A New Method for Understanding Sequences of Utterances by Multiple Speakers

Noboru Miyazaki, Jun-ichi Hirasawa, Mikio Nakano and Kiyoaki Aikawa
nmiya@atom.brl.ntt.co.jp  http://www.kecl.ntt.co.jp/
Communication Science Laboratories
Nippon Telegraph and Telephone Corporation (NTT)

Abstract

This paper presents a new method for understanding utterances in spoken dialogue. The method is characterized by an understanding rule that accepts an utterance sequence in which each utterance has its own speaker identifier. This new idea enables the speech understanding system to understand a dialogue composed of sequences of short phrases each possibly uttered by different speakers. The method is also applied to a spoken dialogue system by regarding the dialogue system itself as one of the speakers. We give a formal description of the new dialogue understanding method. Sample behavior of a spoken dialogue system with our new method is also presented followed by an evaluation in terms of the amount of dialogue description. The evaluation shows our method reduces the amount of understanding state description.

1 INTRODUCTION

The main topic of this paper is spoken language dialogue understanding. Conventional spoken dialogue systems are designed to understand a single speaker’s utterances that are long enough so that their meanings are clear after language processing [1, 5, 2]. However, in human-human dialogue, speakers change frequently. Although such changes usually occur at sentence boundaries, sometimes they occur at the boundary of phrases within a sentence. It is sometimes very difficult to determine the meaning of one speaker’s utterance by simply accounting for his/her speech fragments. This is because the meaning of each utterance may depend on the preceding utterances of the other speaker. In these cases, understanding of one speaker’s utterance must be performed taking into account with the other speaker’s utterances.

To cope with this problem, we propose a new method of dialogue understanding. In this method, a dialogue understanding system accepts a sequence of short phrases as inputs, where each phrase in the sequence is possibly uttered by different speakers including the dialogue system itself. A set of dialogue understanding rules that refer to a sequence of utterances with speaker identifiers are incorporated to process those inputs. We describe our dialogue understanding method in section 2, a sample dialogue system in section 3. Section 4 shows sample behavior and some evaluation of the system. Section 5 is a conclusion.

2 DIALOGUE UNDERSTANDING

2.1 Formalization

Let \( z \in Z \) denote a speaker in a dialogue. Also, we define \( w \in W \) as an utterance unit such as a phrase or word, where \( W \) denotes all meanings that could be uttered in a certain dialogue domain. With \( z \) and \( w \), we define an utterance \( x \) as

\[
x = (w, z),
\]

which we regard as an input to the dialogue understanding system. We represent an understanding state of the system as \( s \in S \), where \( S \) denotes all possible internal states in certain dialogue domain.

The system holds an utterance sequence \( X \) and a state sequence \( S \) to process a dialogue. At time \( t \), each sequence is defined by

\[
X_t = \{x_1, x_2, \ldots, x_t\}
\]

\[
S_t = \{s_0, s_1, s_2, \ldots, s_t\}
\]

where \( s_0 \) denotes the initial understanding state of the system. Time variable \( t \) starts from 1 and is incremented every time the system accepts an utterance \( x \). The initial sequences are:

\[
S_0 = \{s_0\}, \quad X_0 = \{\phi\}
\]

Every time the system accepts \( x_t \), it updates \( X \) and \( S \) with the procedure in Fig. 1. In the procedure, \( \mathcal{R} \) denotes a set of dialogue understanding rules, which are described in section 2.2. Function \( f \) extracts a set of rules \( \Omega_t \in \mathcal{R} \) whose elements can be applied to \( X_t \) and \( S_{t-1} \). Function \( g \) selects an appropriate rule \( r_t \in \Omega_t \). The rule \( r_t \) that is applied to \( X_t \) and \( S_{t-1} \) generates a new understanding state \( s_t \).

2.2 Dialogue understanding rule

A dialogue understanding rule \( r \) is described as follows:

\[
r = (p, c, U)
\]

\[
p, c \in S
\]

Here, \( p \) means a premise to apply rule \( r \), and \( c \) denotes a result after applying rule \( r \). An utterance sequence pattern
1. \( \mathbf{X}_t = \{\mathbf{X}_{t-1}, x_t\} \)

2. \( \Omega_t = f(\mathbf{R}, \mathbf{X}_t, \mathbf{S}_{t-1}) \)

3. If \( \Omega_t \neq \phi \) then
   \[ r_t = g(\Omega_t) \]
   \[ s_t = r_t(\mathbf{X}_t, \mathbf{S}_{t-1}) \]
   else
   \[ s_t = s_{t-1} \]

4. \( \mathbf{S}_t = \{\mathbf{S}_{t-1}, s_t\} \)

5. \( t = t + 1 \)

