

# Natural Language Response Generation in Mixed-Initiative Dialogs using Task Goals and Dialog Acts

Helen M. Meng, Wing Lin Yip, Oi Yan Mok and Shuk Fong Chan

Human-Computer Communications Laboratory  
 Department of Systems Engineering and Engineering Management  
 The Chinese University of Hong Kong  
 {hmmeng, wlyip, oymok, sfchan}@se.cuhk.edu.hk

## Abstract

This paper presents our approach towards natural language response generation for mixed-initiative dialogs in the CUHK Restaurants domain. Our experimental corpus consists of about 4000 customer requests and waiter responses. Every request/response utterance is annotated with its task goal (TG) and dialog act (DA). The variable pair  $\{TG, DA\}$  is used to represent the dialog state. Our approach involves a set of corpus-derived dialog state transition rules in the form of  $\{TG, DA\}_{request} \rightarrow \{TG, DA\}_{response}$ . These rules encode the communication goal(s) and initiatives of the request/response. Another set of hand-designed rules associate each response dialog state with one or more text generation templates. Upon testing, our system parses the input customer request for concept categories and from these infers the TG and DA using trained Belief Networks. Application of the dialog state transition rules and text generation templates automatically generates a (virtual) waiter response. Ten subjects were invited to interact with the system. Performance evaluation based on Grice's maxims gave a mean score of 4 on a five-point Likert scale and a task completion rate of at least 90%.

## 1. Introduction

Mixed-initiative spoken dialog systems (MI-SDS) are becoming more and more sophisticated in handling human-computer interactions that allow both parties to influence the dialog flow. Continual performance improvements in speech recognition and natural language understanding enable SDS to handle increasingly complex user inputs. In order to enable the computer to generate cooperative and coherent system outputs that tailor to the user's information needs, efforts have been devoted towards response generation (RG) [1]. Cooperative system responses are critical for the overall usability and perceived intelligence of the SDS. We have chosen to work on RG in the context of the CUHK Restaurants domain, where our prototype system simulates the interaction between a customer and a waiter. Our long-term goal is to develop a corpus-driven approach for RG. This paper begins by addressing the following research issues: (1) To identify and represent the communicative goal(s) of a response in relation to those in the user's input for a given dialog turn – Belief Networks are used to infer the task goal (TG) and dialog act (DA) of an input customer request. The variable pair  $\{TG, DA\}$  is used to present the dialog state of the request. We also use a set of corpus-derived dialog state transition rules that govern transitions from request to response dialog states. (2) To verbalize the response message with appropriate selection of semantic, syntactic and lexical structures – we have hand-designed a set of text generate templates, each of which is associated with a

response dialog state. The templates specify sentential structures that can incorporate semantic categories parsed from the user requests to generate a coherent system response.

## 2. The CUHK Restaurants Corpus

We have collected a corpus of dialogs that capture interactions between a customer and a waiter in a restaurant. Our original corpus contains 199 dialogs collected from websites and books for English learning [2]. Our current corpus contains an additional 61 dialogs collected from recruited subjects. We divide the corpus into disjoint training (220 dialogs, with 1543 customer requests and 1872 waiter responses) and test sets (40 dialogs, with 245 customer requests and 297 waiter responses). Each customer request or waiter response is segmented into individual utterances [2]. Each utterance is annotated with a task goal (TG), dialog act (DA), and concept categories. Some categories associate with the corresponding TG (denoted as  $C_{TG}$ ), while other categories associate with the DA (denoted as  $C_{DA}$ ). Table 1 shows an example dialog.

