

# Cooperative Dialogue Planning with User and Situation Models via Example-based Training

Ian R. Lane, Shinichi Ueno, Tatsuya Kawahara

School of Informatics, Kyoto University

Yoshida-Hommachi, Sakyo-ku, Kyoto 606-8501, Japan

e-mail: {ian,kawahara}@ar.media.kyoto-u.ac.jp

**Keywords:** Dialogue Planning, User Modeling, Machine Learning, Spoken Dialogue System

## Abstract

To provide a high level of usability, spoken dialogue systems must generate cooperative responses for a wide variety of users and situations. We introduce a dialogue planning scheme which incorporates user and situation models, making such dialogue adaptation possible. Manually developing a set of dialogue rules to accommodate all possible model combinations is very difficult and obstructs system portability. To overcome this problem, we propose a novel example-based training scheme, where example dialogues from a role-playing simulation are used to train the dialogue planner via machine learning. The proposed scheme is evaluated on the Kyoto city voice portal, a multi-domain spoken dialogue system. Subjects participated in a role-playing simulation where they selected appropriate system responses at each dialogue turn based on given scenario information. Experimental results show that the system successfully trains the dialogue planner and provides reasonable system performance.

## 1 Introduction

The continual improvement of speech recognition and mobile communication technologies has enabled the development of interactive voice response (IVR) systems that allow users to obtain a variety of information via mobile phone based voice interfaces. However, such systems are typically difficult to operate for non-experts, and do not provide cooperative dialogue. Whether a system is cooperative to a user depends on user characteristics, such as whether the user is a novice, or in a hurry, and other external factors including time of day. For a spoken dialogue system to interact cooperatively with a user, such information must be considered during dialogue planning and response generation.

Previous research includes several methods to adapt dialogue strategies based on various cues [1, 2, 3, 4, 5]. Factors used for adaptation include, user knowledge level in the target domain [6, 7] and skill level using the system [8]. External information such as time of day and user location was incorporated in a mobile navigation system in [9]. These studies, however, typically focus on only single factors and modeling is generally task dependent. In order to generate truly cooperative responses, multiple factors must be considered simultaneously during dialogue planning.

In this paper, we present a comprehensive modeling scheme to generate user and situation-adapted responses for spoken dialogue systems. As domain independent user characteristics, skill level to the system, degree of hastiness, and dialogue goal clarity are used and detected in real time. External factors including time of day, location of the place of interest, and external events that may affect the task are also taken into account. These models provide non-linguistic information that enables detailed user and situation specific dialogue plans to be generated.

The main problem in implementing a dialogue management scheme incorporating the above models is plan complexity. Manually generating an optimal set of dialogue rules to account for all possible model combinations would be very difficult, and there is no guarantee that these rules would generate optimal dialogue flows. To overcome this problem we introduce a machine learning approach to dialogue planning.

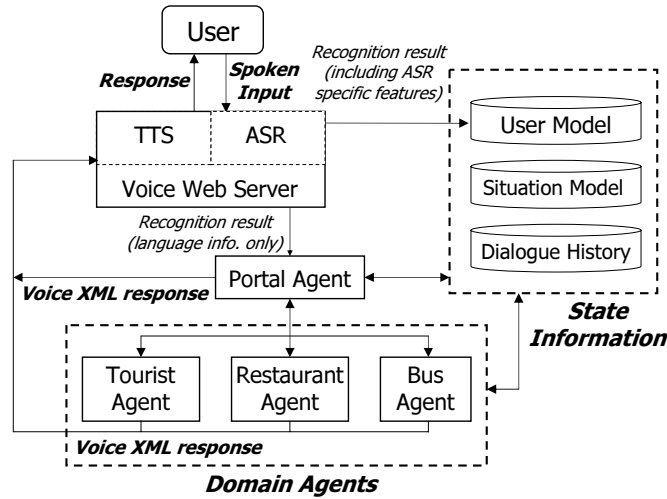


Figure 1: Kyoto City Voice Portal System Architecture

Training is performed independently for each user with data collected via role-playing dialogues, enabling a user adapted system to be realized.

In this paper, first, we introduce the Kyoto city voice portal system, which is used as a research platform for this work. Next, we describe the incorporation of user and situation models into this framework, and describe the dialogue planning scheme and machine learning algorithm applied during training. Finally, we evaluate the performance of the proposed scheme in its ability to adapt to individual users.

