An Empirical Evaluation of User Tailored Generation in MultiModal Dialogue *

M.A.Walker a S.J.Whittaker a A. Stent b P. Maloor c J. Moore d
M. Johnston c G. Vasireddy c

a University of Sheffield
Sheffield, England, S1 4DP

b SUNY at Stony Brook
Stony Brook, NY, USA, 11794

c AT&T Labs - Research
Florham Park, NJ, USA, 07932
d University of Edinburgh
Edinburgh, Scotland, EH8 9LW

Abstract

When people engage in conversation, they tailor their utterances to their conversational partners, whether these partners are other humans or computational systems. This tailoring, or adaptivity to the partner, has been shown to take place in all facets of human language use, and is based on a mental model or a user model of the conversational partner. Such adaptation has been shown to improve listeners’ comprehension, their satisfaction with an interactive system, the efficiency with which they execute conversational tasks, and the likelihood of achieving higher level goals such as changing the listener’s beliefs and attitudes. Our focus here is on one aspect of adaptivity, namely the tailoring of the content of dialogue system utterances for the higher level processes of persuasion, argumentation and advice-giving. Our hypothesis is that algorithms that adapt content for these processes, according to a user model, will improve the usability, efficiency, and effectiveness of dialogue systems. We describe a multimodal dialogue system and algorithms for adaptive content selection based on multi-attribute decision theory. We demonstrate experimentally the improved efficacy of system responses through the use of user-models to both tailor the content of system utterances and to manipulate their conciseness.

Key words: dialogue systems, user modeling, multimodal, user-tailored generation

* This work was partially supported by DARPA grant MDA 972 99 3 0003 to AT&T Labs Research.

Email address: walker@dcs.shef.ac.uk (M.A.Walker).

Preprint submitted to Elsevier Science
1 Introduction

When people engage in conversation, they tailor their utterances to their conversational partners, whether these partners are other humans or computational systems [Brennan, 1991]. This tailoring, or adaptivity to the partner, has been shown to take place in all facets of human language use, from speaking rate and response delay [Ward and Nakagawa, 2002, Darvės and Oviatt, 2002], to amplitude and prosodic range [McLemore, 1992, Coulston et al., 2002], in lexical and syntactic choice [Levelt and Kelter, 1982, Kempen and Hoenkamp, 1987, Brennan, 1996], choice and modality of referring expressions [Garrod and Anderson, 1987, Brennan and Clark, 1996, Schober, 1998, Bell et al., 2000] and in higher level discourse processes such as the selection of content and form for persuasive arguments and negotiation [Mayberry and Golden, 1996, McGuire, 1968]. This adaptive behavior is based on a mental model or a user model of the conversational partner [Brennan and Clark, 1996, Levelt, 1989, Zukerman and Litman, 2001, Wahlster and Kobsa, 1989]. Such adaptation has been shown to improve listeners’ comprehension, their satisfaction with an interactive system [Nass et al., 1995], the efficiency with which they execute conversational tasks [Brennan, 1996, Clark and Wilkes-Gibbs, 1986], and the likelihood of achieving higher level goals such as changing the listener’s beliefs and attitudes [Luchok and McCroskey, 1978, GiuseppeCarenini and Moore, 2001, Carenini and Moore, 2000b, Zukerman and McConachy, 1993].

Our focus here is on one aspect of adaptivity, namely the tailoring of the content of dialogue system utterances for the high level processes of persuasion, argumentation and advice-giving. Dialogue systems are one of the few examples of an intelligent artifact that can interact with humans to carry out a variety of tasks. Various hypotheses about conversational interaction can be tested in dialogue systems by implementing algorithms that control the system’s conversational behavior. As such dialogue systems provide an important experimental vehicle for cognitive science and theories of interaction. In addition, recent technological advances have made it feasible to experiment directly with fully implemented real-time speech-enabled dialogue systems [Walker et al., 2002], rather than facsimiles, such as wizarded simulations or mock-ups. This makes it possible to test cognitive hypotheses about the system’s behavior in the actual context of use, an important requirement for algorithms intended to support real-time interaction with humans [Sche gloss, 1982, Zukerman and Litman, 2001, Oviatt, 1999a].

Our hypothesis is that algorithms that adapt content for higher level discourse processes, according to a user model, will improve the usability, efficiency, and effectiveness of dialogue systems. Dialogue systems have a particularly strong need to produce concise, informative and relevant utterances, especially during the information presentation phase of the dialogue. In this phase, the
system has queried a database for options that match a user's constraints, and needs to present these options to the user. It is important for the system to present the options in a form that will help the user understand and evaluate the tradeoffs among various options. Dialogue strategies for recommending particular options, or for making balanced comparisons between options should help users make such evaluations.

Recommendations for particular options, and comparisons among options, are one form of evaluative arguments. An evaluative argument typically consists of a main claim and evidence relevant to the claim. Argumentation theory provides a number of guidelines for producing effective evaluative arguments [Mayberry and Golden, 1996, Miller and Levine, 1996, Corbett and Connors, 1999, McGuire, 1968, Zukerman et al., 2000], which are summarized by Carenini and Moore [Carenini and Moore, 2000a]. These guidelines require:

1. Identifying supporting and opposing evidence: Evidence must be based on a model of the user's values and preferences, e.g. superb restaurant decor can only be used to support an argument for going to a restaurant if the user is oriented to decor.
2. Positioning the main claim: placing the main claim first helps users follow the line of reasoning, but delaying the claim until the end of the argument can also be effective if the user is likely to disagree with the claim.
3. Selecting supporting and opposing evidence: An argument cannot include all the possible evidence, so only strong evidence should be presented in detail, and weak evidence only briefly mentioned or omitted entirely.
4. Arrangement of supporting evidence: The strongest support should be presented first, but if possible one effective piece of supporting evidence should be saved for the end to leave the user with a final impression of the strength of the argument;
5. Addressing and ordering opposing evidence: The options are not to mention any opposing evidence, to acknowledge it without refuting it, or to acknowledge it and refute it. The order should minimize its effectiveness with strong opposing evidence in the middle and weak evidence at the beginning and end;
6. Ordering between supporting and opposing evidence: If the reader is aware of the opposing evidence, then it should come before the supporting evidence, otherwise after.

Our research has been carried out with the goal of improving the dialogue interaction capabilities of the MATCH (Multimodal Access to City Help) multimodal dialogue system, a system that provides information on restaurant and entertainment options in New York City [Johnston et al., 2002]. We aim to modify MATCH's response generation algorithms to produce effective recommendations and comparisons by drawing on these guidelines from argumentation theory. However, as Carenini and Moore point out [GiuseppeCarenini
and Moore, 2001], they must first be formalized to be used in a computational system. The formalization requires representing the user’s values and preferences (guideline 1), providing a way to measure the strength of supporting or opposing evidence (guidelines 3, 4, 5), representing whether the user is aware of certain facts (guideline 6), and developing strategies for ordering and structuring the selected content into coherent and persuasive arguments (guidelines 2, 4, 5, 6).

Carenini and Moore formalized these guidelines in a computational system for interactive data exploration in the real estate domain, and evaluated their effectiveness [Carenini and Moore, 2000b, a, Giuseppe Carenini and Moore, 2001, Carenini, 2000]. They base their operationalization of the user model on multi-attribute decision theory, drawing on work by Klein [Klein, 1994]. Multi-attribute decision theory provides both a way to represent the user’s values and preferences and to measure the strength of supporting and opposing evidence (as we explain in more detail below). The strength of evidence measure then is the basis for strategies for selecting and structuring the content of the argument and making it concise. As mentioned above, dialogue systems have a strong requirement to produce concise, informative, and relevant utterances because of users’ difficulties in processing and making decisions about complex information. We extend Carenini and Moore’s work, developing algorithms based on similar user models and their observations about argumentation theory. To our knowledge, such algorithms have never been embodied in a real-time multimodal dialogue system and been evaluated for effectiveness. We present a detailed comparison between our work and previous research in Section 6.

Section 2 describes the MATCH system and how the user-tailored dialogue strategies are used to support interaction. We also describe how we can use MATCH to test various cognitive hypotheses about user tailored interaction. Section 3 describes the use of multi-attribute decision theory for user modeling and provides detailed examples of user models from our user group. Section 4 describes the content selection algorithms based on the user models, and how they are utilized in dialogue strategies based on argumentation theory. Section 5 describes the design, hypotheses, and results of two evaluation experiments, which demonstrate the benefits of tailoring and the benefits of the user models in manipulating the conciseness of utterances. We delay a detailed review of relevant prior research to Section 6, and present conclusions and future work in Section 7.
2 The MATCH multimodal dialogue system

Our research on user-tailored adaptation in dialogue has been embodied in the MATCH (Multimodal Access to City Help) multimodal dialogue system [Johnston et al., 2002]. MATCH is designed to run on a small, portable, tablet computer, and to provide a testbed for research on multimodal dialogue interaction in a mobile setting. See Figure 2. Users interact with MATCH using a multimodal user interface client. The tablet screen is divided into an area showing a street map of New York City (See Figures 3 and 4), a box for input and output of text, a table showing options that match the user’s current request (if any), and control buttons for input and output of speech and accessing error messages and system state. The street map is dynamic, and can be panned and zoomed [DiFabbrizio et al., 1999, 2000]. Users may take a turn in the dialogue with speech, gesture, or handwriting or by a combination of these [Johnston and Bangalore, 2000, Johnston, 1998, Bangalore and Johnston, 2000], as in previous systems for interacting with maps [Oviatt, 1997, Cohen et al., 1998, Oviatt, 1999a, Wu et al., 1999]. The system’s responses are generated in speech, or using the graphical display, or using the text window, or with a combination of these modes. AT&Ts Watson engine is used for speech recognition [Sharp et al., 1997], and AT&Ts Natural Voices engine is used for text-to-speech (TTS) [Beutnagel et al., 1999].

Our goal is to improve the system’s dialogue interaction capabilities by making it easier for users to understand the tradeoffs between different restaurant options. Our hypothesis is that this can be done in three ways: (1) By providing the system with the ability to respond to requests for recommending one of a
“Show me Italian restaurants in the West Village”

[zooms to the West Village, displays Italian restaurants. (Figure 3)]

[circles some highlighted restaurants and says “summarize”. (Figure 4)]

“The 3 restaurants have the same price range of 28 to 33 dollars. They differ in food quality and decor.”