Waiter <sub>1</sub> : (numbered with dialog turn index)	"May I take your order, sir?" $C_{TG}$ : Request Order QuestMark $C_{DA}$ : Request ActionWord QuestMark $TG$ : ORDER_FOOD $DA$ : OFFER
Customer <sub>2</sub> :	"Let's see." $C_{TG}/C_{DA}$ : DeferPhrase Period $TG$ : OUT-OF-DOMAIN (OOD) $DA$ : DEFER
	"I'd like a fresh ground beef steak." $C_{TG}/C_{DA}$ : Preference Food_Item Period $TG$ : ORDER_FOOD $DA$ : PREFER
Waiter <sub>2</sub> :	"How would you like that done?" $C_{TG}$ : How Cook QuestMark $C_{DA}$ : Wh_Word Cook QuestMark $TG$ : ORDER_FOOD $DA$ : REQUEST_INFO
Customer <sub>3</sub> :	"I prefer medium." $C_{TG}/C_{DA}$ : Preference SteakStyle Period $TG$ : ORDER_FOOD $DA$ : PREFER
Waiter <sub>3</sub> :	"Anything else, sir?" $C_{TG}/C_{DA}$ : Else QuestMark $TG$ : ORDER_FOOD $DA$ : REQUEST_INFO
	"How about mixed vegetables on the side?" $C_{TG}/C_{DA}$ : Suggest Food_Item QuestMark $TG$ : ORDER_FOOD $DA$ : SUGGEST
Customer <sub>4</sub> :	"No, that's all." $C_{TG}/C_{DA}$ : NoWord Period $TG$ : ORDER_FOOD $DA$ : FEEDBACK_NEGATIVE
Waiter <sub>4</sub> :	"You're welcome." $C_{TG}/C_{DA}$ : Closing Period $TG$ : ORDER_FOOD $DA$ : CLOSE

Table 1: An example dialog in the CUHK Restaurants domain with annotated task goals (TG), dialog acts (DA), and the concept categories with the TG ( $C_{TG}$ ) and DA ( $C_{DA}$ ).

## 3. Task Goal and Dialog Act Identification

The CUHK Restaurants domain has six domain-specific TGs: ASK\_INFO, BILL, COMPLAINT, ORDER\_FOOD, RESERVATION AND SERVE. The domain also has fourteen DAs, adopted from

Verbmobil-2 [4] – BACKCHANNEL, BYE, DEFER, GREET, PREFER, FEEDBACK\_POSITIVE, FEEDBACK\_NEGATIVE, REQUEST\_ACTION, REQUEST\_COMMENT, REQUEST\_INFO, REQUEST\_SUGGEST, SUGGEST, THANK, and INFORM. Among these, INFORM is a catch-all DA we have inserted for our domain.

We have trained a suite of Belief Networks (BN) to infer the TG or DA of a given customer utterance. Details have been described in [3]. Each BN corresponds to one TG or DA. The BN receives categories parsed from the input utterance, i.e.  $C_{TG}$  and  $C_{DA}$ . Parsing involves a single set<sup>1</sup> of handwritten grammar rules and is a two-pass procedure. The second pass is simple and serves to unify some categories for DA inference. There are 110 semantic and 3 syntactic<sup>2</sup> categories in total. Example rules are shown in Table 2. The categories parsed from the current utterance, together with other categories selectively inherited [2] from the previous dialog turn(s), form the input to the BNs. Each BN has its own set of input categories automatically identified to be most indicative of the TG/DA based on the Information Gain [3] criterion. Also, the topology used for the BNs assumes that the categories are independent of one another. Based on the presence/absence of its input categories, each BN applies Bayesian inference to make a binary decision regarding the presence/absence of its corresponding TG/DA.

<b>Semantic/Syntactic Category → Terminals</b>
Bill → <i>settle my bill   bill   ...</i>
Preference → <i>prefer   let me   would like   ...</i>
Period → .
<b>Grammar rules used in the second pass</b>
Action → Bill   Reserve   Order   ...
Wh word → Where   What   Which   ...

Table 2: Example grammar rules used to parse for concept categories (C) related to task goal (TG) and dialog act (DA) inference.

We trained the BNs using the training set of the newly expanded CUHK Restaurants corpus and tested on the disjoint test set. Evaluation is based on the customer requests only. Our BN-based framework achieves an accuracy of 87.4% on TG identification, and 89.8% on DA identification.

#### 4. Cooperative Response Generation

This section describes our approach towards response generation. We begin by automatically deriving dialog state transition rules from the training data to govern transitions from requests to response. We have also hand-designed, with reference to the training data, a set of text generation templates to be associated with each dialog state transition rule. The templates can be applied to existing and inherited concept categories from the request to generate a coherent response.

