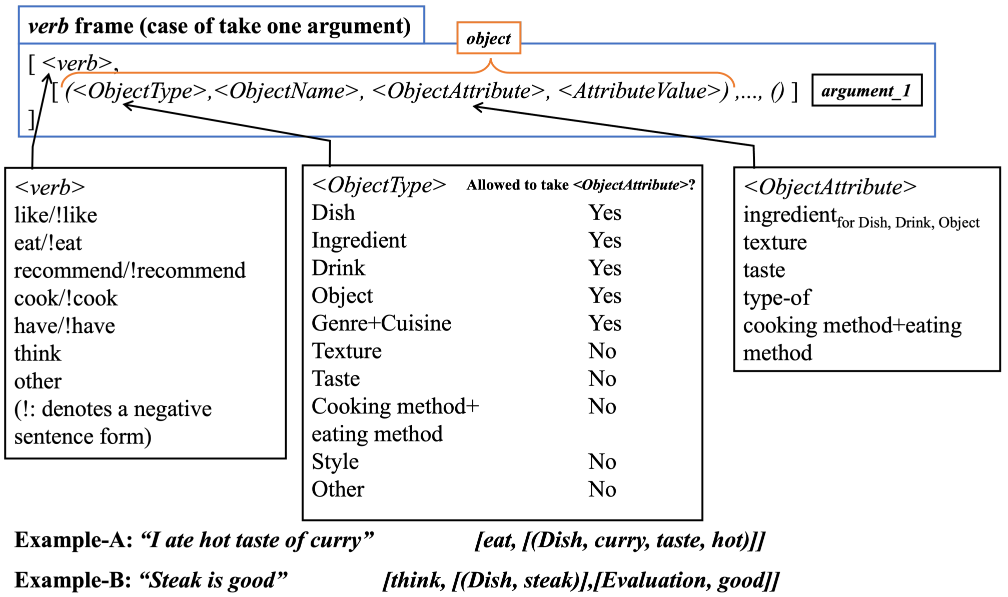
verb frame (case of take one argument)
object
[ <verb>,
[ (<ObjectType>,<ObjectName>, <ObjectAttribute>, <AttributeValue>) ,..., () ]
argument_1
]
<verb>
like/!like
eat/!eat
recommend/!recommend
cook/!cook
have/!have
think
other
(!: denotes a negative sentence form)
<ObjectType>
Allowed to take <ObjectAttribute>?
Dish Yes
Ingredient Yes
Drink Yes
Object Yes
Genre+Cuisine Yes
Texture No
Taste No
Cooking method+ eating method No
Style No
Other No
<ObjectAttribute>
ingredient for Dish, Drink, Object
texture
taste
type-of
cooking method+eating method
Example-A: "I ate hot taste of curry" [eat, [(Dish, curry, taste, hot)]]
Example-B: "Steak is good" [think, [(Dish, steak)],[Evaluation, good]]

Figure 2: Structure of semantic content and values for <*verb*>, <*ObjectType*>, and <*ObjectAttribute*>. Two examples of interviewer sentence and its semantic content are shown at the bottom of the figure.

called the object features. The *"hot curry"* is an object of the verb *'eat'*. It contains a set of features: *ObjectType='Dish'*, *ObjectName='curry'*,*ObjectAttribute='taste'*, and *AttributeValue='hot'*. We simply expressed this set as *(Dish, curry, taste, hot)*. Details of the object features are presented below.

***<ObjectType>***: We defined 10 object types: Dish, Ingredient, and Drink. Each name begins with a capital letter. For example, *"Dish"* is assigned as the *ObjectType* value for *curry*, *"Ingredient"* for *carrot*, and *"Genre+Cuisine"* for *Indian food*.

***<ObjectName>***: This feature indicates the name of the target object in an interviewer's sentence.

***<ObjectAttribute>***: As shown in Example-A in Figure 2, there are many detailed questions and utterances about the target object, such as the taste of the food, its recipe, and how to eat it. We believe that such information is important for food preferences. To include it in the semantic content, we defined the attributes of objects with a specific *ObjectType*. The values of these attributes are described later in this study.

***<AttributeValue>***: The value for the *ObjectAttribute* is specified in this section. A set of possible values is not defined, and the value is freely specified, as in *ObjectName*.