[circles a selection of restaurants and says “compare” (Figure 5)]

“Among the selected restaurants, the following offer exceptional overall value. Uguale’s price is 33 dollars. It has excellent food quality and good decor. Da Andrea’s price is 28 dollars. It has very good food quality and good decor. John’s Pizzeria’s price is 20 dollars. It has very good food quality and mediocre decor.”

[writes word “recommend”]

“Uguale has the best overall value among the selected restaurants. Uguale’s price is 33 dollars.”

Fig. 2. Example dialogue with MATCH, U = user, S = System

set of restaurants, for comparing small sets of restaurants, or for summarizing the similarities and differences among a group of restaurants; (2) By making these recommendations, comparisons and summaries tailored to a model of the user’s individual preferences; and (3) by making the responses sufficiently concise for the user to understand and remember important information.

Figure 2 shows a sample dialogue with MATCH using the user-tailored dialogue capabilities described in the rest of the paper. The summary, comparison and recommendation examples are for the user OR (See Figure 8 for OR’s user model).

In Figure 2, in utterance U1, the user specifies the query Show Italian Restaurants in the West Village in speech. The system responds in S1 by presenting a map of New York, zooming to the West Village and highlighting Italian restaurants (Figure 3). At this point, the user has too many options to decide between and so s/he circles some highlighted restaurants (Figure 4) and says summarize (U2). The system then produces a summary (S2), highlighting options and attributes relevant to the user. The user decides to select a different set with a gesture (Figure 5) and compare them (U3). S3 is that comparison. Since all the restaurants mentioned in S3 are acceptable the user asks the system to recommend one by writing the word “recommend” (U4). The recommendation operates on the current dialogue context which is the selected set (from U3). This example illustrates our goal of allowing users to finesse the problem of having too many complex options to evaluate by presenting compact descriptions, and highlighting only those options and attributes that are directly relevant to the user.
Fig. 3. MATCH’s graphical system response to *Show me Italian Restaurants in the West Village.*

Fig. 4. User circles subset for summary using a pen gesture.

The role of the user model on system responses is to affect both the ranking of options returned from the database and the selection of which attributes to mention in a recommendation, summary, or comparison. The user model makes a prediction about which restaurant options the user is likely to accept,
Fig. 5. User circles subset of Italian West Village restaurants for comparison.

and which facts about each restaurant are most relevant to the user. Note also that the user model reflects a user’s dispositional biases about restaurant selection, but these can be overridden by situational constraints specified in a user query. For example, the user models (as described below) allow us to represent the fact that some users have strong preferences for particular food types. However, in a particular dialogue situation, these can be overridden by interactively requesting a different food type, e.g. Italian food as in Figure 2. Thus dispositional biases never eliminate options from the set of options returned by the database, they simply affect the ranking of options, and the weighting of their attributes.

The primary hypothesis that we wish to test through user interactions with the MATCH system is that user-tailored responses are more effective. In the evaluation experiments described below, we compare the users’ evaluation of dialogue responses tailored to their own model, vs. responses tailored to a randomly selected model of another user. We also utilize the strength of evidence as defined by the user model to vary the conciseness of system responses, and compare user’s evaluation of terse, concise and verbose dialogue responses. Sample recommendations for a task of finding a Japanese restaurant in the East Village for two different users, with varying levels of conciseness as generated by our algorithms are shown in Figure 6. The user model leads to selection of different restaurants to recommend and to mentioning different facts depending on the user model and the conciseness parameter $z$.

A second set of hypotheses concern potential interactions between user-tailoring
<table>
<thead>
<tr>
<th>User</th>
<th>Concise?</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>CK</td>
<td>Concise (z = 0.3)</td>
<td>Bond Street has the best overall value among the selected restaurants. Bond Street has excellent food quality.</td>
</tr>
<tr>
<td>BA</td>
<td>Concise (z = 0.3)</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's price is 29 dollars. It's a Japanese, Latin American restaurant.</td>
</tr>
<tr>
<td>CK</td>
<td>Sufficient (z = -0.7)</td>
<td>Bond Street has the best overall value among the selected restaurants. Bond Street's price is 51 dollars and it has excellent food quality and good service. It's a Japanese, Sushi restaurant.</td>
</tr>
<tr>
<td>BA</td>
<td>Sufficient (z = -0.7)</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's price is 29 dollars and it has very good service and very good food quality. It's a Japanese, Latin American restaurant.</td>
</tr>
<tr>
<td>CK</td>
<td>Verbose (z = -1.5)</td>
<td>Bond Street has the best overall value among the selected restaurants. Bond Street's price is 51 dollars and it has excellent food quality, good service and very good decor. It's a Japanese, Sushi restaurant.</td>
</tr>
<tr>
<td>BA</td>
<td>Verbose (z = -1.5)</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's price is 29 dollars and it has very good service, very good food quality and good decor. It's a Japanese, Latin American restaurant.</td>
</tr>
</tbody>
</table>

Fig. 6. Recommendations for Users CK and BA, for the East Village Japanese Task, of Varying Levels of Conciseness.

and the mode in which information is presented in a multimodal dialogue system, i.e. in speech or in text. Prior research has shown that unimodal speech may be less effective than text for the presentation of complex information because the transient nature of speech increases the cognitive load of remembering information [Whittaker et al., 1991]. In multimodal systems, one mode can sometimes compensate for the limitations of another [Oviatt, 1997, McKeeown et al., 1998, Mittal et al., 1995], but there is often insufficient screen real estate with small mobile devices such as MATCH to present information redundantly in all modes. Our hypotheses explore some of the underlying principles governing mode allocation. Consistent with prior research, we expected the ephemeral nature of speech (and the resulting cognitive load) would make this a less effective output mode than text. However we also expect that tailoring might address some of the inherent limitations of speech, so that spoken information that is user-tailored will be more effective than untailored textual presentations.

3 Multi-Attribute Decision Models in the Restaurant Domain

User models derived from multi-attribute decision theory have been shown to be effective for guiding user interaction in various types of interactive systems, including dialogue systems [Thompson and Goker, 1999, Jameson et al., 1995, Klein, 1994, Linden et al., 1997, GiuseppeCarenini and Moore, 2001]. They have also been found to be good predictors of user’s consumer behavior [Solomon, 1998]. For our current purposes they have two other important properties, namely (a) they are quantitative, which makes them easy to operationalize (b) it is relatively easy to gather the data necessary for constructing
Multi-attribute decision models are based on the claim that if anything is valued, it is valued for multiple reasons [Keeney and Raiffa, 1976]. In the restaurant domain, this implies that a user’s preferred restaurants optimize tradeoffs among restaurant attributes. To define a model for the restaurant domain, we must determine the attributes and their relative importance for particular users. We use a standard procedure called SMARTER, that has been shown to be a reliable and efficient way of eliciting multi-attribute decision models for particular users or user groups [Edwards and Barron, 1994].

3.1 Structure of the Model

The first step of the standard SMARTER procedure is to determine the structure of a tree model of the objectives in the domain. In MATCH, the top-level objective is to select a good restaurant. User interviews along with an analysis of online restaurant databases indicated that six attributes contribute to this objective: the quantitative attributes food quality, cost, decor, and service; and the categorical attributes food type and neighborhood. These attributes are structured into the one-level tree shown in Figure 7. A more complex structure that grouped decor, neighborhood and service under a higher level objective called ambiance was considered, but informal questioning of users suggested this structure was less intuitive.

The structure is user-independent with user-dependent weights on the branches as explained below. We apply this to a database of approximately 1000 restaurants populated with information freely available from the web. Values for each of these attributes for each restaurant are stored in the database.
3.2 Normalizing attribute values

The second step is to transform the real-domain values of attributes $x$ into single-dimension cardinal utilities $u(x)$ such that the highest attribute value is mapped to 100, the lowest attribute value to 0, and the others to values in the interval 0 to 100. This is necessary to normalize the values of the different attributes. In the restaurant database that we accessed from the Web *food quality, service* and *decor* range from 0 and 30, with higher values more desirable, so 0 is mapped to 0 and 30 to 100 in our model. The *cost* attribute ranges from $10 and $90 and higher values are less desirable, so $90 is mapped to 0 on the utility scale. Preferred values for categorical attributes such as *food type* are mapped to 90, dispreferred values to 10 and others to 50.\(^1\) Table 1 shows the attributes in the restaurant domain, with the functions mapping the values of each attribute in the web database into cardinal utilities.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Range of values</th>
<th>Mapping of values to cardinal utilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food quality, Service, Decor</td>
<td>0–30</td>
<td>value x 3 1/3</td>
</tr>
<tr>
<td>Cost</td>
<td>0–90</td>
<td>100 - (10/9 x value)</td>
</tr>
<tr>
<td>Food type, Neighborhood</td>
<td>e.g. Italian, French, West Village</td>
<td>top values listed by user are mapped to 90, bottom ones to 10 and all others to 50</td>
</tr>
</tbody>
</table>

Table 1
Mapping of attribute values to utilities in the restaurant domain

The vector of $u(x)$ values are aggregated into a scalar in order to determine the overall utility $U_h$ of each option $h$. The most widely used model for such aggregations is the additive model (over 95% of models used in practice are additive), and standard heuristic tests with users suggested that an additive model is a good approximation [Edwards and Barron, 1994]. Use of an additive model means that each attribute is assumed to be independent of every other one. The individual attribute utilities are combined into an overall utility using a simple additive function; the value for each attribute is multiplied by its weight and all the weighted values are summed. Thus, if $h$ ($h = 1,2, \ldots H$) is an index identifying the restaurant options being evaluated, $k$ ($k = 1,2, \ldots K$) is an index of the attributes, and $w_k$ is the weight assigned to each

\(^1\) This simplification is motivated by the large number of food types available in New York City and our requirement to keep the enrollment process short and simple.
attribute:

$$U_h = \sum_{k=1}^{K} w_k u_k(x_{hk})$$

### 3.3 Allocating Weights to Attributes

The final step of decision model construction is the assignment of specific weights $w_k$ to each attribute $k$. Attribute weights are user-specific, reflecting individual preferences about tradeoffs between options in the domain, and are based on users’ subjective judgments elicited using the SMARTER elicitation procedure. SMARTER’s main advantage over other elicitation procedures is that it only requires the user to specify the ranking of domain attributes. There is considerable experimental evidence showing that simple attribute ranking is both efficient, and nearly as accurate as more time-consuming methods, whereby users allocate weights directly [Edwards and Barron, 1994, Srivastava and Connolly, 1995].