##### 4.1 Corpus-derived Dialog State Transition Rules

We use both TG and DA to represent the dialog state of the request/response, and capture dialog state transition rules in the form of  $\{TG, DA\}_{request} \rightarrow \{TG, DA\}_{response}$  automatically from the training set. Since ours is a service-oriented domain, we assume that the TG of the waiter’s response always follows

that of the customer. Hence the generated responses are deemed *cooperative*.

A customer request may contain multiple utterances and hence have multiple TGs and DAs. An example is provided by the initial customer request in Table 1. For such cases, we derive only a single request dialog state  $\{TG, DA\}_{request}$  from the latest utterance as a simplification step. Referring to Table 1, the customer request “*Let’s see. I’d like a fresh ground beef steak.*” will be represented by the dialog state  $\{ORDER\_FOOD, PREFER\}_{request}$  only.

A waiter response may also have multiple utterances. While the task goals of these response utterances are consistent with that of the customer request, the dialog acts are not. Hence we obtain multiple dialog states from the response utterances. We observe from our training corpus that waiter responses have two utterances on average, and from which we can derive multiple dialog acts. Again, if we refer to Table 1 for an illustrative example, the waiter response is “*Anything else, sir? How about mixed vegetables on the side?*” The first utterance relates to the dialog act REQUEST\_INFO while the second to SUGGEST. Our scheme derives a dialog state transition rule with conjoined response states (see Rule A in Table 3).

It is conceivable that there may be alternative responses to a given customer request, just as an alternative response to the above example request may simply be “*Anything else, sir?*” Alternative responses derive alternative dialog state transition rules. Rule B in Table 3 provides an illustration. In our training corpus, the request dialog state  $\{ORDER\_FOOD, PREFER\}_{request}$  occurred 134 times, of which 10 responses map to Rule A and 50 responses map to Rule B. The remaining cases lead to other response dialog states. Those responses with high occurrences are chosen to form dialog state transition rules. Each request dialog state may associate with more than one rule. Overall our training corpus produced 104 dialog state transition rules.

##### Dialog State Transition Rule Format:

$\{TG, DA\}_{request} \rightarrow \{TG, DA\}_{response}$

##### Rule A

$\{ORDER\_FOOD, PREFER\}_{request} \rightarrow$   
 $\{ORDER\_FOOD, REQUEST\_INFO\}_{response}$   
 &  $\{ORDER\_FOOD, SUGGEST\}_{response}$

##### Rule B (Alternative)

$\{ORDER\_FOOD, PREFER\}_{request} \rightarrow$   
 $\{ORDER\_FOOD, REQUEST\_INFO\}_{response}$

Table 3: Dialog state transition rule derived from the third dialog turn from the example dialog in Table 1, followed by an alternative rule based on a similar customer request.

##### 4.2 Hand-designed Text Generation Templates

We have hand-designed 126 text generation templates with reference to the training corpus. Each response dialog state maps to one or more templates. The same template may be mapped by more than one response dialog state -- for example, the response dialog states  $\{ORDER\_FOOD, GREET\}_{response}$  and  $\{RESERVATION, GREET\}_{response}$  map to the same template GREETING. Selection among template options is conditioned upon the parsed categories from the current utterance and their corresponding values (i.e. terminals). As can be seen from Table 4, a template may include one or more verbalization options. Each option may specify concept categories (denoted by ‘#’) whose values are obtained either from the parsed customer utterance with its inherited discourse, or from terminals of the corresponding grammar rule. Table 4 also

<sup>1</sup> This is different from the setup in [2] which used separate grammars to parse for  $C_{TG}$  and  $C_{DA}$  respectively. We combined the grammars to promote sharing, avoid redundancies and ease further grammar development.

<sup>2</sup> The three syntactic categories correspond to punctuation, i.e. question mark, exclamation mark and period.

illustrates with the response dialog state  $\{ORDER\_FOOD, REQUEST\_INFO\}_{response}$  that can map to the templates ASK\_STEAK\_STYLE or ANYTHING\_ELSE. The former template is selected if the utterance contains the grammar terminal *steak* that is parsed into the category Food\_Item.