## 2 Kyoto City Voice Portal System

To investigate the proposed dialogue planning scheme, we have developed the Kyoto city voice portal system, a multi-domain spoken dialogue system, which provides a spoken interface to three inter-related domains;

**Tourist domain:** Information on tourist spots within Kyoto city, including operating hours, entrance fees, access methods, as well as information on festivals and special events that take place within the city.

**Restaurant domain:** Information on restaurants within Kyoto city. The system allows users to search for restaurants by food category, area, and budget.

**Bus domain:** Bus route and time-table information including real-time bus location. The system enables users to determine the correct bus to take between a given location and destination, and also provides information on how close the approaching bus is to the specified bus-stop.

The three domains are inter-related, providing bus access information to restaurants, tourist spots and other landmarks and enabling users to search for restaurants near tourist spots or landmarks.

### 2.1 System Architecture

An overview of the system is shown in Figure 1. VoiceXML scripts are generated dynamically by back-end dialogue agents based on the users' response and relevant dialogue state information. TTS and ASR engines are driven by the given VoiceXML script.

---

<b>Sys 1:</b>	<b>[Portal Agent]</b> Welcome to the Kyoto city voice portal.
<b>Sys 2:</b>	<b>[Tourist Agent]</b> For information on a tourist spot or event, simply ask a question. For example “How do I get to Kiyomizu temple?”
<b>User 1:</b>	<i>I would like to go to Kiyomizu temple.</i>
<b>Sys 3:</b>	From Shijo station, take the city bus to Kiyomizu-street or Gojyo-zaka bus-stops, it is a 10 minute walk from either stop.
<b>Sys 4:</b>	<b>[Portal-agent: Switch to bus domain]</b> Would you like bus information for Kiyomizu temple?
<b>User 2:</b>	<i>Yes</i>
<b>Sys 5:</b>	<b>[Bus-agent]</b> What is your current location?
<b>User 3:</b>	<i>Kyoto station</i>
<b>Sys 6:</b>	You require bus information from Kyoto station to Kiyomizu temple, is this correct?
<b>User 4:</b>	<i>Yes</i>
<b>Sys 7:</b>	Take bus number 206 bound for Kita-oji bus terminal. A bus has departed four bus stops away and will be arriving shortly.
<b>Sys 8:</b>	<b>[Portal-agent: Switch to restaurant domain]</b> Would you like to know about restaurants near Kiyomizu temple?
<b>User 5:</b>	<i>Yes</i>
<b>Sys 9:</b>	<b>[Bus-agent]</b> What style restaurant would you like; Japanese, Western, or Chinese.
<b>User 6:</b>	<i>Japanese please.</i>
<b>Sys 10:</b>	There are 30 matches in that area. How much do you intend to spend?
<b>User 7:</b>	<i>Around 2000 yen.</i>
<b>Sys 11:</b>	Ten restaurants match your query.
	...

---

Figure 2: Kyoto City Voice Portal System Dialogue Example

The system contains two types of dialogue agents, a portal agent, and multiple domain agents; tourist, restaurant and bus. The portal agent controls the overall dialogue flow and regulates switching between domains, selecting the appropriate domain-agent for each user query. The portal agent also enables information sharing between domain agents. Dialogue agents perform dialogue management within a single domain. These agents generate responses to user queries based on the current state information and information from back-end databases or Web based information sources. State information consists of information relevant to the current dialogue state, including dialogue history, and user and situation model attributes. This information is updated at every dialogue turn.

An example of a dialogue between a user and the system is shown in Figure 2. The system is targeted at users who require tourist information within Kyoto city, and thus it is assumed that the dialogue will begin in the “tourist” domain. Users can switch between domains multiple times within a single session enabling information from multiple domains to be gathered quickly and efficiently. The transitions between domains and response generation is handled during dialogue planning and is dependent not only on dialogue history but also on user and situation model attributes. These models are described in detail in the following section.

Table 1: User and Situation Model Attributes

Model	Attribute	Attribute Value	
		0	1
User model	Skill level	High/Unknown	Low
	Goal clarity	High/Unknown	Low
	Hastiness	Low/Unknown	High
Situation model	Meal-time	No	Yes
	Operating hours	Yes	No
	Accessibility	Convenient	Not
	Events	Yes	No

### 3 User and Situation Modeling

To generate cooperative responses, we investigate the application of two information models, the first, a user model, which provides information on user characteristics, and the second, a situation model, which provides information on external factors likely to affect dialogue. The attributes for each model are shown in Table 1. Attributes are given binary values, where a “1” indicates that the dialogue strategy is likely to differ from the baseline approach.