For example, the *ObjectType* of *"hot curry"* is a *'Dish'*, and *ObjectType='Dish'* can take an *ObjectAttribute* (see Figure 2, Allowed to take <*ObjectAttribute*>?: Yes). Then, "hot" belongs to *"taste"*, which is defined as an *ObjectAttribute*. As a result, *"hot curry"* is interpreted as an object feature. *ObjectType='Dish'*, *ObjectName='curry'*, *ObjectAttribute='taste'*, *AttributeValue='hot'*.

When the interviewer's utterance is a question, such as a Yes/No question or WH question, the object of the question is indicated as a *'?'*. For example, in the WH question, "What taste of curry do you like?", the *AttributeValue* for *ObjectAttribute='taste'* is the target of this question. In this case, the semantic content is described as *[like, [(Dish, curry, taste, ?)]]* .

For a Yes/No question, where (default) values are already assigned, the features are described as *ObjectName+?* and *AttributeValue+?*. For example, the semantic content for *"Do you like curry hot?"* is described as *[like, [(Dish, curry, taste, hot?)]]*

Some sentences, such as "Steak is good" (Example-B in Figure 2), express an evaluation of the target object. In such a case, "think" is assigned to (<*verb*>), and two arguments are used; the object information is described in *argument_1* and the evaluation in (*argument_2*). In this example, *argument_2* describes a pair of values: "Evaluation" and the (<*EvaluationValue*>) denoting the value of the evaluation. Thus, (*argument_2*) is [Evaluation, good].

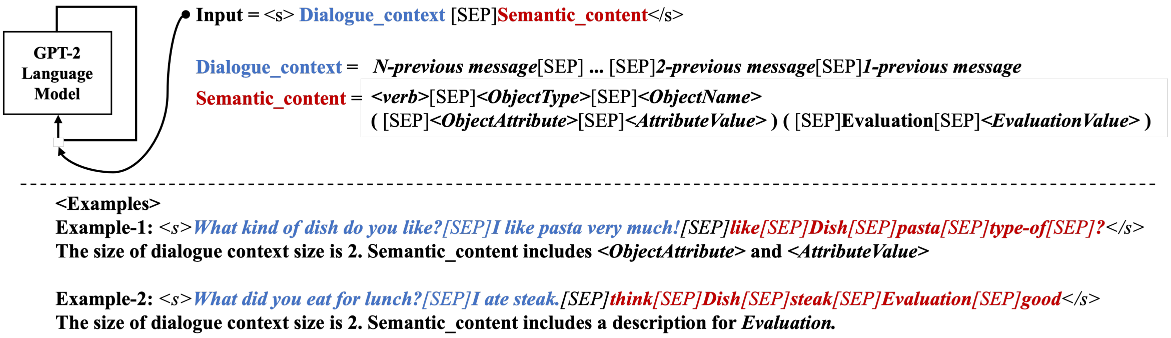
GPT-2 Language Model
Input = <s> Dialogue_context [SEP]Semantic_content</s>
Dialogue_context = N-previous message[SEP] ... [SEP]2-previous message[SEP]1-previous message
Semantic_content = <verb>[SEP]<ObjectType>[SEP]<ObjectName>
( [SEP]<ObjectAttribute>[SEP]<AttributeValue> ) ( [SEP]Evaluation[SEP]<EvaluationValue> )
<Examples>
Example-1: <s>What kind of dish do you like?[SEP]I like pasta very much![SEP]like[SEP]Dish[SEP]pasta[SEP]type-of[SEP]?</s>
The size of dialogue context size is 2. Semantic_content includes <ObjectAttribute> and <AttributeValue>
Example-2: <s>What did you eat for lunch?[SEP]I ate steak.[SEP]think[SEP]Dish[SEP]steak[SEP]Evaluation[SEP]good</s>
The size of dialogue context size is 2. Semantic_content includes a description for Evaluation.

Figure 3: Training a Semantic Content Generation (SCG) model: The description in the parentheses of Semantic_content is used when the target sentence includes the corresponding information.

## 4 Models

With the goal of building a dialogue system that generates the interviewer's appropriate questions to acquire the customer's food preferences, we present two machine learning models in this section for communicative function prediction and semantic content generation.

### 4.1 Semantic Content Generation (SCG)

As part of the interviewing system, we created a Semantic Content Generation (SCG) model that generates the semantic content of the interviewer's next sentence. The model takes the history of messages of both the interviewer and customer as input and predicts the semantic content of the last sentence in the next interviewer's message [1]. The representation of semantic content follows the annotation scheme described in Section 3.2.