SMARTER also specifies the standard form of questions used to elicit the rankings for a user model. We implemented these as a sequence of web pages. The first web page says *Imagine that for whatever reason you’ve had the horrible luck to have to eat at the worst possible restaurant in the city. The price is 100 dollars per head, you don’t like the type of food they have, you don’t like the neighborhood, the food itself is terrible, the decor is ghastly, and it has terrible service. Now imagine that a good fairy comes along who will grant you one wish, and you can use that wish to improve this restaurant to the best there is, but along only one of the following dimensions. What dimension would you choose? Food Quality, Service, Decor, Cost, Neighborhood, or Food Type?* After the user chooses an attribute on this page, the scenario is repeated omitting the chosen attribute, until all attributes have been selected. Users are then asked to specify whether they have any *neighborhood* or *food type* likes or dislikes. We elicit a user model during the enrollment process, because it is common in spoken dialogue systems to enroll new users to construct a user profile [Walker et al., 2001c].

Given the ranking, the weights are calculated using the following equation, which guarantees that the total sum of the weights add to 1, a requirement for multi-attribute decision models:

$$w_k = \frac{1}{K} \frac{1}{i}$$
3.4 Resulting User Models

To date, 29 different user models have been elicited and stored in a database that MATCH uses. Figure 8 shows attribute weightings and likes and dislikes for five of these users. What is most striking about the table are the large differences between users. When differences in categorical preferences are taken into account, no two users in our sample are alike, but even if we only consider the relative importance of various attributes, we find that only two pairs of users are identical in the ranking of attributes. For 25 of these users, we found that cost and food quality are always in the top three attributes, but user BA ranked food type highest, followed by cost and service. Even for users who ranked both cost and food quality in their top three attributes, the relative importance of lower ranked attributes, such as decor, service, neighborhood and food type, varies widely. For example, every user in the sample rates the importance of service differently as reflected by the different weights in the Service column in Fig. 8. User CK in Figure 8 ranks decor as the least important attribute, while user OR ranks it third in importance, and users CK and SD rank food type as the second most important attribute while users OR and MSh rank food type last.

After examining these differences qualitatively, we decided it would be useful to be able to quantify the differences among user models. We defined a measure of the distance between user models which is simply the sum of the absolute values of the differences in the weights for each attribute in the user models. That is, for users $i,j$ and attributes $k$ indexed from $1 \ldots K$, with weights $w_k$,

$$distance_{ij} = \sum_{k=1}^{K} |w_{ki} - w_{kj}|$$

For example, the distance between users CK and VM in Figure 8 is 0.84, and the distance between users CK and BA in Figure 8 is .89. The average distance between user models in our current user group is .57. The distance metric enables us to manipulate differences between models and to quantify the effect of those manipulations.

4 The SPUR Speech planner

So far, we have described the nature of user models derived from multi-attribute decision theory. We now explain how these are used to generate outputs that are tailored to the specific user preferences in an interactive system.
<table>
<thead>
<tr>
<th>User</th>
<th>FQ</th>
<th>SVC</th>
<th>Dec</th>
<th>Cost</th>
<th>Nbhd</th>
<th>FT Likes</th>
<th>Nbhd Dislikes</th>
<th>FT Likes</th>
<th>FT Dislikes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA</td>
<td>0.10</td>
<td>0.16</td>
<td>0.06</td>
<td>0.24</td>
<td>0.03</td>
<td>Downtown, Midtown, East Village, TriBeCa, SoHo</td>
<td>The Bronx, Harlem</td>
<td>Cajun, Creole, Greek, Italian, Japanese, Seafood</td>
<td>Coffeehouses, Desserts, German, Steak</td>
</tr>
<tr>
<td>CK</td>
<td>0.41</td>
<td>0.10</td>
<td>0.03</td>
<td>0.16</td>
<td>0.06</td>
<td>Midtown, Chinatown, TriBeCa, E. Village</td>
<td>Harlem, Bronx</td>
<td>Indian, Mexican, Chinese, Japanese, Seafood</td>
<td>Vegetarian, Vietnamese, Korean, Hungarian, German</td>
</tr>
<tr>
<td>OR</td>
<td>0.24</td>
<td>0.06</td>
<td>0.16</td>
<td>0.41</td>
<td>0.10</td>
<td>W. Village, Chelsea, Chinatown, TriBeCa, E. Village</td>
<td>Upper E. Side, Upper W. Side, Uptown, Bronx, Lower Manhattan</td>
<td>French, Japanese, Portuguese, Thai, Middle Eastern</td>
<td>no-dislike</td>
</tr>
<tr>
<td>MSb</td>
<td>0.41</td>
<td>0.10</td>
<td>0.06</td>
<td>0.24</td>
<td>0.16</td>
<td>Flatiron, Chelsea, W. Village, Midtown E., Midtown W.</td>
<td>Chinatown, Lower E. Side, E. Village, Upper E. Side, Upper W. Side</td>
<td>Indian, Mexican, Ethiopian, Thai, French</td>
<td>Steakhouse, Russian, Korean, Filipino, Diner</td>
</tr>
<tr>
<td>VM</td>
<td>0.24</td>
<td>0.10</td>
<td>0.03</td>
<td>0.41</td>
<td>0.06</td>
<td>Upper West Side</td>
<td></td>
<td>Cajun, Creole, Chinese, Coffeehouses, Indian, Tapas</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8. Example User Models: FQ = Food Quality, SVC = Service, DEC = Decor, Nbhd = Neighborhood, FT = Food Type

The module of MATCH that generates turns in dialogue using the user models described above is called SPUR (Speech Planning with Utilities for Restaurants). The user model is used by SPUR for two aspects of content selection: (1) The user model ranks the options returned from a database query, and the ranking is used by SPUR to select a subset of options to recommend or compare; (2) The user model determines a subset of attributes that are selected for each option, with the size of the subset depending on the setting of a conciseness parameter.

SPUR takes as input: (1) a speech-plan goal; (2) a user model; and (3) a set of restaurant options returned by the database matching situational constraints specified in the user's query. Given the options, SPUR constructs a speech-plan specific to each speech-plan goal and user model. The aim of the resulting speech-plans is to filter the information presented to the user so that only options and attributes that are most relevant to the user are mentioned,
contrasted and highlighted. This should make it easier for the user to evaluate the trade-offs among options in a set, reducing dialogue duration and increasing user satisfaction.

We first illustrate how the user model reranks the option set to which we then apply our system dialogue strategies, and describe the details of the three speech-plan types: Recommend, Compare and Summarize.

### 4.1 The Effect of User Model on Option Ranking

<table>
<thead>
<tr>
<th>User</th>
<th>Restaurant</th>
<th>$U_h$</th>
<th>FQ (wtd.)</th>
<th>SVC (wtd.)</th>
<th>DEC (wtd.)</th>
<th>Cost (wtd.)</th>
<th>Nbhd (wtd)</th>
<th>FT (wtd)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA</td>
<td>Komodo</td>
<td>77</td>
<td>22(7)</td>
<td>22(10)</td>
<td>19(4)</td>
<td>29(18)</td>
<td>90(2)</td>
<td>90(36)</td>
</tr>
<tr>
<td>BA</td>
<td>Japonica</td>
<td>71</td>
<td>23(7)</td>
<td>18(7)</td>
<td>15(3)</td>
<td>37(16)</td>
<td>90(2)</td>
<td>90(36)</td>
</tr>
<tr>
<td>BA</td>
<td>Takahachi</td>
<td>71</td>
<td>21(6)</td>
<td>17(6)</td>
<td>14(2)</td>
<td>27(19)</td>
<td>90(2)</td>
<td>90(36)</td>
</tr>
<tr>
<td>BA</td>
<td>Shabu-Tatsu</td>
<td>70</td>
<td>20(5)</td>
<td>18(7)</td>
<td>15(3)</td>
<td>31(17)</td>
<td>90(2)</td>
<td>90(36)</td>
</tr>
<tr>
<td>BA</td>
<td>Bond Street</td>
<td>69</td>
<td>25(8)</td>
<td>19(8)</td>
<td>22(4)</td>
<td>51(11)</td>
<td>90(2)</td>
<td>90(36)</td>
</tr>
<tr>
<td>BA</td>
<td>Dojo</td>
<td>66</td>
<td>15(2)</td>
<td>12(2)</td>
<td>8(1)</td>
<td>14(23)</td>
<td>90(2)</td>
<td>90(36)</td>
</tr>
<tr>
<td>CK</td>
<td>Bond Street</td>
<td>63</td>
<td>25(34)</td>
<td>19(3)</td>
<td>22(2)</td>
<td>51(5)</td>
<td>50(7)</td>
<td>50(12)</td>
</tr>
<tr>
<td>CK</td>
<td>Japonica</td>
<td>59</td>
<td>23(29)</td>
<td>18(3)</td>
<td>15(1)</td>
<td>37(7)</td>
<td>50(7)</td>
<td>50(12)</td>
</tr>
<tr>
<td>CK</td>
<td>Komodo</td>
<td>59</td>
<td>22(26)</td>
<td>22(4)</td>
<td>19(2)</td>
<td>29(8)</td>
<td>50(7)</td>
<td>50(12)</td>
</tr>
<tr>
<td>CK</td>
<td>Takahachi</td>
<td>54</td>
<td>21(24)</td>
<td>17(2)</td>
<td>14(1)</td>
<td>27(8)</td>
<td>50(7)</td>
<td>50(12)</td>
</tr>
<tr>
<td>CK</td>
<td>Shabu-Tatsu</td>
<td>52</td>
<td>20(22)</td>
<td>18(3)</td>
<td>15(1)</td>
<td>31(7)</td>
<td>50(7)</td>
<td>50(12)</td>
</tr>
<tr>
<td>CK</td>
<td>Dojo</td>
<td>30</td>
<td>15(10)</td>
<td>12(1)</td>
<td>8(0)</td>
<td>14(10)</td>
<td>50(7)</td>
<td>50(12)</td>
</tr>
<tr>
<td>VM</td>
<td>Komodo</td>
<td>66</td>
<td>22(16)</td>
<td>22(7)</td>
<td>19(2)</td>
<td>29(31)</td>
<td>50(3)</td>
<td>50(7)</td>
</tr>
<tr>
<td>VM</td>
<td>Takahachi</td>
<td>61</td>
<td>21(14)</td>
<td>17(4)</td>
<td>14(1)</td>
<td>27(32)</td>
<td>50(3)</td>
<td>50(7)</td>
</tr>
<tr>
<td>VM</td>
<td>Japonica</td>
<td>58</td>
<td>23(17)</td>
<td>18(4)</td>
<td>15(1)</td>
<td>37(26)</td>
<td>50(3)</td>
<td>50(7)</td>
</tr>
<tr>
<td>VM</td>
<td>Shabu-Tatsu</td>
<td>57</td>
<td>20(13)</td>
<td>18(4)</td>
<td>15(1)</td>
<td>31(29)</td>
<td>50(3)</td>
<td>50(7)</td>
</tr>
<tr>
<td>VM</td>
<td>Bond Street</td>
<td>56</td>
<td>25(20)</td>
<td>19(5)</td>
<td>22(2)</td>
<td>51(19)</td>
<td>50(3)</td>
<td>50(7)</td>
</tr>
<tr>
<td>VM</td>
<td>Dojo</td>
<td>65</td>
<td>15(6)</td>
<td>12(1)</td>
<td>8(0)</td>
<td>14(39)</td>
<td>50(3)</td>
<td>50(7)</td>
</tr>
</tbody>
</table>

Fig. 9. Results of DB Query for East Village Japanese for users BA, CK and VM: $U_h$ = Overall Utility, WTD = Weighted utility for each attribute, FQ = Food Quality, SVC = Service, DEC = Decor, Nbhd = Neighborhood, FT = Food Type

As mentioned above, the user model reflects a user’s dispositional biases about restaurant selection, but these biases can be overridden by situational constraints specified in a user query. This means that a user’s model, reflecting dispositional biases, doesn’t eliminate any options from the set of options returned by the database in response to a user query. Rather, the role of the user model is to affect the ranking of options, and what is said about an option when it is described to the user.