Both the user and situation models are updated every dialogue turn. Decision tree classifiers are applied to estimate each attribute of the user model, while attributes in the situation model are obtained directly from external databases. Each model attribute is described in detail in the following sub-sections.

#### 3.1 User Modeling

We have previously shown in [10] that adapting dialogue strategy based on user characteristics improves usability, especially for novice users. We apply a similar approach here. The user model consists of the following three attributes:

**Skill level to the system:** As there is a vast difference in skill levels between users, applying a fixed dialogue strategy for all users would be detrimental. Depending on the level of experience, the system should alternate between system-initiative and user-initiative dialogue strategies.

**Degree of hastiness:** Users are likely to access the system requiring information for a bus that will soon depart, in such cases an urgent response is required. In this situation the dialogue strategy should change, to minimize the amount of input and output required.

**Goal clarity:** Users accessing the system can typically be classified as one of two types, those that have a clear query goal, for example, “I want to visit Kiyomizu Temple” and those that require information but lack a definite goal, for example, “Where is a interesting place to go?”. For the second type of users the system should increase the amount of relevant information presented.

#### 3.2 User Classification

The three attributes of the user model are estimated in real-time for each input utterance. Classification is based on decision trees as described in [10]. Decision trees are constructed using the C5.0 training algorithm [11] with data collected from an earlier version of the system. Close to 30 features are used during classification including not only semantic information contained in the utterance but also information specific to spoken dialogue, such as the silence duration prior to the utterance and the presence of barge-in.

Table 2: Tourist Agent Utterance-plans

<b>Utterance-Plan</b>	<b>Single word guidance</b>
<b>Prerequisite Condition</b>	Tourist spot yet to be filled
<b>Evaluation Function</b>	Skill level: high → Evaluation score: high Hastiness: high → Evaluation score: high
<b>Action</b>	“Please state the name of a tourist spot”
<b>Utterance-Plan</b>	<b>Give example response</b>
<b>Prerequisite Condition</b>	Tourist spot yet to be filled
<b>Evaluation Function</b>	Skill level: low → Evaluation score: high Hastiness: high → Evaluation score: low
<b>Action</b>	“Please state the name of a tourist spot, for example “Kiyomizu temple” or “The golden pavilion”

### 3.3 Situation Modeling

In addition to user characteristics, a number of external factors likely to affect dialogue strategy are investigated. These factors were selected specifically for the Kyoto city voice portal task, but are generic to most domains. The situation model consists of four attributes relating to current time, location of the place of interest, and major events currently happening in the city. This information is gained directly from external databases.

**Meal-time:** If it is currently a meal time perhaps restaurant information should be suggested.

**Operating-hours:** If the place of interest is not currently open the user should be informed of this immediately.

**Accessibility:** If the place of interest is not easily accessible the user should be given detailed directions, and the bus schedule should be checked.

**Event:** If there is a festival or major event currently taking place within Kyoto, this knowledge may alter the users’ original plans and thus the user should be informed.

## 4 Example-based Training for Dialogue Planning

Introducing the above models provides additional meta-information enabling the system to generate more detailed and cooperative responses. However, manually generating an adequate set of dialogue rules to handle all model combinations and dialogue strategies is extremely difficult and there is no guarantee that such rules would generate cooperative dialogue from the users’ point of view. Therefore, we introduce a machine learning approach to dialogue planning where the planning scheme is trained on dialogue examples from users. First, a set of dialogue examples is collected through a role-play simulation. The collected data is then used to train the planning scheme using a machine learning algorithm.

In the proposed framework, dialogue planning is based on a 2-layer hierarchical structure, the top layer consists of *domain-plans*, which determines the sub-task of the dialogue, and the lower layer consists of *utterance-plans*, which determine the actual system response. During system development, a set of dialogue and *utterance-plans* are manually created, each plan consists of the following elements:

---

<b>Sys 1:</b>	Welcome to the Kyoto city voice portal. For information on a tourist spot or event, simply ask a question. For example “How do I get to Kiyomizu temple?”
<b>User 1:</b>	<i>I would like to go to Kiyomizu temple.</i>
<b>Sys 2:</b>	From Shijo station, take the city bus to Kiyomizu-street or Gojyo-zaka bus-stops, it is a 10 minute walk from either stop. <ul style="list-style-type: none"> <li>• <b>[Tourist-agent: Suggest other information]</b> Do you have any other questions, for example, entrance fees, operating hours, or access methods?</li> <li>• <b>[Portal-agent: Switch to bus domain]</b> Would you like bus information for Kiyomizu temple?</li> <li>• <b>[Portal-agent: Switch to restaurant domain]</b> Would you like to know about restaurants near Kiyomizu temple?</li> </ul>
	...