To train the SCG model, we used a pre-trained Japanese language model [2] of the Transformer-based GPT-2 model (Radford et al., 2019), which is commonly used for conversation generation and fine-tuned it using our own small dataset described in Section 3.1.

Figure 3 illustrates GTP-2 fine tuning to create the SCG model. Each sample of the training data is a pair of dialogue context and semantic content of the interviewer's next sentence. As the dialogue context, messages preceding the prediction target sentence are concatenated. The end of each context message is indicated by [SEP] special token. The maximum number of context messages is five. This sequence is concatenated with the semantic content of the prediction target (the interviewer's sentence) and fed to GPT-2.

The semantic content is represented as a sequence of tokens: verb, object-features, and evaluation description if necessary. Example-1 in Figure 3 shows an example of object-features consisting of *ObjectAttribute* and *AttributeValue*, in which the semantic content of the interviewer's next sentence is *"[like, [(Dish, pasta, type-of, ?)]]"* (original sentence: "What kind of pasta do you like?"). The *verb*, *ObjectType*, *ObjectName*, *ObjectAttribute*, and *AttributeValue* are concatenated into a sequence. Each of these is separated by a [SEP]. Additionally, the <s> and </s> tokens indicate the beginning and end of each sample, respectively. In Example-2, the semantic content contains the evaluation part: *"[think, [(Dish, steak)], [Evaluation, good]]"* (original sentence: "Steak is good."), where the second argument *[Evaluation, good]* is added.

Each input sequence is tokenized by the tokenizer, and GPT-2 optimizes the model weights by minimizing the negative log-likelihood for the next-token prediction.

### 4.2 Communicative Function Prediction (CFP)

This section proposes a Communicative Function Prediction (CFP) model that predicts the communicative function label to specify the intention of the next interviewer's message, such as self-disclosure and questions.

A fine-tuning approach was employed to train the CFP model. We used the BERT (Devlin et al., 2019) Japanese pre-trained model[3].

[1] When the next interviewer message consists of multiple sentences, the semantic content of the last sentence is used as the prediction target. This is because the main assertion of the message is often made in the last sentence.

[2] japanese-gpt2-small: https://huggingface.co/rinna/japanese-gpt2-small

[3] BERT base Japanese: https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

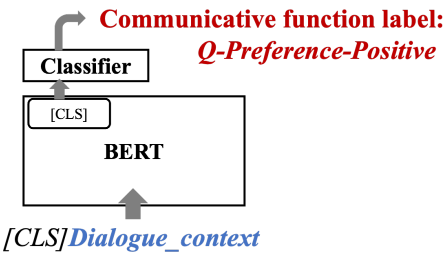

Figure 4: Training a model for communicative function prediction (CFP)

As demonstrated in Figure 4, the input is a dialogue context consisting of multiple previous messages concatenated using [SEP]. This sequence is the same as that used to train the SCG model in Section 4.1. Using this sequence as the input, we trained a model that predicted the communicative function label of the interviewer's next message.

We use the representation of the final layer of the special classification token ([CLS]), which is placed at the beginning of the input, as the input for a downstream classification task. As described in Section 5.1, the communicative function classifier predicts 7 labels, reduced from the 32 labels presented in Section 3.2.

## 5 Experiments and Evaluation

### 5.1 Detail of Dataset

Table 1(top) lists the details of the corpus collected in Section 3. Table 1(bottom) shows the number of instances [4] that was used to train the CFP and SCG models. The dataset was divided into train/valid/test sets at a ratio of 7:1:2.

Although we defined 32 communication function labels in the original dataset, many of them were not frequently observed. Thus, we merged the labels whose frequency was lower than 20% of all samples and used the seven labels listed in Table 2 in this experiment.

We calculated the inter-coder reliability using three dialogues annotated by two coders. For the seven labels of communicative function, Cohen's kappa was 0.75, which indicated substantial agreement. For semantic content, which is a combination of verb and object-features, the percentage of agreement was 0.72. Because we achieved a sufficient agreement level, the remaining data were annotated by either coder.