To show the effects of user model on option ranking, we present the restaurant options that match the query *Show Japanese restaurants in the East Village.*
Figure 9 shows how the user models for CK, BA and VM ranks these options. The third column gives the overall utility. The subsequent columns give the attribute values and in parentheses the weighted utilities (WTD). Note that food quality contributes most strongly to the weighted utilities in the CK model ranking, while cost contributes most strongly to the ranking for both BA and VM. However, BA and VM differ in that VM’s second most important attribute is food quality, while for BA the second most important attribute is food type. This modifies the ranking of the restaurant set.

Let us consider in detail the differences in overall ranking for CK and VM resulting from different attribute weightings. Bond Street (a highly priced restaurant with excellent food quality) is fifth for VM because VM ranks cost first and food quality second. Bond Street’s 25 rating for food quality results in 34 utils (utils are units of weighted utility) for CK, but only 20 utils for VM. Also, Bond Street’s price of $51 per person results in only 19 utils for VM; all of the restaurants ranked higher by VM than Bond Street are less expensive. On the other hand, Komodo is more highly ranked for VM than CK. This is mainly because its modest price gets 31 utils for VM but only 8 for CK. Note also that Dojo, which is very inexpensive, is as good as Bond Street in overall utility for VM (both get 56) but for CK, Dojo’s lower food quality means that it has a much worse overall utility.

4.2 SPUR Speech Plans

We defined three types of speech-plans for SPUR: (1) RECOMMEND one of a selected set of restaurants; (2) COMPARE three or more selected restaurants; (3) SUMMARIZE a selected set of restaurants. For each response, SPUR outputs a speech-plan to the surface realizer, described below. Each speech-plan uses the overall utility $U_h$ to rank the options as described in Section 4.1. For recommendations, the algorithm selects the top-ranked option. For comparisons, the algorithm selects a top-ranked subset of options for comparison. Then the weighted attribute values are used to select the content for each option.

Conciseness is controlled with a parameter that determines whether an option or attribute is an outlier with respect to other options or attributes. Outliers are deemed worth mentioning because they deviate from the norm [Klein, 1994]. According to multi-attribute decision theory, the weighted attribute model also enables us in principle to determine the likelihood that mentioning a given attribute will change the user’s belief state. For example, compare the recommendations in Figure 6. The most concise recommendation for both CK and BA mentions one attribute. The weighted attribute values for each user in Figure 9 predict how convincing a recommendation would be that includes that attribute. Figure 9 indicates that telling CK about Bond Street’s food quality
should provide 34 util (units of utility) out of a possible 63. Similarly, telling BA about Komodo's *food type* is predicted to provide 36 util out of a possible 77. Including more attributes makes the recommendation more convincing, e.g. adding the *food type* attribute as in CK’s Sufficient recommendation in Figure 6 should provide 46 (34 + 12) util out of a possible 63 total util.

In sum, we map conciseness directly onto the weighted attribute ranking of the user model: more concise descriptions select the subset of attributes that maximally affect the user’s belief state. More verbose descriptions also include lower weighted attributes. Obviously, however, there is a trade-off between maximizing expected utility, and verboseness. Mentioning more attributes increases expected utility while requiring the user to remember more information.

Below, we first describe how outliers are identified (Section 4.2.1), and then describe the algorithms for constructing each type of speech plan. Section 4.3 describes the strategies for realizing the speech-plans.

### 4.2.1 Defining Outliers

Carenini and Moore define a response as *tailored* if it is based on a user’s known biases and preferences. A response is *concise* if it includes only those options with high utility, or possessing *outliers* with respect to a population of attribute values. We use the *z-score* (standard value) of an option’s overall utility, or of the weighted attribute value \( v \), to define an *outlier*:

\[
z(v) = \frac{v - \mu_v}{\sigma_v}
\]

The *z-score* expresses how many standard deviations \( \sigma_v \), a value \( v \), is away from the mean \( \mu_v \), of a population of values \( V \). The population of values \( V \) that are used to calculate \( \mu_v \) and \( \sigma_v \) can be (a) other attributes for the same option (for RECOMMEND), or (b) the same attribute for other options (for COMPARE). Depending on a threshold for \( z \), different numbers of options or attribute values are considered to be worth mentioning, because they stand out from other values. For example, when the threshold for \( z \) is 1.0, the weighted attribute values must be more than one standard deviation away from the mean for that attribute to be selected for expression. This threshold can obviously be modified to generate responses at different levels of conciseness. In the examples below, we illustrate responses for different settings of \( z \). In our experiments, to investigate the value of tailoring, we chose a value that we empirically validated would produce recommendations on average realizing two attributes for both recommendations and comparisons. In the conciseness experiment, which looked at conciseness more directly, we had users evaluate system responses generated using different threshold values, in order to
identify optimal conciseness levels.

We demonstrate below the differences in content selected for several different user models: (1) The user model for VM in Figure 8; (2) The user model for CK in Figure 8; and (3) The user model for BA in Figure 8. We compare the output for these three models because the different ranking of attributes in these models as shown in Figure 8 leads to different responses.

4.2.2 Recommendation Speech Plans

The system's strategy for making a recommendation is to select the best option (based on overall utility) and provide convincing reasons (based on weighted attribute values) for the user to choose it. Figure 10 provides the algorithm for selecting the content for the RECOMMEND speech plan.

1. Select the restaurant option \( R \) with highest overall utility from returned options.
2. Using the setting for \( z \), identify the attributes \( a_i \) whose weighted attribute values \( v_i \) for that option are outliers.
3. Construct a speech-plan with the claim that \( R \) has the best overall value, because \( R \) possesses attributes \( a_i \) with values \( v_i \), as exemplified in Figure 13.

Fig. 10. Algorithm for recommendation generation

First, consider the effect of the user model on recommendations at a fixed level of conciseness. Figure 11 shows sample responses, for the East Village Japanese task, for users CK, BA and VM. As discussed above, the option ranking generated Bond Street as the top option for CK, while Komodo is the top option for BA and VM. Thus the recommendations in Figure 11 recommend Bond Street to CK and Komodo to BA and VM. Then, for a \( z \)-value of 0.3 the attributes selected for each user are different, resulting in the three recommendations in Figure 11.

<table>
<thead>
<tr>
<th>User</th>
<th>( z ) value</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>CK</td>
<td>0.3</td>
<td>Bond Street has the best overall value among the selected restaurants. Bond Street has excellent food quality.</td>
</tr>
<tr>
<td>BA</td>
<td>0.3</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's price is 29 dollars. It's a Japanese, Latin American restaurant.</td>
</tr>
<tr>
<td>VM</td>
<td>0.3</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's price is 29 dollars and it has very good food quality.</td>
</tr>
</tbody>
</table>

Fig. 11. Recommendations for Users CK, BA and VM, for the East Village Japanese Task, for \( z = 0.3 \).

Now, consider the algorithm's implementation for the BA model for varying thresholds of \( z \). The setting for \( z \) determines the number of attributes selected to provide evidential support for recommending Komodo. In the experiments
reported here, for recommendations, $z$ ranges from -1.5 to 1.5. Generally this means that that there is usually at least one outlying attribute, even for the highest value of $z$. In certain cases, depending on the set of options returned, the algorithm may select no attributes, and then it simply makes the claim that $R$ has the highest overall value.

Sample responses for user BA at varying levels of conciseness are in Figure 12. Figure 9 provides the relevant utility and weighted attribute values. The weighted attribute values for Komodo are $7,10,4,18,2,36$ for *food quality*, *service*, *decor*, *cost*, *neighborhood* and *food type* respectively. Outliers are calculated for recommendations with respect to the values for other attributes for the same restaurant. Depending on the $z$-score different sets of attributes are determined to be outliers. When $z$ is 1.5 or 0.7, only the *food type* attribute is selected. When $z$ is 0.3, the attributes *cost* and *food type* are selected. When $z$ is -0.5, the attributes *cost*, *service* and *food type* are selected. When $z$ is -0.7, the attributes *cost*, *service*, *food quality* and *food type* are selected. When $z$ is -1.5, the attributes *cost*, *service*, *food quality*, *decor* and *food type* are selected.

<table>
<thead>
<tr>
<th>User</th>
<th>Z-value</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA 1.5</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's a Japanese, Latin American restaurant.</td>
<td></td>
</tr>
<tr>
<td>BA 0.7</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's a Japanese, Latin American restaurant.</td>
<td></td>
</tr>
<tr>
<td>BA 0.3</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's price is 29 dollars. It's a Japanese, Latin American restaurant.</td>
<td></td>
</tr>
<tr>
<td>BA -0.5</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's price is 29 dollars and it has very good service. It's a Japanese, Latin American restaurant.</td>
<td></td>
</tr>
<tr>
<td>BA -0.7</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's price is 29 dollars and it has very good service and very good food quality. It's a Japanese, Latin American restaurant.</td>
<td></td>
</tr>
<tr>
<td>BA -1.5</td>
<td>Komodo has the best overall value among the selected restaurants. Komodo's price is 29 dollars and it has very good service, very good food quality and good decor. It's a Japanese, Latin American restaurant.</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 12. Recommendations for User BA, for the East Village Japanese Task, of Varying Levels of Conciseness.