<p><b>Text Generation Templates:</b></p> <p>Template label: GREETING Template contents: <i>Hi.   Hello.</i></p> <p>Template label: ASK_STEAK_STYLE Template contents: <i>How would you like that done?</i></p> <p>Template label: ANYTHING_ELSE Template contents: <math>\langle\{request: \#Food\_Item\}\rangle</math> <i>Anything else, sir?   Is there anything else?   Is there anything else, sir?</i></p> <p>Template label: SUGGEST Template contents: <i>How about {grammar: #Food_Item}?   I would recommend {grammar: #Food_Item}?   What about {grammar: #Food_Item}?</i></p> <p>Template label: STOP (N.B. denotes that text generation stops at this point and will not continue despite the existence of further response dialog states, e.g. those appended with the ‘&amp;’ symbol as in Rule A)</p>
<p><b>Response Dialog State:</b> <math>\{ORDER\_FOOD, REQUEST\_INFO\}_{response}</math></p> <p><b>Associated Text Generate Templates:</b>  <b>Option 1:</b> ASK_STEAK_STYLE, STOP  <b>Option 2:</b> ANYTHING_ELSE</p> <p><b>Template Selection Rule:</b>  If parsed categories <math>C_{TG}</math> contains (Food_Item <math>\rightarrow</math> <i>steak</i>) then select template ASK_STEAK_STYLE otherwise select template ANYTHING_ELSE</p>
<p><b>Response Dialog State:</b> <math>\{ORDER\_FOOD, SUGGEST\}_{response}</math></p> <p><b>Associated Text Generation Templates:</b> SUGGEST</p>

Table 4: Examples of hand-designed text generation templates. Each response dialog state is mapped to one or more templates. The template may incorporate concept categories (denoted by ‘#’) whose values are obtained either from the parsed customer request (denoted by  $\{request: \#category\}$ ) or from terminals of the corresponding grammar rule (denoted by  $\{grammar: \#category\}$ ). The brackets  $\langle \rangle$  indicate that the category is optional in text generation (i.e.  $\langle\{request: \#Food\_Item\}\rangle$  shows the waiter may confirm the food ordered by the customer.)

### 4.3 Rule Application and Template Selection

Upon testing with an incoming customer request, the suite of BNs will infer the corresponding TG and DA [3] to produce the request dialog state. Our system then invokes the appropriate dialog state transition rule to produce the response dialog state. In the event that alternative rules are available, one will be chosen at random for invocation (see Table 3). Rule invocation produces one or more response dialog states, which are mapped respectively to their associated text generation templates. Application of a template involves a randomized selection among the verbalization options and incorporation of the appropriate semantic concept categories either extracted from the parsed customer request or from the terminals of the corresponding grammar rule (see Table 4). Should multiple options in grammar terminals be available, an option will be selected at random. Table 5 illustrates this process with examples of generated responses. The grammar terminal “*mixed vegetables on the side*” is selected at random

from the category Food\_Item. This category appears in the template SUGGEST which is used for text generation (see Table 4).

<p><b>Customer Request:</b>  <i>“I wish to order an omelette with lobster meat.”</i>  Request dialog state: <math>\{ORDER\_FOOD, PREFER\}_{request}</math></p>
<p><b>Response Generation (Case 1)</b>  <b>Response dialog states</b> produced by invocation of Rule A (see Table 3) are: <math>\{ORDER\_FOOD, REQUEST\_INFO\}_{response}</math> &amp; <math>\{ORDER\_FOOD, SUGGEST\}_{response}</math>  <b>Templates Applied</b> (see Table 4): ANYTHING_ELSE &amp; SUGGEST  <b>Generated response:</b> <i>“Omelette with lobster meat. Anything else, sir? How about mixed vegetables on the side?”</i></p>
<p><b>Response Generation (Case 2)</b>  <b>Response dialog states</b> produced by invocation of Rule B (see Table 3) are: <math>\{ORDER\_FOOD, REQUEST\_INFO\}_{response}</math>  <b>Template Applied</b> (see Table 4): ANYTHING_ELSE  <b>Generated response:</b> <i>“Is there anything else, sir?”</i></p>

Table 5: Illustration of the response generation procedure. The table presents two possible responses that may be generated.

