ELSEVIER

Cognitive Science 28 (2004) 811–840

COGNITIVE SCIENCE

http://www.elsevier.com/locate/cogsci

Generation and evaluation of user tailored responses 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] *Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, UK*
[b] *SUNY at Stony Brook, Stony Brook, NY 11794, USA*
[c] *AT&T Labs Research, Florham Park, NJ 07932, USA*
[d] *University of Edinburgh, Edinburgh EH8 9LW, Scotland*

Received 16 July 2003; received in revised form 16 June 2004; accepted 23 June 2004

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 adaptation to the partner takes 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. We focus on one aspect of adaptation, 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.

Keywords: Dialogue systems; User modeling; User-tailored generation

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;

* Corresponding author.
E-mail addresses: walker@dcs.shef.ac.uk, m.a.walker@sheffield.ac.uk (M.A. Walker).

0364-0213/$ – see front matter © 2004 Cognitive Science Society, Inc. All rights reserved.
doi:10.1016/j.cogsci.2004.06.002

Schober, 1998). This tailoring, or adaptation to the partner, has been shown to take place in all facets of human language use, including speaking rate and response delay (Darves & Oviatt, 2002; Ward & Nakagawa, 2002), amplitude and prosodic range (Coulston, Oviatt, & Darves, 2002; McLemore, 1992), lexical and syntactic choice (Brennan, 1996; Kempen & Hoenkamp, 1987; Levelt & Kelter, 1982), choice and modality of referring expressions (Bell, Boye, Gustafson, & Wirn, 2000; Brennan & Clark, 1996; Garrod & Anderson, 1987; Schober, 1998) and in higher level discourse processes such as the selection of content and form for persuasive arguments and negotiation (Joshi, 1982; Joshi, Webber, & Weischedel, 1984; Mayberry & Golden, 1996; McGuire, 1968; Walker, 1996; Webber & Joshi, 1982). This adaptive behavior is based on a *mental model* or a *user model* of the conversational partner (Brennan & Clark, 1996; Levelt, 1989; Wahlster & Kobsa, 1989; Zukerman & Litman, 2001). Such adaptation has been shown to improve listeners' comprehension (Clark & Wilkes-Gibbs, 1986), their satisfaction with an interactive system (Nass, Steuer, & Tauber, 1995), the efficiency with which they execute conversational tasks (Brennan, 1996; Clark & Wilkes-Gibbs, 1986), and the likelihood of achieving higher level goals such as changing the listener's beliefs and attitudes (Luchok & McCroskey, 1978; [Carenini &] Moore, 2000b, 2001; Zukerman & McConachy, 1993).

Our focus here is on one aspect of adaptation, 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. Our research also has the practical goal of improving the dialogue interaction capabilities of the Multimodal Access to City Help (MATCH) multimodal dialogue system, a system that provides information on restaurant and entertainment options in New York City (Johnston et al., 2002b).

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 requirement to produce concise, informative and relevant utterances, especially during the information presentation phase of the dialogue (Walker et al., 2002a). In this phase, the system has a number of possible options that match a user's constraints, which need to be presented 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 them. Dialogue strategies for recommending particular options, or for making balanced comparisons between options, should help users make such evaluations. To be effective in spoken dialogue, these recommendations and comparisons should also be concise.

Previous work on user modeling has primarily applied models of user expertise or knowledge to the generation of user tailored texts, rather than to system utterances in a dialogue system. The first such system, developed by Rich (1979), 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. There has been considerable subsequent research on developing interactive systems that utilize models of users' capabilities, preferences

or biases for recommendation, advice or explanation (Cawsey, 1993; [Carenini &] Moore, 2000b, 2001; Chin, 1989; Jameson, Schafer, Simons, & Weis, 1995; Joshi, 1982; Joshi et al., 1984; Joshi, Webber, & Weischedel, 1986; Klein, 1994; Linden, Hanks, & Lesh, 1997; Morik, 1989; Moore & Paris, 1993; Paris, 1988; Thompson & Goker, 1999; Walker, 1996; Webber & Joshi, 1982) *inter alia*, and on methods for automatically inferring such models from user actions (Goecks & Shavlik, 2000; Linden et al., 1997; Rafter, Bradley, & Smyth, 2000; Rogers & Fiechter, 1999). Generation of text recommendations based on 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 (Cogentex, 2003).

Recommendations for particular options, and comparisons among options, are one form of *evaluative argument*. 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 (Corbett & Connors, 1999; Mayberry & Golden, 1996; McGuire, 1968; Miller & Levine, 1996; Zukerman, McConachy, & Korb, 2000), which are summarized by Carenini and Moore (Carenini & 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 choices are not to mention any opposing evidence, to acknowledge it without refuting it, or to acknowledge it and refute it. The opposing evidence should be presented so as to 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.

These guidelines 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 3–5), 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 and evaluated these guidelines in the context of a system for interactive data exploration in the real estate domain (Carenini, 2000; [Carenini &] Moore, 2000a,b, 2001). Their operationalization of user models is based on multi-attribute decision

theory (Keeney & Raiffa, 1976; Klein, 1994). Multiattribute 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 is then the basis for strategies for selecting and structuring the content of 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 recommendations were preferred over non-tailored recommendations, and that concise recommendations based on the user models were preferred over verbose recommendations.