Figure 1: procedure to update understanding state

\( \mathbf{U} \) has an arbitrary length that depends on each rule, and has the form of utterance sequence \( \mathbf{S} \).

\[
\mathbf{U} = (u_1, u_2, u_3, \ldots, u_m)
\]
\[
u_i = (v_i, y_i)
\]
\[v_i \in \mathcal{W}, \ y_i \in \mathcal{Z}\]

At each time \( t \), the function \( f \) selects every \( r \) that satisfies conditions (7) and (8) with the context \( \mathbf{X}_t \) and \( \mathbf{S}_{t-1} \).

\[
P = s_{t-m}
\]
\[
\mathbf{U} = (x_{t-m+1}, x_{t-m+2}, \ldots, x_t)
\]

If \( \Omega_t \) contains more than one rule, function \( g \) selects one rule \( r_t \) which will be applied to \( \mathbf{X}_t \) and \( \mathbf{S}_{t-1} \). \( \epsilon_t \) that \( r_t \) addresses becomes the new understanding state \( s_t \). Fig. 2 shows the way the system applies a rule. Here, dashed utterances and dashed states have no effect on the evaluation of (7) and (8).

### 2.3 Characteristics

This new method has three features. (1) It receives a speaker identifier with each utterance. (2) It accounts for the speaker of each utterance to understand the input sequence. (3) It imposes no constraint on the length of the utterance sequence. These features result in two advantages. One is that it provides a general structure of dialogue understanding rules that can handle multi party dialogues. The other is that it allows us to apply dialogue understanding rules to a sequence with more than two concepts or utterances, which cannot be modeled by adjacency pairs [4] or dialogue state [5]. For example, a typical interaction at the opening or closing of a dialogue can contain more than two utterances.

### 3 DIALOGUE SYSTEM

#### 3.1 system definition

Although the dialogue understanding method described above can handle multi party dialogue by generalized speaker identifier \( z \), we can also adopt it to a dialogue between a user and a dialogue system under following constraint.

\[
\mathcal{Z} = \{\text{User, System}\}
\]

Fig. 3 shows an architecture of a dialogue system using our dialogue understanding method. In this architecture, most of the domain dependent knowledge to understand dialogue is given in the dialogue understanding rules, whereas most of the dialogue understanding module works independently of the domain knowledge\(^1\). The response generation module receives the current understanding state from the dialogue understanding module and outputs one or several meanings sequentially in a certain semantic representation form. The meanings are not only input to the language generation module and then to the TTS module, but also to the dialogue understanding module after they are gathered with user utterances. We do not discuss about the ASR, parsing, language generation and the TTS because these functions are out of the scope of this paper.

#### 3.2 Task settings

We have implemented a simple “weather information service” task. The task is to receive three kinds of information:

1. Function \( g \) may be domain dependent.
understanding state is as follows. We use a feature set to represent an utterance meaning. Three of the attributes (place, day, type) are used to represent each kind of information, while the rest of the attributes (pf, df, tf) are used as confirmation flags. The flags take either true or false values. The formal description of a dialogue understanding state is as follows.

\[ s = (\text{place}, \text{pf}, \text{day}, \text{df}, \text{type}, \text{tf}) \] (10)
\[ s_0 = (\text{null}, \text{F}, \text{null}, \text{F}, \text{null}, \text{F}) \] (11)

We use a feature set to represent an utterance meaning \( w \). We incorporate three features: \( w_{\text{cat}} \in W_{\text{cat}} \) as a word category, \( w_{\text{mea}} \in W_{\text{mea}} \) as a word meaning, \( w_{\text{mod}} \in W_{\text{mod}} \) as a modality.

\[ w = (w_{\text{cat}}, w_{\text{mea}}, w_{\text{mod}}) \in W \] (12)

Table 1 shows some of the feature values.

| \( W_{\text{cat}} \) | \( \text{place}, \text{day}, \text{type}, \text{discourse} \) |
| \( W_{\text{mea}} \) | \( \text{Tokyo, Osaka, ...}, \text{today}, \text{Monday},... \) |
| \( W_{\text{mod}} \) | \( \text{assert, question, confirm, ...} \) |

The formal description of a dialogue understanding state is as follows.

\[ s = (\text{place}, \text{pf}, \text{day}, \text{df}, \text{type}, \text{tf}) \] (10)
\[ s_0 = (\text{null}, \text{F}, \text{null}, \text{F}, \text{null}, \text{F}) \] (11)

We use a feature set to represent an utterance meaning \( w \). We incorporate three features: \( w_{\text{cat}} \in W_{\text{cat}} \) as a word category, \( w_{\text{mea}} \in W_{\text{mea}} \) as a word meaning, \( w_{\text{mod}} \in W_{\text{mod}} \) as a modality.

\[ w = (w_{\text{cat}}, w_{\text{mea}}, w_{\text{mod}}) \in W \] (12)

Table 1 shows some of the feature values.