## 5. Evaluation

We have incorporated the cooperative response generation mechanism in an initial prototype of the interactive CUHK Restaurants system. The system accepts typed natural language queries as input. We recruited 10 subjects and asked them to interact with the system as if they were talking to a waiter. Each subject is given three tasks that they need to complete: (i) reserve a table; (ii) order a meal; and (iii) ask for the bill. All interactions are automatically logged by the system. An example dialog is shown in Table 6. The average number of dialog turns for each task are shown in Table 7. The number of dialog turns for RESERVATION is relatively higher because more information is needed. We evaluate the dialogs in terms of the task completion rate, Grice’s maxims [5] as well as overall user satisfaction. Details of the evaluation process are provided in the following subsections.

Customer <sub>1</sub>	<i>“May I have the menu, please?”</i>
Waiter <sub>1</sub>	<i>“Certainly, sir. Here is the menu. Have you decided on something?”</i>
Customer <sub>2</sub>	<i>“Green Salad with chicken, and a lobster for my friend, please.”</i>
Waiter <sub>2</sub>	<i>“Is there anything else?”</i>
Customer <sub>3</sub>	<i>“No, thanks.”</i>
Waiter <sub>3</sub>	<i>“You have ordered green salad with chicken and a lobster. Your order should arrive within 15 minutes.”</i>

Table 6: An example dialog extracted from the system log during the evaluation process.

Tasks	Reservation	Order_	Bill
Average no. of dialog turns	6.6	4.6	2.6

Table 7: Average number of dialog turns across the 10 evaluation dialogs for each of the three tasks.

### 5.1 Task Completion Rate

All the evaluation dialogs logged by the system have been checked for task completion. A task is considered complete if the appropriate confirmation message is present in the dialog. For the reservation task, we search for the system confirmation – e.g., “*You have reserved a table for a party of \_\_ by the window,<sup>1</sup> at \_\_ am/pm tomorrow.<sup>1</sup>”* For the

<sup>1</sup> Optional, depends on whether the customer requested a particular table location.

ordering task, we search for the system confirmation – e.g., “*You have ordered \_\_.*” For the billing task, we search for the system confirmation, “*Your bill comes to \$\_\_.*” A task is considered complete as long as the appropriate confirmation message exists, even if there are incoherent dialog turns involved. This and the simplicity of our evaluation tasks have led to high task completion rates across the evaluation dialogs (See Table 8).

Tasks	Reservation	Order	Bill
Task Completion Rate	100% (10/10)	90% (9/10)	100% (10/10)

Table 8: Task completion rates across the 10 evaluation dialogs for each of the tasks – reservation, ordering food and requesting the bill.

## 5.2 Grice’s Maxims and Perceived User Satisfaction

We attempted to evaluate response generation in terms of Grice’s Maxims as well as overall user satisfaction. Each subject was asked to fill out a questionnaire that contains three sets of questions, one for each task (i.e. reservation, ordering food and requesting the bill). The set of questions is identical across the tasks and relate to Grice’s Maxims as well as overall user satisfaction. The questions are listed as follows:

- (i) **Maxim of Quality**, i.e. system responses should be true with adequate evidence – “*Do you think that the answers of the virtual waiter are accurate and true?*”
- (ii) **Maxim of Quantity**, i.e. system should give sufficient information – “*Do you think that the answers of the virtual waiter are informative?*”
- (iii) **Maxim of Relevance**, i.e. system responses should be relevant to the ongoing conversation – “*Do you think that the answers of the virtual waiter are relevant to the conversation?*”
- (iv) **Maxim of Manner**, i.e. system responses should be brief and clear, with no obscurity or ambiguity – “*Do you think that the answers of the virtual waiter are clear?*”
- (v) **Overall User Satisfaction** – “*To what extent are you satisfied with the overall performance of the system in responding to your questions?*”

The subjects were asked to respond to these questions on a five-point Likert scale: very poor / poor / average / good / very good. Table 9 shows the average scores and standard deviations (in brackets). A *t*-test shows that our results are significantly better than average (Likert score 3) at  $\alpha=0.05$ .