Table 1: Details of the interview dialogue corpus collected (top) and number of instances used to train the CFP and SCG models (bottom).

| | |
|---|---|
| # dialogues | 118 |
| # messages | 4871 |
| - interviewer | 2471 |
| - customer | 2400 |
| # sentences | 8921 |
| - interviewer | 4647 |
| - customer | 4274 |

| | train/validation/test |
|---|---|
| # dialogues | 84 / 10 / 24 |
| # instances for communicative function | 1735 / 209 / 482 |
| # instances for semantic content | 1663 / 205 / 458 |

Table 2: Merged communicative function labels

| | |
|---|---|
| SD-Fact&Experience | Q-Fact&Experience |
| Q-Habit | Q-Preference-Positive |
| Q-Preference-Neutral | Reply |
| Other | |

### 5.2 Baselines

We compared the proposed models with two baseline models: the retrieval model and text generation model.

**Retrieval Model:** We simply applied a technique used in information retrieval to a response selection, as proposed in (Ritter et al., 2011; Sordoni et al., 2015b). The customer's message and the interviewer's response to it were paired as an input–response pair. In the response selection process, among all pairs, the one whose input sentence had the highest similarity to the customer's input was selected, and the response part of this pair was used as the system's (interviewer's) response. The sentence vector was a hidden representation of the [CLS] token obtained from BERT, and cosine similarity was used to calculate the sentence similarity.

**Text Generation Model:** A GPT-2 language model was trained using pairs of dialogue context and the next interviewer's sentence. The difference from the SCG model is that the dialogue context was paired with the text (not the semantic content) of the interviewer's response. Therefore, this model generated an interviewer's response text rather than

[4] Messages that were not related to the task (e.g., greetings at the beginning of the task, gratitude at the end of the task) were excluded from the dataset.

Table 3: Average BLEU-4 scores. Numbers in parentheses indicate the length of the dialogue history in the best model using the validation dataset. In the retrieval model, the length of the dialogue history was set to one.

| Model | BLEU-4 score<br>(standard deviation) |
|---|---|
| Retrieval | 11.5 (20.6) |
| Text Generation (N=4) | 13.0 (22.3) |
| SCG (Proposed) (N=3) | **17.3** (24.7) |

the semantic content of the sentence.

### 5.3 Automated Evaluation for SCG

To evaluate the output produced by the models, we conducted an automated evaluation using the BLEU with respect to the semantic content. For this purpose, we treated the semantic content of the target interviewer's sentence as a sequence of words (e.g., *"like[SEP]Dish[SEP]pasta[SEP]type-of[SEP]?"*) and used it as the ground truth.

For the SCG model, the BLEU score was calculated by comparing the generated semantic content with the ground truth. For the retrieval model, the semantic content annotation for the response part was compared to the ground truth. For the text generation model, the semantic content was assigned by annotating the generated message and comparing it with the ground truth to calculate the BLEU score.

As an evaluation of semantic content consisting of a combination of the verb and object-features, we show the average of BLEU scores using 4-grams in the test set in Table 3. The proposed model achieved the highest BLEU score. We changed the dialogue context length from 1 to 5 and found that a model with a dialogue context length of three achieved the best performance in the validation dataset. These results suggest that the proposed SCG model performed the best in reproducing the semantic content of the interviewer's message.

### 5.4 Performance of CFP

We evaluated the performance of the CFP model by setting the length of the context to three as this setting performed best in the SCG model. The results showed that the model performance for the seven-classes classification was 0.39 in accuracy and 0.30 in weighted average of the F1 score.

### 5.5 Samples of Generated Response

In this section, we present examples of the responses generated by our interview system. We first describe the template-based response-generation mechanism and then discuss examples of interview generation.

**Template-based Response Generation**

As shown in Figure 1, the system receives outputs from the SCG and CFP models and generates the interviewer's responses using the template-based generation method.

Suppose that the outputs from the two prediction models are as follows:
communicative function label: *Q-Preference-Positive*
semantic content: *like[SEP]Dish[SEP]pasta[SEP] type-of[SEP]?*

By referring to this information: *communicative function='Q-Preference-Positive'*, *verb='like'*, *ObjectAttribute='type-of'*, and *AttributeValue='?'*, the system selects a template: "*{ObjectName} no Shurui de Nani ga Sukidesuka?*" (in English, "What kind of *{ObjectName}* do you like?"). Then, a response sentence is generated by replacing *{ObjectName}* with the value *'pasta'*.

**Discussion on Generated Responses**

Table 4 presents the sequence of five context utterances and the interviewer's utterance which follows the context. "Human" is the real interviewer utterance (ground truth). "Retrieval," "Text Generation," and "Proposed" are the outputs by the methods examined in our experiment.