SPUR outputs a speech-plan to the realization engine. An example speech-plan that would be realized as the recommendation for user BA for $z$ of -0.7 is in Figure 13. The representation of the speech-plans is based on previous work [Marcu, 1997, Mellish et al., 1998], where each plan consists of a set of *assertions* about facts which must be communicated to the user and a set of *rhetorical relations* that hold between those assertions that may be communicated as well. Each rhetorical relation designates one or more facts as the *nuclei* of the relation, i.e. the main point, and the other facts as *satellites*, i.e. the supplementary facts [Mann and Thompson, 1987]. The speech-plan in Figure 13 specifies that the nucleus (main claim) is the assertion that *Komodo has the best overall value*., and that the satellites are the evidential support
for this assertion.

<table>
<thead>
<tr>
<th>strategy:</th>
<th>recommend</th>
</tr>
</thead>
<tbody>
<tr>
<td>items:</td>
<td>Komodo, Japonica, Takahachi, Shabu-Tatsu, Bond Street, Dojo</td>
</tr>
<tr>
<td>relations:</td>
<td>justify(nuc:1;sat:2); justify(nuc:1;sat:3); justify(nuc:1;sat:4); justify(nuc:1;sat:5)</td>
</tr>
<tr>
<td>content:</td>
<td>1. assert(best(Komodo))</td>
</tr>
<tr>
<td></td>
<td>2. assert(has-att(Komodo, cost(29)))</td>
</tr>
<tr>
<td></td>
<td>3. assert(has-att(Komodo, foodquality(verygood)))</td>
</tr>
<tr>
<td></td>
<td>4. assert(has-att(Komodo, service(verygood)))</td>
</tr>
<tr>
<td></td>
<td>5. assert(has-att(Komodo, footype(Japanese,Latin American)))</td>
</tr>
</tbody>
</table>

Fig. 13. A Speech-Plan representation for a recommendation for user BA for a Japanese restaurant in the East Village for z of -0.7.

4.2.3 Generating Comparison Speech Plans

The goal of a comparison is to generate several potential candidate options (those with highest overall utility) and provide the user with different reasons (expressed as different weighted attribute values) for choosing among them. Comparisons should help users evaluate tradeoffs among different options.

(1) If the number of restaurants is greater than 5 then

(1a) Select the restaurant options $R_i$ that are positive outliers for overall utility (outstanding restaurants). Add a claim $C_j$ to the speech-plan that the elements of the set $R_i$ have outstanding value.

(1b) If there are no outstanding restaurants, select the 5 highest ranked restaurant options $R_i$ for overall utility $U_h$. Add a claim $C_j$ to the speech-plan that the elements of the set $R_i$ are the top 5 in overall value.

Fig. 14. Algorithm for selecting a subset of options to compare

(1) For each option $R_i$, for each attribute $a_i$

(1a) If the weighted attribute value $v_i$ is an outlier when compared against the weighted attribute value for other options, then add attribute to $\$OUTLIER-LIST$.

(2) For each option $R_i$, for each attribute $a_i$ in $\$OUTLIER-LIST$, add an assertion $s_i$ to the speech-plan that $R_i$ has the attribute value $v_i$, and a relation that $s_i$ elaborates the claim $C_j$.

(3) For each assertion $s_i$ about an attribute $a_i$, add a contrast relation to the speech-plan with the $s_i$ as joint nuclei.

Fig. 15. Algorithm for selecting content for subset of options to compare

SPUR's COMPARE strategy can be applied to three or more options. If there are more than five, a subset are first selected according the algorithm in Figure 14. Then the content for each option is selected using the algorithm in Figure 15,
and descriptions reflect the number of options selected. Because comparisons are inherently contrastive, the algorithm in Figure 15 describes a procedure whereby if a weighted attribute value is an outlier for any option, the attribute value is realized for all options. This is motivated for two reasons: (1) it is not possible for the user to compare options without the same information about all of them; and (2) mentioning the same attributes about each option allows a parallel structure in the realization, which supports the user’s inference of contrast [Prevost, 1995, Prince, 1985, Meteer, 1991].

Figure 16 illustrates the effect of the user models on comparisons, for the East Village Japanese task, for a fixed level of conciseness. Each comparison selects a subset of options that are outliers for overall quality for the particular user, given the setting for z. In this case, the z-value of 0.3 selects a different number of options as outliers for each user, i.e. three options for CK, two for VM and only one for BA. The selected information that is provided about each restaurant option focuses on the attributes that are predicted to be both salient and significant for the specific user, because they are outliers with respect to the population of attribute values under consideration. The attributes whose values show significant variability are food quality, service and decor, which are all outliers for Komodo for z of 0.3. Thus these three attributes are selected for all of the comparisons, and their real values are realized as in Figure 16. The food type attribute is selected for user CK because of the larger set of restaurants from which outliers are calculated.

<table>
<thead>
<tr>
<th>User</th>
<th>Z value</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>CK 0.3</td>
<td>Among the selected restaurants, the following offer exceptional overall value. Bond Street’s price is 51 dollars. It has excellent food quality, good service and very good decor. It’s a Japanese, Sushi restaurant. Japonica’s price is 37 dollars. It has excellent food quality, good service and decent decor. It’s a Japanese, Sushi restaurant. Komodo’s price is 29 dollars. It has very good food quality, very good service and good decor. It’s a Japanese, Latin American restaurant.</td>
<td></td>
</tr>
<tr>
<td>VM 0.3</td>
<td>Among the selected restaurants, the following offer exceptional overall value. Komodo’s price is 29 dollars. It has very good food quality, very good service and good decor. Takahachi’s price is 27 dollars. It has very good food quality, good service and decent decor.</td>
<td></td>
</tr>
<tr>
<td>BA 0.3</td>
<td>Among the selected restaurants, the following offer exceptional overall value. Komodo has very good service, very good food quality and good decor.</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 16. Comparisons for Users CK, VM and BA, for the East Village Japanese Task.

Now, consider the algorithms in Figure 14 and Figure 15 applied using the VM user model for varying levels of conciseness. Figure 17 shows the variation in conciseness of comparisons for user VM for z values ranging from -1.5 to 1.5. Figure 9 shows the relevant values for overall utility \( U_h \) and weighted attribute values. Again, as in recommendations, the number of options and attributes selected depends on the value for z. Consider the algorithm’s operation for a z value of 0.3. The option selection algorithm in Figure 14 determines that Takahachi and Komodo are outliers for overall utility. Then z-scores for the weighted attribute values are calculated for each attribute across these op-
tions, i.e. the population of values that is used to calculate z-scores are the values for a particular attribute across a set of restaurants under consideration.

<table>
<thead>
<tr>
<th>User</th>
<th>Z-value</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td>1.5</td>
<td>Among the selected restaurants, the following offer exceptional overall value. Komodo has very good service.</td>
</tr>
<tr>
<td>VM</td>
<td>0.7</td>
<td>Among the selected restaurants, the following offer exceptional overall value. Komodo has very good service and good decor.</td>
</tr>
<tr>
<td>VM</td>
<td>0.3</td>
<td>Among the selected restaurants, the following offer exceptional overall value. Komodo's price is 29 dollars. It has very good food quality, very good service and good decor. Takahachi's price is 27 dollars. It has very good food quality, good service and decent decor.</td>
</tr>
<tr>
<td>VM</td>
<td>-0.5</td>
<td>Among the selected restaurants, the following offer exceptional overall value. Komodo's price is 29 dollars. It has very good food quality, very good service and good decor. Takahachi's price is 29 dollars. It has very good food quality, good service and decent decor. Japonica's price is 37 dollars. It has excellent food quality, good service and decent decor.</td>
</tr>
<tr>
<td>VM</td>
<td>-0.7</td>
<td>Among the selected restaurants, the following offer exceptional overall value. Komodo's price is 29 dollars. It has very good food quality, very good service and good decor. Takahachi's price is 27 dollars. It has very good food quality, good service and decent decor. Japonica's price is 37 dollars. It has excellent food quality, good service and decent decor. Shab-u-Tatsu's price is 31 dollars. It has very good food quality, good service and decent decor.</td>
</tr>
<tr>
<td>VM</td>
<td>-1.5</td>
<td>Among the selected restaurants, the following offer exceptional overall value. Komodo's price is 29 dollars. It has very good food quality, very good service and good decor. Takahachi's price is 27 dollars. It has very good food quality, good service and decent decor. Japonica's price is 37 dollars. It has excellent food quality, good service and decent decor. Shab-u-Tatsu's price is 31 dollars. It has very good food quality, good service and decent decor. Bond Street's price is 51 dollars. It has excellent food quality, good service and very good decor. Dojo's price is 14 dollars. It has decent food quality, mediocre service and mediocre decor.</td>
</tr>
</tbody>
</table>

Fig. 17. Comparisons for User VM, for the East Village Japanese Task, at Varying Levels of Conciseness.

The speech-plan that SPUR outputs for the VM response when z is 0.3 is in Figure 18. The nucleus (main claim) is the assertion that Komodo and Takahachi are exceptional restaurants for the selected set. The satellites of the speech-plan are evidential support for this assertion, which consists of the information provided about the selected attributes for each restaurant. Contrast relations hold between pairs of assertions about attributes.

### 4.2.4 Summary Speech Plans

The goal of a summary is to provide an overview of the overall utility of the option set, and to describe the dimensions along which elements of that set differ with respect to their attribute values. The aim is to inform users about both the range of choices and the range of reasons for making those choices; the assumption is that a summary first gives an overview and orients the user to information that will be presented later in the dialogue in a comparison or a recommendation, as illustrated in the sample MATCH dialogue in Figure 2.
Fig. 18. A Speech-Plan representation for a Comparison for a Japanese restaurant in the East Village for user VM

To produce a summary, SPUR examines the user-selected set of restaurants and determines which attributes have the same real values and which attributes have different real values. Unlike the other strategies, summaries do not use the weighted utilities for calculations because of the potential for inconsistent lexicalizations of similarities and differences. The summary simply states the ways in which the restaurants are similar or different.

(1) For each attribute $a_i$, for each option $R_i$, if attribute's real-values all map to the same lexical value, or if cost difference is less than 10 dollars, then attribute is similar, otherwise it is different.

(2) Select similar attributes ordered by user rank.

(3) Select different attributes ordered by user rank.

Fig. 19. Algorithm for summary generation

Figure 19 shows the algorithm for summary generation. Figure 20 illustrates the effects of user model on summaries when Bond Street, Japonica, Takahachi and Komodo are the selected options. The differences in the realized content are a function of the ranking of the attributes in the user model, as specified for users BA, CK and VM in Figure 8.