---

Figure 3: Response by Candidate Domain-plans

---

<b>Sys 1:</b>	Welcome to the Kyoto city voice portal. For information on a tourist spot or event, simply ask a question. For example “How do I get to Kiyomizu temple?”
<b>User 1:</b>	<i>Umm...</i>
<b>Sys 2:</b>	<ul style="list-style-type: none"> <li>• <b>[Repeat Question]</b> For information on a tourist spot or event, ask a question such as “Tell me how to get to Kiyomizu temple.”</li> <li>• <b>[Ask again]</b> Could you say that again please.</li> <li>• <b>[Single word guidance]</b> Please state the name of a tourist spot.</li> <li>• <b>[Give example response]</b> Please state the name of a tourist spot, for example “Kiyomizu temple” or “The golden pavilion”</li> </ul>
	...

---

Figure 4: Response by Candidate Utterance-plans

**Prerequisite condition:** Condition to determine whether the plan can be applied at the current dialogue state, based on dialogue history.

**Action:** Action to apply when the plan is selected; sets dialogue sub-task for a *dialogue-plan* and defines the system response for an *utterance-plan*.

**Evaluation function:** Appropriateness to apply plan at this time; linear function of user and situation model attributes. This function is automatically derived during training.

The prerequisite condition and action for each plan are defined manually. The evaluation function, however, is derived from a training set of dialogue examples. Two *utterance-plans* for the tourist domain are shown in Figure 2. Both plans describe the system response when the tourist spot name has yet to be defined but provide alternative responses based on the state of the user model. When user hastiness is high, the first plan, providing a more concise response, should be applied, however, if hastiness is low the second plan is more appropriate.

A set of *domain-plans* and *utterance-plans* are initially created to provide various dialogue strategies in response to a users’ query. An example set of *domain-plans* which are possible candidate for the current system response, are shown in Figure 3. That plan with the highest evaluation function score will be selected by the system, and the appropriate action will be performed. In this example, if it is currently lunch-time, the third response suggesting restaurant information is most likely to be selected,

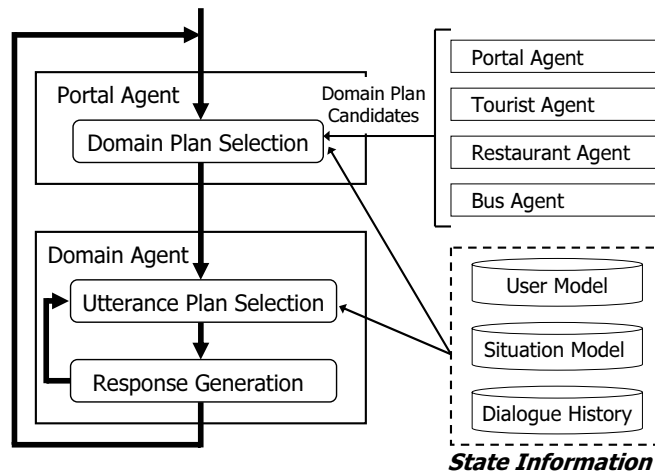


Figure 5: Plan Selection Flow

if access is inconvenient bus information may also be provided. Similarly, in Figure 4 a set of candidate *utterance-plans* are shown. Here alternative system responses are provided, enabling the system to generate responses adapted to user skill level. In the system evaluated in Section 5, a total of 16 *domain-plans*, and 97 *utterance-plans* were initially defined.

#### 4.1 Dialogue Planning

Dialogue planning is performed in two steps as shown in Figure 5. First, a *domain-plan* is selected by the portal agent. In this step, each domain agent submits candidate *domain-plans*, where prerequisite conditions match the current dialogue state. The evaluation function for each plan is then calculated, taking into account, user and situation model attribute values. The plan with maximum score is selected and the dialogue domain and sub-task is set. The second stage involves selecting an appropriate *utterance-plan* from the candidates belonging to the active *domain-plan*. The active domain agent calculates the evaluation score for each applicable *utterance-plan* and selects that with maximum score. A system response is then generated by performing the *utterance-plan* action. The selection of an *utterance-plan* at each dialogue turn is repeated until a “domain-termination” action is encountered. Once a *domain-plan* is completed the portal agent regains control and begins *domain-plan* selection again.