In Dialogue-1 in Table 4, the interviewer utterance generated by the retrieval model asks whether the user eats vegetables. This utterance is not appropriate because in previous-3, the customer had already said that he/she eats vegetables. By contrast, the proposed model generated a question to elicit more information according to the current context of the hot-pot dish by asking the favorite ingredients for the dish.

In Dialogue-2 in Table 4, all three models failed to generate an utterance about the current topic focus (cheese), but the retrieval and text generation models still successfully generated a natural response. However, the utterance generated by the proposed model appears to be abrupt. This is because the selected template was not appropriate or expressive. Providing more templates and improving the template selection mechanism are necessary to generate more expressive responses.

Table 4: Two dialogue examples. Each table contains 5 messages (previous-5 to -1) preceding the prediction target interviewer's sentence, human ground truth responses (Human), and model outputs: Retrieval, Text Generation, and Proposed system. I/C indicates interviewer and customer.

| | Dialogue-1 |
|---|---|
| previous-5 (C) | It is strange, isn't it? We hear about rice cake accidents at least two or three times every year, don't we? I would be happy if hormones were a little easier to eat as well, since motsu (pork guts) are delicious and easy to eat. |
| previous-4 (I) | Do you ever cook hot pot dishes by yourself? |
| previous-3 (C) | I often make and eat hot pot dishes in winter because it is a good way to get vegetables as well. |
| previous-2 (I) | Is the hot pot for one person? Do you cook for your family? |
| previous-1 (C) | I make one for my family. |
| **Human** | What do you pay attention to when you cook for your family members? |
| **Retrieval** | You mentioned that you can't eat eggplant, but are you a vegetable eater? |
| **Text Generation** | Do you like meat? |
| **Proposed** | What were your favorite ingredients for the hot pot dishes? |

| | Dialogue-2 |
|---|---|
| previous-5 (C) | For Korean food, spicy jjigae, chijimi, and bibimbap. I also like to look for good kimchi. |
| previous-4 (I) | I see. Do you like cheese Dak galbi, which has been getting popular. |
| previous-3 (C) | I like it because it is a rich taste, as I like cheese. I have even tried to recreate it at home. |
| previous-2 (I) | That's great! It is a western food, but do you like cheese fondue too? |
| previous-1 (C) | I love it! I was addicted for a while and bought a cheese fondue pot! |
| **Human** | Do you like Hamburg steak with lots of cheese? |
| **Retrieval** | Sometimes a sweet treat is okay. |
| **Text Generation** | I see. I love it too. I'll try it next time! |
| **Proposed** | What is your favorite genre? |

## 6 Conclusion

In this study, we created a dialogue model to interview the food preferences of users. Text-based dialogues between an interviewer and customer were collected, and the communicative function and semantic content of the interviewer's utterances were annotated. Using this dataset, we created models to predict the communicative function of the interviewer's utterances and generate semantic content. The outputs of these two models were then applied to template-based response generation to produce a response. In the model evaluation for generating semantic content, the proposed model outperformed the two baseline models, retrieval and generative, in the automatic evaluation using BLEU-4.

As future work, we will improve the response generation mechanism to generate a variety of expressions because the current template-based response generation may not be sufficient in its expressiveness. For example, one of the ideas would be presenting candidates such as Japanese, Chinese, and Italian when asking about preferences for a genre and asking the user to select one. It would also be useful to predict the user's preference based on the dialog history and user information and generate questions such as "Do you prefer Chinese to Italian? Thus, by using question content (e.g., genre) and related vocabulary and knowledge (Chinese and Italian as examples of genre), the question variation can be increased. Another possibility is to automatically extract or determine the response templates through machine learning, but this is a challenging task.

Further, a user study should be conducted, as it is known that automatic evaluation using BLEU does not always correlate with human evaluation (Liu et al., 2016). In the user study, users interact with the system, and then they evaluate the quality of the responses generated from the system, and judge whether the system effectively elicits information from the user.

## Acknowledgements

This work was supported by JST Moonshot R&D Grant Number JPMJMS2011 and JST AIP Trilateral AI Research (PANORAMA project, grant no. JPMJCR20G6) and JSPS KAKENHI (grant numbers JP19H01120 and JP19H04159).

## References

Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David Traum. 2012. ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 430–437, Istanbul, Turkey. European Language Resources Association (ELRA).