Figure 21 shows the speech-plan that SPUR outputs for the summary for user VM in Figure 20. The speech-plan consists of two assertions about the differences in the cost and food quality attributes of the selected set of restaurants. The weak rhetorical relation of joint, which has no semantic content, is the only relation that holds between these two assertions.
<table>
<thead>
<tr>
<th>User</th>
<th>Summary Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>CK</td>
<td>The 4 restaurants differ in food quality.</td>
</tr>
<tr>
<td>BA</td>
<td>The 4 restaurants differ in cost, and service.</td>
</tr>
<tr>
<td>VM</td>
<td>The 4 restaurants differ in cost, and food quality.</td>
</tr>
</tbody>
</table>

Fig. 20. Summaries for users CK, BA and VM when Bond Street, Japonica, Takahachi and Komodo are selected with a pen gesture.

<table>
<thead>
<tr>
<th>strategy:</th>
<th>summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>items:</td>
<td>Bond Street, Komodo, Japonica, Takahachi</td>
</tr>
<tr>
<td>relations:</td>
<td>joint(nuc:1, nuc:2)</td>
</tr>
<tr>
<td>content:</td>
<td>1. assert(differ (Komodo's, Takahachi's, Japonica, Bond Street) attr(cost))</td>
</tr>
<tr>
<td></td>
<td>2. assert(differ (Komodo's, Takahachi's, Japonica, Bond Street) attr(foodquality))</td>
</tr>
</tbody>
</table>

Fig. 21. A Speech Plan representation for the Summary for user VM from Figure 20

4.3 Realization of Speech Plans

Because our goal was to test the effectiveness of the content selection algorithms described above, we developed a template-based realizer to generate the system turns, leaving the integration of a sentence-planner and surface realizer for future work. Given the speech-plans as in Figure 13, Figure 18, and Figure 21, the template-based realizer uses principles from argumentation theory as described in Section 1 to generate a marked-up string to be passed to the text-to-speech module.

The template-based realizer first lexicalizes each attribute value of the content assertions for both recommendations and comparisons. Each attribute value except for cost is mapped to a predicative adjective using the following mapping: 0-13 → mediocre; 14-16 → decent; 17-19 → good; 20-22 → very-good; 23-25 → excellent; above 25 → superb. Cost is not lexicalized in this way, because user pilots showed little consensus between users about mapping absolute cost to specific lexical items, i.e. 30 dollars is an expensive meal for some, but cheap for others. Thus the cost attribute is referred to as price and its real value is given in the description.

Figure 13 illustrates a speech-plan for recommendations that the template-based realizer takes as input. Following guidelines from argumentation theory, the strategy for realizing recommendations is to order the claim (nucleus) first followed by the supporting evidence (satellites) [GiuseppeCarenini and Moore, 2001]. In other words, a specific option is first recommended, and then the relevant attribute values are provided as reasons for choosing this option. The satellites are ordered so as to maximize the opportunity for aggregation, in order to produce the most concise recommendations given the content to
be communicated. Examples are given in Figure 6, Figure 11, and Figure 12.

Figure 18 illustrates a speech-plan for comparisons that the template-based realizer takes as input. The strategy for realizing comparisons focused on the goal of communicating both the elaboration and the contrast relations. One way to communicate the elaboration relation between the nuclei and the satellites is to structure the comparison so that all the satellites items about a particular restaurant are grouped together, following the nucleus. In order to communicate the contrast relation, these satellites are produced in a fixed order, with a parallel structure maintained across options [Prevost, 1995, Prince, 1985]. The satellites are initially ordered in terms of their evidential strength, but then are reordered to allow for aggregation in order to produce the most succinct descriptions. Examples are given in Figure 16 and Figure 17.

Figure 21 illustrates a speech-plan for summaries that the template-based realizer takes as input. Since the joint relation is the only rhetorical relation for summaries, the strategy for realizing summaries focuses on ordering the similarities and differences to maximize the opportunity for aggregation. Examples are given in Figure 20.

5 Experimental Evaluation

We wish to explore the effects of tailoring on users’ ratings of system responses, and to determine whether concise tailored responses are more effective than verbose responses. However, because our system is multi-modal, we also want to test potential interactions between the mode in which the response is delivered, i.e. speech or text, and user-tailoring. A further factor that we wished to consider is whether users’ knowledge of the domain and familiarity with New York City restaurants affected their interaction with the system Brennan [1990], Schober [1998]. Our specific hypotheses are that: (a) Users will rate user-tailored responses more highly, whether these responses are generated in speech or in text; (b) Users will rate user-tailored responses in text more highly than user-tailored responses in speech; (c) Users will rate user-tailored responses generated in speech more highly than non-tailored responses generated in text in the same situation; (d) Users who are familiar with the domain will rate speech responses more highly than users who are not familiar with it; (e) Users will rate concise responses more highly than verbose responses.

The experimental procedure we followed for both experiments is based on the idea of treating the subject as an “overhearer” of a series of dialogues, each involving one restaurant-selection task [Walker et al., 2001a, Whittaker and Walker, 2002]. This is possible because the dialogue logging facility in MATCH supports the playback of each dialogue [Ehlen et al., 2002]. The motivation
for the "overhearer" method for testing generation algorithms is that it allows
users to give specific feedback about each system response in the context in
which it was originally provided. See [Whittaker and Walker, 2002] for further
discussion of the method.

The user models for the subjects in both experiments were collected in a
separate process that took place before the experiments. User model elicitation
was done over the web, as described above, using the SMARTER method. This
is done as part of enrolling a subject as a user of the dialogue system.

In order to examine the effect of familiarity with the domain, we asked the
subjects in both experiments to provide information about where they live,
the frequency of eating in restaurants in general, and their familiarity with
Manhattan.

We test the first four hypotheses in the tailoring experiment described in Sec-
tions 5.1 and 5.2. We test the fifth hypothesis with a separate experiment that
directly manipulates the conciseness parameter while holding the user model
constant for the particular user who is acting as subject in the experiment.
This experiment is described in Sections 5.3 and 5.4.

5.1 Experimental Design for Tailoring Experiment

There are four tasks in this experiment, each involving one or two constraints
on the selection of a set of restaurant options: (a) French restaurants; (b)
restaurants in Midtown West; (c) Italian restaurants in the West Village;
(d) Asian restaurants in the Upper West Side. The tasks were chosen after
extensive piloting to accommodate a variety of user models, to be fairly easy
for subjects to remember, and to provide sets of restaurants large enough to be
interesting. The order of the presentation of tasks is consistent across subjects.

Each dialogue exchange between the user and the system is presented on a
separate web page. The initial web page for each task sets up the task by
showing the MATCH system's graphical response for an initial user query,
e.g. Show Italian restaurants in the West Village. Then the following pages
show the user circling some subset of the restaurants and asking the system to
summarize, then compare and finally recommend options from the circled sub-
set. We wanted the task dialogues to follow a typical dialogue structure. Prior
research indicated that one characteristic pattern was for users first to ask
for an overview of possible restaurants (summary), then to identify promising
candidate restaurants and request more information about these (compare)
and finally request detailed information about a single specific option (re-
ommend) [Whittaker et al., 2002b]. All dialogues therefore conformed to a
standard style of presentation in which the three strategies were presented in
In order to test whether users rate user-tailored responses more highly (hypothesis a), the subject sees a system response tailored to her user model, and a different response tailored to the user model of another randomly selected subject. We then compare subjects’ judgments of the two responses. Remember that above we defined a measure of distance between two user models. By randomly selecting another user model (Random), not only can we test whether having one’s own model is better than someone else’s, but we can also quantify how much distance there has to be between two user models to make a difference in the subject’s perception of system responses. The order of the tasks, and the order of appearance of strategies within the task is consistent across subjects, for the reasons given above. However, the order of presentation of subject-tailored and other-tailored responses is randomized from page to page.

For each instance of a recommend, summary, or compare, the subject is asked to state her degree of agreement (on a 5-point Likert scale) with the following statement, intended to determine the informativeness, or information quality, of the response: The system’s utterance is easy to understand and it provides exactly the information I am interested in when choosing a restaurant. The question refers to both comprehensibility and the exact information expressed. We asked about both these dimensions because our algorithms were intended to simultaneously optimize both the exact information presented (the second part of the question), but also the format in which it was presented (the first part of the question). This compound question was arrived at after extensive user piloting.

Since the algorithms for recommendations and comparisons consist first of algorithms for ranking restaurant options, and then for selecting content, we ask users to provide judgements related to the ranking of options produced by the algorithms as a secondary measurement of the efficacy of system responses. For each instance of a recommendation, the subject is asked to state her degree of agreement with this statement: I am confident that the recommended restaurant is someplace I would like to go. A similar statement is used to evaluate the ranking of options for comparisons between three or more restaurants: I am confident that the restaurants being described are places I would like to go. User responses to these questions are measured as the variable RankingConfidence in the results below.

---

2 We did not directly compare differences between tailoring and utterances produced without any user model at all, firstly because prior research had shown that tailored responses are superior to those generated without a model [Carenini and Moore, 2000b] and secondly because we were interested in quantifying differences between models, which our distance metric enabled us to manipulate directly.
In order to examine differences in ratings related to mode of presentation (hypotheses b and c), the entire sequence of web pages is presented twice. The first time through, the subject can only read (not hear) the system responses. The second time, she can only hear them. We used this read-then-hear approach (again after extensive piloting), to obtain subject ratings that are not biased by the generally low user performance ratings for TTS. Previous research has shown that one reason why TTS outputs are rated negatively is their failure to correctly pronounce proper names because these often have non-standard pronunciation. Prior textual presentation means that proper names are primed for users in the speech condition making them less likely to be misunderstood. But the fact that text and speech presentations of the same task are 10-15 minutes apart means that users cannot remember their judgments for the previous instance of the task.

We test the second hypothesis by asking users to judge the same responses in the same context in speech and in text. Overall, we expect text to be rated more highly than speech for several reasons. First, as mentioned above, there is a greater cognitive load inherent in processing speech output for complex information [Whittaker et al., 1991]. In addition, despite the fact that AT&Ts Natural Voices is the most highly evaluated commercial TTS system available, users still indicate limitations with this technology [Beutnagel et al., 1999].

We test the third hypothesis by comparing judgements for user-tailored responses in speech with non-tailored responses in text. We expect that providing a user model will have greater effects for speech than text because of the greater problems that users experience in processing complex speech outputs.