Our research extends that carried out by Carenini and Moore in four ways. First, we test whether user models based on multi-attribute decision theory generalize across domains, by applying this approach to the problem of restaurant selection in New York City. Second, we extend this approach to multi-modal dialogue, where the requirements for interactive information presentation are different from those for text presentations. The system developed by Carenini and Moore is interactive but does not carry on a natural language dialogue with the user; instead it presents a single text recommendation in a multi-modal context. Third, we extend user-tailored generation to include comparisons as well as recommendations. We evaluate the effects of user-tailoring on these strategies. Finally, we explore the relationship between tailoring and mode of information presentation by exploring the effect of presenting these strategies using text or speech.

Section 2 describes the MATCH system and how we use it 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 argumention 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 sum up in Section 6.

2. The MATCH dialogue system and specific hypotheses

The MATCH system runs on a small, portable, tablet computer, providing a testbed for research on multimodal dialogue interaction in a mobile setting. Fig. 1 shows the size of MATCH relative to the human hand and illustrates a user gesture. 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 Figs. 3 and 4), a table showing options that match the user's current request (if any), buttons to activate the speech recognizer, and a panel that provides feedback on recognition and the system state. The street map is dynamically rendered and can be panned and zoomed. Users may take a turn in the dialog with speech, gestures made with the pen, handwriting, or a multimodal combination of inputs (Johnston et al., 2002b). Inputs are parsed, integrated, and assigned a combined meaning representation using a multimodal language processing architecture based on finite-state techniques (Bangalore & Johnston, 2000; [Johnston] & Bangalore, 2000, 2001). In addition to finite-state multimodal integration, MATCH also differs from previous multimodal systems for interacting with maps, such as QuickSet (Cohen et

Fig. 1. User interacting with MATCH running on Fujitsu PDA.

al., 1998; Johnston, 1998; Johnston et al., 2002a; Oviatt, 1999; Wu, Oviatt, & Cohen, 1999), in that it supports handwritten input in addition to pen gestures, provides a dynamic map display, and incorporates a multimodal dialog manager. The multimodal dialog manager, implemented in Java, is in the style of TrindiKit (Larsson, Bohlin, Bos, & Traum, 1999) and features extensions to support the visual mode in addition to spoken interaction (Hastie, Ehlen, & Johnston, 2002; Johnston et al., 2002b). The system's responses can be speech, changes in the map display or text window, or coordinated multimodal presentations of these different output modes. AT&T's Watson and Natural Voices engines are used for speech recognition and text-to-speech (TTS) (Beutnagel, Conkie, Schroeter, Stylianou, & Syrdal, 1999; Sharp et al., 1997).

U1	*"Show me Italian restaurants in the West Village"*
S1	[zooms to the West Village, displays Italian restaurants. (Figure 3)]
U2	[circles a selection of restaurants and says *"compare"* (Figure 4)]
S2	*"Among the selected restaurants, the following offer exceptional overall value. Uguale's price is $33. It has excellent food quality and good decor. Da Andrea's price is $28. It has very good food quality and good decor. John's Pizzeria's price is $20. It has very good food quality and mediocre decor."*
U3	[writes word "recommend"]
S3	*"Uguale has the best overall value among the selected restaurants. Uguale's price is $33."*

Fig. 2. Example dialogue with MATCH: U, user; S, system.