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fabian Galetzka, Jewgeni Rose, David Schlangen, and Jens Lehmann. 2021. Space efficient context encoding for non-task-oriented dialogue generation with graph attention transformer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7028–7041.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.

Michael Johnston, Patrick Ehlen, Frederick G Conrad, Michael F Schober, Christopher Antoun, Stefanie Fail, Andrew Hupp, Lucas Vickers, Huiying Yan, and Chan Zhang. 2013. Spoken dialog systems for automated survey interviewing. In *Proceedings of the SIGDIAL 2013 Conference*, pages 329–333.

Dan Jurafsky. 1997. Switchboard swbd-damsl shallow-discourse-function annotation coders manual. *Institute of Cognitive Science Technical Report*.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. *Advances in neural information processing systems*, 31.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. 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*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Toyomi Meguro, Yasuhiro Minami, Ryuichiro Higashinaka, and Kohji Dohsaka. 2014. Learning to control listening-oriented dialogue using partially observable markov decision processes. *ACM Transactions on Speech and Language Processing (TSLP)*, 10(4):1–20.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. 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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 583–593, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Chinnadhurai Sankar and Sujith Ravi. 2019. Deep reinforcement learning for modeling chit-chat dialog with discrete attributes. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 1–10.

Iulian Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron Courville. 2017. Multiresolution recurrent neural networks: An application to dialogue response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015a. A neural network approach to context-sensitive generation of conversational responses. In *HLT-NAACL*.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015b. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado. Association for Computational Linguistics.

Amanda Stent, Svetlana Stenchikova, and Matthew Marge. 2006. Dialog systems for surveys: The rate-a-course system. In *2006 IEEE Spoken Language Technology Workshop*, pages 210–213. IEEE.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou, and Zhonghai Wu. 2020. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5811–5820.

Heng-Da Xu, Xian-Ling Mao, Zewen Chi, Fanshu Sun, Jingjing Zhu, and Heyan Huang. 2021. Generating informative dialogue responses with keywords-guided networks. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 179–192. Springer.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2031–2043.

## A Communicative Function Set

Communicative function set

| | |
|---|---|
| **Self-Disclosure(SD-)** | Provide own information and opinions about food. |
| SD-Fact&Experience | e.g., I ate pasta yesterday. |
| SD-Preference-Positive | e.g., I like oranges. |
| SD-Preference-Negative | e.g., I don't like fish. |
| SD-Preference-Neutral | e.g., Coriander is iffy. |
| SD-Habit | e.g., I often drink coffee. |
| SD-Desire | e.g., I want to eat pizza. |
| SD-Plan | e.g., I will have sushi tonight. |
| SD-Other | |
| **Question (Q-)** | Ask questions about their food information and opinions. |
| Q-Fact&Experience | e.g., What did you eat for breakfast? |
| Q-Preference-Positive | e.g., What is your favorite dish? |
| Q-Preference-Negative | e.g., What food do you dislike? |
| Q-Preference-Neutral | e.g., Can you eat apples? |
| Q-Habit | e.g., Do you eat eggs often? |
| Q-Desire | e.g., What do you want to eat for dinner? |
| Q-Plan | e.g., What are you planning to eat for dinner? |
| Q-Other | |
| Proposal | Recommendations. e.g., Chocolate is recommended. |
| Acknowledge | Encourage the conversational partner to speak. e.g., Huh. Yes. |
| Appreciation | Express understanding. e.g., Okay. I understand. |
| Repeat | Repeat the partner's utterance. |
| Summarize&Reformulate | Paraphrasing, evaluating, and summarizing the partner utterance. |
| Exclamation | Express emotion utterance. e.g., Oh. |
| Accept&Agree&Sympathy | Expressing affirmation or agreement. |
| Partial Accept | Partially expressing affirmation or agreement. |
| Maybe | Ambiguous utterance. e.g., Maybe so. |
| Partial Reject | Partially express denial or disagreement. |
| Reject&Non-Sympathy | Express denial or disagreement. |
| Greeting | Greeting. e.g., Hello. |
| Thanks | Express thanks. e.g., Thank you. |
| Apology | Express apologies. e.g., Excuse me. |
| Filler | Utterance that fills in the pauses when stuck. e.g., Umm. Well. |
| Other | Other utterances. |

We defined the labels with reference SWBD-DAMSL (Jurafsky, 1997) and Meguro et al. (2014)'s dialogue acts.