To summarize, each subject “overhears” a sequence of four dialogues about different restaurant-selection tasks. The entire sequence is presented twice (once for text, once for speech). The subject makes six information quality judgments for each dialogue each time. The total number of information quality judgments per subject is 48. The subject makes four ranking confidence judgements for each dialogue each time. The total number of confidence judgements per subject is 32. The total time required to complete the experiment is approximately half an hour per subject.

Sixteen subjects who had previously enrolled with the system took part in the experiment as volunteers. All were fluent English speakers. Most eat out moderately often (seven eat out 3-5 times per month, six 6-10 times). All sixteen currently live in northern New Jersey. Eleven described themselves as somewhat or quite familiar with Manhattan, while five thought they were not very familiar with it. After the experiment, ten subjects identified themselves as very interested in using a system like MATCH in the future.
5.2 Tailoring Experimental Results

We first tested whether any difference at all in the user model affected subjects’ rankings of the information quality of the system’s responses. A paired t-test confirmed our prediction that people perform better with their own model than with a randomly assigned model (t(511)=1.63, p=0.05, for a one-tailed test).

However, as noted before, although the predicted effect is significant, this is a conservative test: the Random model condition includes cases where the randomly assigned model is close to the User’s Own model. We therefore filtered the original set of judgments to exclude cases where the distance between the Random Model and the User’s Own Model was less than 0.3, to exclude these similar cases. This removed 9% of judgments from the original data set. To test the hypotheses, we conducted an analysis of variance with Model Type (Own,Random) * Mode (Speech,Text) * Strategy (Recommend, Compare, Summary) * Familiarity (Not, Somewhat, Quite) as independent variables and Judgments of Information Quality or Ranking Confidence as the dependent variable.

As predicted, there were main effects for Model Type, both for judgements of Information Quality (F=5.15, df = 1,910, p < 0.02) and for judgements of Ranking Confidence (F=4.8, df =1,440 p < 0.03), showing that using the User’s Own Model significantly improved judgments of the system response quality.

For Information Quality, as predicted, Mode was significant, (F=8.18, df = 1,910, p < 0.01), with Text responses being rated more highly than Speech. This showed that despite our attempts to improve the perceived quality of speech responses by priming and previous exposure, this was still not sufficient to elevate judgments of speech to those of text. Mode has no significant effect on users’ Ranking Confidence, (F=0.24, df = 1 p = 0.6).

There were also differences between the different Strategies for both Information Quality (F=188.96, df = 2,910, p < 0.0001), and Ranking Confidence (F=6.182, df = 1,440, p < 0.01). For Information Quality the Recommend strategy was judged better than Compare which in turn is better than Summary. An exploration of this difference shows that the Summary strategy was clearly less effective than the other strategies (the mean score for summaries was 2.33; that for comparisons was 3.53; and that for recommendations was 4.08). User comments at the end of the experiment also qualitatively confirm this finding; many users commented that the summaries were too high-level to be useful. Ranking Confidence judgements were not elicited for summaries, and here the Recommend strategy was also judged better than Compare.
Familiarity also had a main effect on Ranking Confidence. Users who were less familiar with Manhattan had greater confidence that the system was selecting suitable restaurants to recommend and compare (F=7.59, df = 2,440, p < 0.001). This may follow from the fact that these users have a greater need for a system that can make recommendations and comparisons.

Finally, and contrary to our predictions, there was no interaction between Model Type and Mode (F=0.02, df = 1,910, p > 0.05). Recall that we expected that the additional difficulty of remembering complex spoken information would lead users to especially prefer responses generated using their Own Model in the Speech condition, given that these are explicitly tailored to their needs. The absence of the interaction term means that this prediction was not confirmed. There were also no 3 way interactions between Model Type, Mode and Strategy (F=0.09, df = 3,910, p > 0.05). Why were there no interaction effects? One possibility is that there were floor effects for Speech judgments and this in turn reduced the variance in these judgments.

Nevertheless, as we predicted, there were no differences between judgments of text with the random model and tailored speech (t(459)=0.33,p > 0.05). This indicates that with previous exposure to restaurant names and proper name priming it is possible to overcome limitations of speech by the use of tailoring.

This result is also indirectly supported by our findings about the interaction between knowledge and mode. As predicted we also found that people who are familiar with Manhattan (and hence more likely to be know its restaurants) rate speech judgments higher. An ANOVA using Mode Type (Speech versus Text), Strategy (Recommend, Compare, Summary) and Familiarity (Unfamiliar, Somewhat Familiar, Very Familiar) as independent variables, and Judgments as dependent variable, showed marginally higher preferences for Speech Information presumably because of familiarity with the information being presented (F=2.41, df = 3,906, p < 0.10).

More evidence for priming comes from our finding similar interactions between the frequency that users went out, and their preference for spoken responses. In another ANOVA using Mode Type (Speech versus Text), Strategy (Recommend, Compare, Summary) and FrequencyGoingOut (Frequently, Often, Rarely, Never) as independent variables, and Judgments as dependent variable, we found an interaction between mode and frequency of going out (F=1.86, df = 3,906, p = 0.05). Again we might expect that people who go out frequently to be more familiar with restaurants and restaurant selection, making it easier for them to process spoken responses.
Figure 6 and Figure 12, showed recommendations generated by MATCH of varying levels of conciseness. The goal of this experiment is to: (1) Test whether our manipulations of conciseness correspond to user’s perceptions of conciseness; and (2) Determine an optimal level of conciseness for recommendations and comparisons.

The experimental procedure is the “overhearer method” as used in the tailoring experiment. There are two additional tasks in this experiment for a total of six tasks. We use the four tasks from the tailoring experiment: (a) French restaurants; (b) restaurants in Midtown West; (c) Italian restaurants in the West Village; and (d) Asian restaurants in the Upper West Side. The two additional tasks are: (e) cheap restaurants; (f) Japanese restaurants in the East Village. Again, the tasks were also chosen after extensive piloting to accommodate a variety of user models, to be fairly easy for subjects to remember, and to provide sets of potential restaurants large enough to be interesting.

As before, each web page set up the task by showing the MATCH system’s graphical response for an initial user query, e.g. Show Japanese restaurants in the East Village. Then the page showed the user circling some subset of the restaurants and asking the system to compare or recommend options from the circled subset.

Subjects saw one page each for recommend, and compare, for each task. On each page, they saw multiple system responses of differing conciseness. We operationalized concise responses as a z-value of 0.3, sufficient responses as a z-value of -0.7, and verbose responses as a z-value of -1.5. The order of the tasks, and the order of appearance of strategies within the task was consistent across subjects. However, the order of presentation of conciseness variants was randomized from page to page. For each instance of a recommend, or compare, the subject was asked to state her degree of agreement (on a 5-point Likert scale) with the following statement, intended to determine the conciseness of the response: When choosing a restaurant, the amount of information provided by the system utterance is; (1) far too little, (2) too little, (3) neither too little nor too much, (4) too much, (5) far too much.

To summarize, each subject “overheard” a sequence of 6 dialogues about 6 different restaurant-selection tasks, with each dialogue consisting first a comparison, then a recommendation. The subject made 6 information quality judgments per task. The total number of information quality judgments per

---

3 We did not test summary responses since our previous experiment indicated that the summary responses overall received low evaluations from subjects.
subject was 36. The total time required to complete the experiment was approximately half an hour per subject.

Twenty one subjects completed the experiment. All were fluent English speakers. We again collected demographic information, about the frequency they ate out, and their familiarity with Manhattan, in case these affect their judgments. Most eat out moderately often (eleven eat out 3-5 times per month, ten 6-10 times). All subjects currently live in northern New Jersey or in Manhattan. Fourteen described themselves as somewhat or quite familiar with Manhattan, while seven thought they were not very familiar with it. After the experiment, 16 subjects (76%) identified themselves as agreeing with the statement that they would like to use a system like MATCH in the future.

5.4 Conciseness Experimental Results

We tested two major hypotheses. Our first hypothesis addressed user’s sensitivity to conciseness and the correspondence between algorithmic conciseness and user judgments of conciseness. Our expectation was users would discriminate between different descriptions in terms of conciseness. More specifically, we expected that outputs we had operationalized as concise should be judged as providing too little information, outputs operationalized as sufficient should be judged as providing the right amount of information, and outputs operationalized as verbose should be judged as providing too much information.

A second hypothesis concerned the relation between conciseness and information provision strategy. Contrast the recommendations in Figure 12 with the comparisons in Figure 17 for varying values of $z$. Of the two strategies, comparisons inherently contain more information than recommendations, because they mention multiple options and their attributes. We should therefore expect users to judge comparisons as more verbose.

We analyzed the user data using ANOVA. Independent measures were Algorithmic Verbosity (Verbose, Sufficient, Concise), Strategy (Recommend, Compare). We first transformed the elicited user judgments on a linear scale, so that an output judged to provide far too little information was scored -2, too little -1, neither too much nor too little 0, too much +1, and far too much +2. The transformed measure of conciseness was used as the ANOVA dependent measure.

Figure 22 indicates the relationship between Algorithmic conciseness and user judgments. It shows both that users are sensitive to conciseness and that user judgments paralleled our algorithmic implementation. Consistent with our hypothesis, outputs generated as concise were more likely to be judged as having too little information than those generated to be sufficient, which in
turn were likely to have less information than those generated to be verbose (F(2,750)= 220.8, p < 0.0001), with post hoc tests showing judged differences between Algorithmically Concise and Sufficient, and between Algorithmically Sufficient and Verbose (both p < 0.0001). These data show that we have algorithmic control over conciseness.

![Graph showing relationship between Algorithmic Conciseness and User Evaluations.](image)

**Fig. 22.** Relationship between Algorithmic Conciseness and User Evaluations.

Nevertheless, Figure 22 also indicates the need for further calibration of the algorithm. Sufficient outputs require little further calibration as they are judged at -0.01 (where “0” indicates exactly the right amount of information), but those generated to be verbose are judged as 0.78, and those generated to be concise are generated as -0.02. These observations suggest that we may be providing marginally too much information for our concise outputs and too little for our verbose outputs. This would imply a need to tune the algorithm, in particular by adding more information to the sufficient statements.

Our second hypothesis concerned the relationship between judged conciseness and strategy. Figure 23 shows as predicted that recommendations are judged to be more concise than comparisons (F(1,750)= 19.7, p<0.0001). Furthermore, there is an interaction between strategy and judgments (F(2,750)= 10.0, p<0.0001), with the main difference being accounted for by users’ tendency to judge verbose comparisons as containing more information than verbose recommendations (post hoc test, p < 0.05). Possibly this was because verbose comparisons mention as many as 10 facts, and this is perceived to be a large additional memory burden.