### 3.3 Rules

Fig. 4 shows several dialogue understanding rules. Here, a symbol starting with '*' is a variable that takes the same value in the rule, '*' only is a wildcard, and has no effect on the evaluations of (7) and (8). The evaluations of (7) and (8) were performed by a matching algorithm. Since we use a feature set as the meaning of a word, an utterance \( u_t \in U \) has four elements that consists of three features and a speaker identifier. For example, rule 4 makes the value of the place attribute \( \text{null} \) and the \( \text{pf} \) attribute \( \text{F} \) if any of the speakers uttered place \( \ast x \) under the condition that the value of the attribute place is \( \ast x \) and if an utterance from User with modality “deny” has followed. The total number of dialogue understanding rules for this system was 38.

### 3.4 Other Modules

We set the function \( g \) to select a rule with the longest utterance pattern \( U \) among \( \Omega_t \). This comes from a heuristic that the longer the contexts with which a system is concerned, the more accurate the understanding results become. The utterance generation module consists of several primitive rules. It consults with the current state \( s_t \) and generates a question or confirmation if any of the attributes place, date, type has a value null or any of the attributes pf, df, tf has a value F.

Such as weather, rainfall probability, temperature, etc.

---

**Table 1: Word features and their values**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_{\text{cat}} )</td>
<td>( \text{place}, \text{day}, \text{type}, \text{discourse} )</td>
</tr>
<tr>
<td>( W_{\text{mea}} )</td>
<td>( \text{Tokyo, Osaka, ...}, \text{today}, \text{Monday},... )</td>
</tr>
<tr>
<td>( W_{\text{mod}} )</td>
<td>( \text{assert, question, confirm, ...} )</td>
</tr>
</tbody>
</table>
4 EXAMPLE AND EVALUATION

4.1 Example dialogue

Fig. 5 shows an example dialogue between a user and a system in Japanese. In each turn, Japanese utterances are in the first line and the dialogue understanding state is in the third line. The slash denotes an utterance unit boundary with roughly phrase level granularity. The italic part in $S_t$ means that the response generation module outputs the meaning, but it was not actually output because of the barge-in $U_t$. In this example, $U_t$ means the understanding of place = “tokyo” is incorrect. If we regard only user utterances as system inputs, we have to prepare local understanding states that denote meanings the system has uttered step by step to understand $U_t$ correctly.

In our method, on the other hand, rule 4 is applied at $t-1$. Both the system utterance “tokyono” and $U_t$ are taken into account in rule 4 so that $U_t$ is correctly interpreted without additional understanding states.

Fig. 6 shows a rule application where the rule has longer utterance sequence pattern. At time $t-1$, rule 6 interprets utterance $U_t$ as negating the entire confirmation the system has just uttered in $S_t$. However, at time $t$, rule 7 interprets the sequence $S_tU_tU_{t+1}$ to negate only the place category. The interpretation of rule 6 at $t-1$ doesn’t effect the condition of rule 7 at $t$.

4.2 Evaluation

Here, we compare dialogue systems with a conventional understanding method and with ours. A conventional method means it regards only user utterances as system input, but has the same ability to handle dialogue that can be processed by our system. We counted the number of states, the number of state transition rules, and computed mean rule length. We differentiate states into two types. One is to describe domain dependent information, and the other is to describe dialogue progress. For example, a value of attribute place is domain dependent information whereas, a value of attribute pf states dialogue progress. Table 2 shows the conventional method requires that many local contexts be described. However, our method requires a sequence of utterances in each dialogue understanding rule as a domain dependent data. A study of the trade-off between rule length and the number of states remains as a future work.

<table>
<thead>
<tr>
<th></th>
<th>proposed</th>
<th>conventional</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain states</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>dialogue progress states</td>
<td>8</td>
<td>28</td>
</tr>
<tr>
<td>rule number</td>
<td>38</td>
<td>53</td>
</tr>
<tr>
<td>mean rule length</td>
<td>2.9</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 6: example dialogue2

5 CONCLUSION

We proposed a new dialogue understanding method in order to handle spoken language dialogue that consists of short utterances. In this method, we assume a sequence of utterances, each possibly uttered by different speakers, as a system input. This framework provides a general structure of dialogue understanding rules that can handle multi-party dialogues. We used our method in a spoken dialogue system under a simple constraint on a set of speakers, and showed the behavior of the system. A comparison between a conventional dialogue understanding method and ours showed that our method needs less system description of dialogue progress.

ACKNOWLEDGEMENT

We thank Dr. Norihiro HAGITA, the executive manager of the Media Information Laboratory for his encouragements and comments. We also thank the members of the Dialogue Understanding Research Group for valuable discussions.

References