#### 4.2 Plan Evaluation Function

The plan evaluation function determines the appropriateness of applying a plan at a certain time, and is a linear discriminate function of user and situation model attributes, as shown below.

$$E_p(\mathbf{m}) = \mathbf{M}_p \cdot \mathbf{m} \quad (1)$$

$E_p(\mathbf{m})$  is the evaluation function for plan  $p$  and is calculated as the dot product of the vector  $\mathbf{m}$ , consisting of user and situation model attribute values ( $\mathbf{m} = (\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n, \mathbf{1})$  where  $n$  is the total number of model attributes) and  $\mathbf{M}_p$ , a vector of discriminate weights. The components of  $\mathbf{M}_p$ ,  $(M_{p,1}, M_{p,2}, \dots, M_{p,n}, M_{p,n+1})$ , correspond to weights for each model attribute. The final element,  $M_{p,n+1}$ , is a priori weight for the plan  $p$ , and is independent of the user and situation models. The elements of  $\mathbf{M}_p$  are trained using a machine learning algorithm as described below.

```

repeat n times
for each training sample
  for each plan  $p_i$  (where prerequisite conditions are met)
    if  $E_{p_i}(\mathbf{m}) > \mathbf{E}_{p_A}(\mathbf{m})$  (where  $p_A$  is correct plan)
      for each model value  $m_j$ 
        if  $m_j == 1$ 
           $M_{A,j} \leftarrow M_{A,j} + 1/k$ 
           $M_{i,j} \leftarrow M_{i,j} - 1/k$ 
          ( $k$  =no. of incorrect plans)

```

Figure 6: Evaluation Function Training Algorithm

### 4.3 Training of Evaluation Functions

The evaluation functions control the overall dialogue planning process. These functions are trained automatically with a set of dialogue role-playing examples. The training algorithm is described in Figure 6. For each sample in the training data, plans whose prerequisite conditions match the current dialogue state are selected and the evaluation function of each plan is calculated. If the correct plan, as labeled in the training data, does not have the maximum evaluation score, the discriminate weights for that and the competing plans are updated, model weights are increased for the correct plan, and decreased for all competing plans, as shown in Figure 6. The training process is iterated over the entire training set multiple ( $n$ ) times to reduce the effect of ambiguities in the training data.

## 5 Experimental Evaluation

### 5.1 Data Collection based on Role-Playing Dialogue

Evaluation was performed with eight subjects. Subject participated in up to 8 dialogues, each corresponding to a distinct dialogue scenario. Initially users were provided with an explanation of the spoken dialogue system, and allowed to use the system a number of times.

Subjects participated in an interactive role-playing dialogue based on a given scenario. Each scenario provided information regarding the task goal and situation. At each dialogue turn, a set of possible system responses were displayed and the user selected that response they thought was most suitable for the current dialogue state, allowing the dialogue to proceed in a role-playing game manner. The data gained at each dialogue turn consisted of dialogue history, correct system response (as selected by the user), and situation and user model information.

### 5.2 Evaluation of Dialogue Plan Selection

Using the data collected, the performance of the proposed dialogue planning scheme was evaluated in its ability to adapt to individual users. In this experiment, data from a single user was applied for both system training and evaluation. The system performance was evaluated by performing cross validation, where the plan selection accuracy for one dialogue was evaluated based on a system trained on the remaining dialogue data. The average selection accuracy for each user for the *domain-plan* and *utterance-plan* is shown in Table 3. As dialogue is not constrained, the number of dialogue turns varies significantly between users. There is a large difference in plan selection accuracy from 40.9% to 68.4% for *utterance-plan* selection, and 0.0% to 75.0% for *domain-plans*, although the number of *domain-plan* samples is much smaller.