Finally, despite our tailoring of information content to individual users’ preferences, it was clear that there were differences between users in terms of their overall perception of conciseness. Figure 8 shows users’ judgments of overall presentation conciseness for each level of algorithmic conciseness. There are
large individual differences between users. While most users judged that presentations provided slightly too little information overall, there was large individual variability, with some users judging presentations provided too little information (overall mean = -1.4), and others judging that presentation provided sufficient information (overall mean = 1.5). This suggests that the level of conciseness should also be tailored to individual preferences.
Previous work on user modeling has applied models of user expertise or knowledge to the generation of tailored texts [Cawsey, 1993, Paris, 1988, Chin, 1989] inter alia. The idea of tailoring recommendations to a user's preferences goes back to Rich 1979, who developed a system that tailored book recommendations to a user's preferences as expressed in a user model. The system first asked the user a series of (yes/no) questions in order to categorize the user into one of its known stereotypes, and adjusted this model as the (typewritten) interaction progressed. The language understanding capabilities of this system were extremely limited, and the "conversation" was a typed interaction. Since then, there has been considerable research on developing systems that utilize models of users' preferences or biases for recommendation or explanation, and on methods for automatically inferring such models from user actions [Morik, 1989, Klein, 1994, Carenini and Moore, 2000b, Giuseppe Carenini and Moore, 2001, Jameson et al., 1995].

In addition, there has been a long history of research on multi-modal interfaces. Early efforts focused on multi-modal input allowing users to combine spoken input with deictic gestures [Bolt, 1980]. Theoretical work provided analyses of the strengths and weaknesses between different modes, especially between graphical and linguistic modalities [Walker, 1989, Cohen et al., 1989, Whittaker and Walker, 1991]. This was followed by the development of various novel systems that exploited modal synergies. These systems were designed to use the strengths of one mode to compensate for the weaknesses of another. For example, input problems in one mode could be finessed by corrections produced in another mode [Bangalore and Johnston, 2000, Oviatt, 1999a, Wu et al., 1999]. A similar synergistic approach has been applied to multi-modal retrieval, where the strengths of visualization in providing global overviews are combined with the search capabilities of text [Leck et al., 2000, Hauptmann and Witbrock, 1996, Whittaker et al., 2002a, Hearst, 1997]. Extensive research has also been conducted on multi-modal generation [Feiner and McKeeown, 1990, McKeown et al., 1998, Mittal et al., 1995, Wahlster et al., 1991]. This work provides principles for governing the allocation of information to different modes, again premised on the notion that certain concepts or processes are more naturally expressed linguistically or visually. See [André, 2002] for an overview of work in this area, and [Oviatt, 1999b] for a detailed theoretical analysis of multi-modal concepts and hypotheses.

To our knowledge, MATCH is the first multi-modal dialogue system to utilize user models for producing speech output. Our work is a direct extension of two lines of previous research. Walker [1996] describes speech-plans for dialogue that (1) use decision theory to rank the options under consideration; (2) motivate an option's acceptance by including content expressing its utility in
proposals for an option. Walker’s PROPOSE-EXPLICIT-WARRANT strategy is similar to our RECOMMEND strategy.

We also build directly on Carenini and Moore’s text generation algorithms for producing tailored and concise arguments in a multi-modal presentation system in the real-estate domain. Carenini and Moore use multi-attribute decision theory to build user models in the real-estate domain. In their system, the user’s task is to select four houses that they wish to view and put them into a “hot list”. Users interact with a multi-modal graphical system which includes: (1) a map showing the location of available houses; (2) sliders, charts and tables displaying various attributes of houses under consideration (e.g., style, overall size, number of rooms). Once the user has made their selection, the system advises them of a new option that has just come on the market, and presents an evaluative recommendation of the house in text. The user is then asked if they wish to change their hot list. They may do as much data exploration as they like before submitting a final hot list. The new option is automatically created by the system to have a utility value that is between the values of the two top ranked houses on the user’s list, according the the user model. By analysing whether users change their hot list (as predicted by the model) and how much data exploration they do, Carenini and Moore can evaluate the effectiveness (or persuasiveness) of the system produced recommendations. The user model is also used to make these recommendations concise, in a similar approach to that described here. Carenini and Moore showed experimentally that tailored descriptions were preferred over non-tailored descriptions, and that concise descriptions based on the user models, were preferred over verbose descriptions [Carenini and Moore, 2000b, GiuseppeCarenini and Moore, 2001, Carenini and Moore, 2000a, Carenini, 2000].

Generation of textual recommendations based on explicitly elicited user preferences is now being commercially deployed by CoGenTex in their Recommender system, which automatically generates natural language descriptions and comparisons of product features, using information obtained from a ranking and comparison engine. (See http://www.cogentex.com/solutions/recommender/index.shtml).

In this work, we have made several advances over the work carried out by Carenini and Moore. In MATCH, we develop user models and dialogue strategies for a new domain, that of restaurant selection in New York City, which allows us to test whether user models based on multi-attribute decision theory are also effective in the restaurant domain. In addition, our utterances are presented as part of an ongoing multi-modal dialogue, where the requirements for information presentation are different than those for presenting text. Note that although the system developed by Carenini and Moore is interactive, it does not carry on a natural language dialogue with the user; it presents a single textual recommendation in a multi-media context. Furthermore, in
addition to dialogue strategies for recommendations, we implement and test
dialogue strategies for comparisons and summaries, using the same underlying
models. Finally, we explore the differences between presenting these strategies
to the user in text or in speech, and the interaction of spoken information
presentation and the user’s cognitive capacity.

Considerable previous research has also been devoted to generating effective
summaries Sparck-Jones [1993, 1999], McKeown and Radev [1995], Rau et al.
[1994], Kupiec et al. [1995], although it has been very difficult to demonstrate
qualitative improvements. Our tailoring experiment shows that our summary
strategy is ineffective; we hope to investigate whether they could be be improved using algorithms described in [Polifroni et al., 2003].

While our work uses directly elicited models, other systems for adaptive pre-
sentation collect user information unobtrusively by observing users’ web brows-
ning behavior [Goecks and Shavlik, 2000], mouse clicks and reading time data [Rafter
et al., 2000], or task-specific actions (e.g., driving time, number of turns, user’s
reaction to system-recommended routes in a route-giving domain) [Linden
et al., 1997, Rogers and Fiechter, 1999]. We believe the direct elicitation pro-
cedure is appropriate for our application as it is common for users to enroll
with a spoken dialogue service.

From a cognitive perspective, our goal in this work was to test hypotheses
about the role of adapting to the conversational partner in dialogue inter-
action. There is substantial evidence that humans adapt their interaction to
their conversational partners, whether these partners are other humans or
computational systems [Brennan, 1991, Schober, 1998]. For example, previ-
ous work has shown that experimental manipulations of the speaking rate,
prosodic range, modality of referring, and response delay of a dialogue system
will be mimicked by the human user [Darves and Oviatt, 2002, Coulston et al.,
2002, Bell et al., 2000], just as humans adapt their interaction style to other
humans [McLemore, 1992]. Previous work has also shown that systems that
adapt to human characteristics, such as speaking rate, or personality, lead to
higher feedback ratings from human users Ward and Nakagawa [2002], Nass
et al. [1995]. The largest body of work examines the ways in which humans
adapt to human conversational partners, with evidence that they modify their
lexical and syntactic choice [Levelt and Kelter, 1982, Kempen and Hoekamp,
1987, Brennan, 1996], their referring expressions [Garrod and Anderson, 1987,
Brennan and Clark, 1996] and the selection of content and form for persuas-
ive arguments and negotiation [Mayberry and Golden, 1996, McGuire, 1968,
Luchok and McCroskey, 1978].

37
7 Conclusions and Future Work

This paper describes an approach to user-tailored generation of evaluative responses for multimodal systems that is based on quantitative user models. We address a pressing problem for current systems, namely that information presentation strategies overload users, and do not effectively support them in making decisions about complex options. We present new algorithms for information presentation based on multi-attribute decision theory that focus the presentation on small sets of options and attributes that are significant and salient to the user. These algorithms enable both option and attribute selection for three different *speech-plans*: summaries, recommendations and comparisons. They have been implemented in SPUR, a speech-planner for the MATCH dialogue system.

The requirements of spoken language, and our application domain, required us to extend previous work to support the generation of both summaries and comparisons, in addition to recommendations. One important contribution of this work is the definition of these new speech act types. Furthermore this framework allows parameters of the speech-plans to be highly configurable: by changing the value of $z$, we can experiment with different definitions of the notion of outlier, and hence generate differently concise speech-plans, that highlight and compare different sets of attributes and options.

The experimental results confirm the utility of tailored responses for all strategy types. Users rated responses generated using their Own Model much more highly than those generated with the Random Model, after we had filtered out spurious matches between the models. As we expected, text responses were rated more highly than speech responses, because of the quality of our TTS system. One prevalent problem that users complained about with speech was the system’s inability to pronounce restaurant names, which are often foreign words. Clearly special purpose techniques would be required to address this highly specific problem.

Contrary to our expectations, we found that the effect of the user model was no greater in the speech than the text condition. However this may have been due to the fact that overall ratings of responses were low in the speech condition. And bearing in mind the TTS pronunciation problems just described, it may be that effects would have been observed in a domain that is less demanding for TTS. In support of this, however, we found that users who ate out more frequently or who knew Manhattan better rated speech responses more highly. This suggests that people who are familiar with restaurant names and general restaurant information are better able to overcome perceptual limitations associated with a TTS that has not been provided with pronunciations for domain specific entities such as foreign restaurant names.
In the future we plan to conduct additional experiments in this framework. First of all, we would like to test the effect of the user model and conciseness parameters on other variables such as task completion, time to completion and user satisfaction, as in other work evaluating spoken dialogue systems [Walker et al., 2002]. Another area of additional experimentation is in the mapping between selected content and dialogue strategy. For example, we did not vary the speech plan structures for each strategy in this experiment, although in our own exploration we identified various possibilities for each strategy. Additional experiments will alter these constraints, and explore subject preferences for the resulting output. In current work, we are enriching SPUR’s ability to structure the selected content, and interfacing SPUR to a sentence planner and surface realizer [Walker et al., 2003, 2001b, Bangalore and Rambow, 2000]. We also hope to conduct field trials of people using the system in a mobile environment.

References


T. Leek, H. Jin, S. Sista, and R. Schwartz. The bbn crosslingual topic detection


Hall, 1998.