	Reservation	Order	Bill
M. Quality	4.0 (0.8)	4.1 (0.7)	4.0 (1.1)
M. Quantity	4.2 (1.1)	4.1 (0.9)	4.0 (0.8)
M. Relevance	3.9 (0.6)	3.8 (0.9)	4.2 (1.0)
M. Manner	4.4 (0.7)	4.1 (0.7)	4.1 (0.7)
Satisfaction	4.1 (0.7)	4.0 (0.7)	4.0 (0.8)

Table 9: Average scores and standard deviations (in brackets) from a five-point Likert scale obtained from evaluation of our response generation mechanism in terms of Grice’s Maxims and overall user satisfaction.

## 5.3 Error Analysis

We have also analyzed individual dialog turns from these evaluation dialogs. In some cases the system did not produce a coherent response to the user’s question due to errors in parsing for concept categories that led to incorrectly inferred

tasks goals or dialog acts. Table 10 presents an illustration – in the second dialog turn, the customer said, “*It is more than enough.*” This is an implicit way of saying ‘no’ to further recommendations, and the dialog act should be FEEDBACK\_NEGATIVE. However, utterance was not parsed properly and our system defaults to the dialog act INFORM. This causes the system to proceed to offer further entrée recommendations. There are about 9 out of 138 (6.5%) incoherent response turns in the evaluation dialogs.

Customer <sub>1</sub>	“ <i>I would like a drink. One cup of hot Chinese tea please. Main course would be smoked turkey with green salad.</i> ”
Waiter <sub>1</sub>	“ <i>Anything else, sir?</i> ”
Customer <sub>2</sub>	“ <b>It is more than enough</b> TG: ORDER_FOOD DA: INFORM (×)
Waiter <sub>2</sub>	“ <i>Anything else, sir? What about mixed vegetables on the side?</i> ”

Table 10: An example extracted from the evaluation dialogs to illustrate an incoherent dialog turn. It is due to a parse error leading to an incorrectly inferred dialog act.

## 6. Conclusions and Future Work

This paper reports on our initial attempt to develop a natural language generation mechanism for a prototype system in the CUHK Restaurants domain. The system behaves like a virtual waiter and can interact with the customer (i.e. user) in natural language. For a given customer request, we parse for relevant concept categories and infer the underlying task goal (TG) and dialog act (DA) of the utterance by means of Belief Networks. The variable  $\{TG, DA\}$  is used to represent the dialog state of the customer request. A set of corpus-derived dialog state transition rules governs the transition from a request dialog state to a response dialog state. The latter is also represented in terms of  $\{TG, DA\}$ . Each response state is associated with a text generation template. The templates are hand-designed with reference to our training corpus. A template is applied to the semantic frame of customer request to produce a coherent response. Evaluation based on thirty interactive dialogs from ten subjects showed a 90% task completion rate as well as a mean score of 4 on a Likert scale that relates to Grice’s Maxims and overall user satisfaction. Future work will be devoted towards the use of semi-automatic techniques to achieve extensibility and scalability.

## 7. Acknowledgments

The work described in this paper was partially supported by grants from the Research Grants Council of the Hong Kong SAR Government, China (Project Nos. CUHK4326/02E and CUHK4177/00E).

## 8. References

- [1] Chu-Carroll, J., *Response generation in collaborative dialogue negotiation*, Proc. of ACL, 1995.
- [2] Chan, S. F and H. Meng, *Interdependencies among Dialog Acts, Task Goals and Discourse Inheritance in Mixed-Initiative Dialog*, Proc. of HLT, 2002.
- [3] Meng, H., W. Lam, C. Wai, *To Believe is to Understand*, Proc. of Eurospeech, 1999.
- [4] Alexandersson, J. et al, *Dialog Acts in VERBMOBIL-2 Second Edition*, Verbmobil Report 226, Universitat Hamburg, DFKI Saarbrücken, Universitat Erlangen, TU Berlin, July 1998.
- [5] R. Frederking, Grice’s maxims: *do the right thing*, in Frederking, R.E. (1996).

<sup>1</sup> Depends on the customer’s requested date.