The average system performance for all eight users is shown in Table 4. For the proposed approach

Table 3: Plan Selection Accuracy for each User

	no. Dialogues	Plan Selection Accuracy		
		Domain-Plan (count)	Utterance-Plan (count)	Total
User 1	8	25.4% (21)	66.3% (86)	59.5%
User 2	8	16.6% (11)	40.9% (55)	36.5%
User 3	8	12.5% (28)	60.3% (157)	53.0%
User 4	8	12.5% (11)	60.3% (54)	53.0%
User 5	8	34.2% (23)	68.4% (98)	60.9%
User 6	6	44.0% (11)	59.4% (43)	56.5%
User 7	5	0.0% (9)	58.2% (41)	38.4%
User 8	4	75.0% (3)	63.6% (22)	64.9%
Total	55	27.5%(117)	57.0%(556)	50.5%

Table 4: Average Plan Selection Accuracy

	Domain-Plan	Utterance-Plan	Total
Chance Rate	9.9%	24.5%	21.8%
Classification	27.5%	57.0%	50.5%

average plan selection accuracy is 50%, this is much higher than the chance rate when plans are selected randomly from the set of candidate responses.

### 5.3 Discussion of Experimental Results

In the experimental task, the system attempts to select an optimal response, from a set of 113 predefined plans (13 dialogue plans, and 97 *utterance-plans*) taking into account the seven internal attributes from the user and situation models. All candidate responses generated by the system are acceptable as they are selected using prerequisite conditions, however, selecting the single best response, as defined by a user, is very difficult. For this difficult task the proposed planning approach obtains reasonable performance.

The proposed approach enables a complex dialogue planner, incorporating information from multiple internal information models, to be automatically trained, enabling a cooperative dialogue system to be realized. Future work will investigate methods to improve system robustness, such as allowing users to select multiple response candidates or to rank candidates during role-playing simulation, and approaches to improve training with sparse data.

## 6 Conclusion

We investigated a dialogue planning approach that generates cooperative responses adaptively to individual users and situations. We introduced user and situation models for adaptive dialogue planning. To overcome the problems of manually defining dialogue strategies that take into account all possible model combinations, we proposed a novel machine learning based training scheme. In the proposed approach, data is collected using a role-playing simulation, and this data is then used to train the dialogue planner, applying machine learning.

The proposed scheme was evaluated on the Kyoto city voice portal spoken dialogue system. Eight subjects participated in a role-playing experiment where they selected appropriate system responses at each dialogue turn. In this difficult task, the proposed approach had reasonable performance. Plan selection accuracy of 50% was gained where the selected domain and response matched that chosen by the user.

## References

- [1] D. J. Litman, and S. Pan, "Predicting and adapting to poor speech recognition in a spoken dialogue system." In Proc. AAAI, 2000.
- [2] J. Chu-Carroll, "MIMIC: An adaptive mixed initiative spoken dialogue system for information queries." In Proc. ANLP, pp.97–104, 2000.
- [3] L. F. Lamel, S. Rosset, J-L. S. Gauvain, and C. K. Bennacef, "The LIMSI ARISE system for train travel information." In Proc. IEEE-ICASSP, 1999.
- [4] S. Lee, E. Ammicht, E. Fosler-Lussier, H.-K. Kuo, and A. Potamianos, "Spoken dialogue evaluation for the bell labs communicator system." In Proc. Human Language Technologies Conference, pp.745–748, 2002.
- [5] S. Larsson, and R. Cooper, "An information state approach to natural interactive dialogue." In Proc. LREC Workshop Proceedings - From Spoken Dialogue to Full Interactive Dialogue, pp.7–7, 2000.
- [6] D. Sadek, "Design considerations on dialogue systems: From theory to technology -the case of artimis-." In Proc. ESCA Workshop on Interactive Dialogue in Multi-Modal Systems, 1999.
- [7] B. Pakucs, "Sesame: A framework for personalized and adaptive speech interfaces." In Proc. EACL-03, Workshop on Dialogue Systems: interaction, adaptation and styles of management, pp.745–748, 2003.
- [8] W. Eckert, E. Levin, and R. Pieraccini, "User Modeling for spoken dialogue system evaluation." In Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, pp.80–87, 1997.
- [9] D. Buhler, W. Minder, J. Haubler, S. Kruger, "Flexible multimodal human-machine interaction in mobile environments." In Proc. ICSLP, pp.87–96, 2002.
- [10] K. Komatani, S. Ueno, T. Kawahara, and H. G. Okuno, "User modeling in spoken dialogue systems for flexible guidance generation." In Proc. EUROSPEECH, pp.745–748, 2003.
- [11] J. R. Quinlan, "C4.5: Programs for Machine Learning." Morgan Kaufmann, San Mateo, CA, 1993. <http://rulequest.com/see5-info.html>.