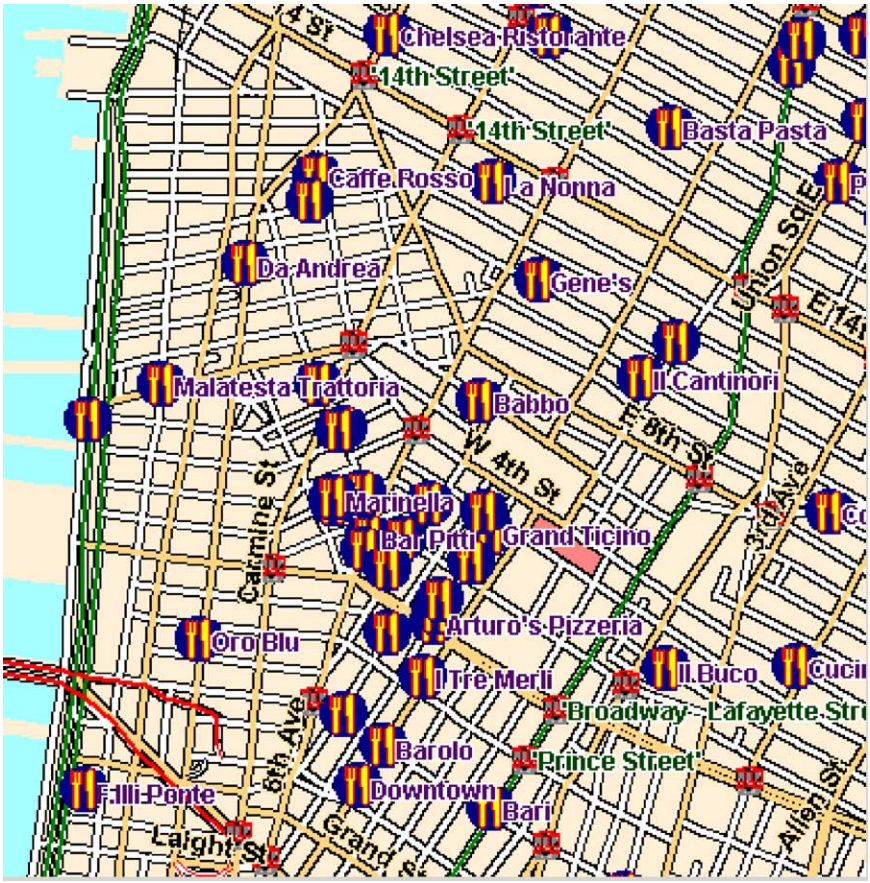

Fig. 3. MATCH's graphical system response to *Show me Italian restaurants in the West Village*.

Our goal is to improve MATCH's dialogue interaction capabilities and general utility by making it easier for users to understand the tradeoffs between different restaurant options. Our view is that this can be done in three ways by providing the system with capabilities for: (1) responding to requests for recommending one of a set of restaurants, or for comparing small sets of restaurants; (2) tailoring these recommendations and comparisons to a model of the user's individual preferences; and (3) making the responses sufficiently concise for the user to understand and remember important information.

The role of the user model in 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 or comparison. Fig. 2 shows a sample dialogue with MATCH exploiting the user tailored dialogue strategies described in the rest of the paper, with examples tailored for the user OR (see Fig. 7 for OR's user model). In Fig. 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 (Fig. 3). At this point, the user has too many options to decide between so he decides to select a set with a

Fig. 4. User circles subset of Italian West Village restaurants for comparison.

pen gesture (Fig. 4) and compare them (U2). S2 is that comparison. Since all the restaurants mentioned in S2 are acceptable, the user asks the system to recommend one by writing the word "recommend" (U3). The recommendation operates on the current dialogue context which is the selected set (from U2).

The system utterances in Fig. 2 are all generated at a fixed level of conciseness. 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 Fig. 5.

User	Conciseness	Output
CK	Concise (z= 0.3)	Bond Street has the best overall value among the selected restaurants. Bond Street has excellent food quality.
BA	Concise (z= 0.3)	Komodo has the best overall value among the selected restaurants. Komodo's price is $29. It's a Japanese, Latin American restaurant.
CK	Sufficient (z= -0.7)	Bond Street has the best overall value among the selected restaurants. Bond Street's price is $51 and it has excellent food quality and good service. It's a Japanese, Sushi restaurant.
BA	Sufficient (z= -0.7)	Komodo has the best overall value among the selected restaurants. Komodo's price is $29 and it has very good service and very good food quality. It's a Japanese, Latin American restaurant.
CK	Verbose (z= -1.5)	Bond Street has the best overall value among the selected restaurants. Bond Street's price is $51 and it has excellent food quality, good service and very good decor. It's a Japanese, Sushi restaurant.
BA	Verbose (z= -1.5)	Komodo has the best overall value among the selected restaurants. Komodo's price is $29 and it has very good service, very good food quality and good decor. It's a Japanese, Latin American restaurant.

Fig. 5. Recommendations for users CK and BA, for the East Village Japanese Task, of varying levels of conciseness.

The user model and the conciseness parameter *z* lead to selection of different restaurants to recommend and to mentioning different facts to each user.

Note also that the user model reflects a users' *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 Fig. 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, with responses tailored to a randomly selected model of another user.

Our second hypothesis concerns conciseness. We utilize the strength of evidence defined by the user model to vary the *conciseness* of system responses. Concise utterances are defined as those mentioning just those restaurants and their attributes that are most relevant to the user's preferences. We compare user's evaluation of concise, sufficient and verbose dialogue responses.

A third hypothesis concerns potential interactions between *user-tailoring and the mode in which information is presented* in a multimodal dialogue system, i.e. in speech or in text. Consistent with prior research, we expect the ephemeral nature of speech (and the resulting cognitive load) to make this a less effective output mode than text (McKeown, Feiner, Dalal, & Chang, 1998; Mittal, Roth, Moore, Mattis, & Carenini, 1995; Oviatt, 1997; Whittaker, Brennan, & Clark, 1991). However, we also expect that tailoring might address some of the inherent limitations of speech, having a greater effect on spoken than text 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 (Jameson et al., 1995; Klein, 1994; Linden et al., 1997; Thompson & Goker, 1999). 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 such user models of this type.

Multi-attribute decision models are based on the claim that if anything is valued, it is valued for multiple reasons (Keeney & 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 these 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 & Hutton 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 and data collection 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* (Whittaker, Walker, & Moore, 2002). These attributes are structured into the one-level tree shown in Fig. 6. 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 (Whittaker et al., 2002).

The structure is user-independent with user-dependent weights on the branches as explained below. We apply this structure 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 0–100. 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 30–100 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

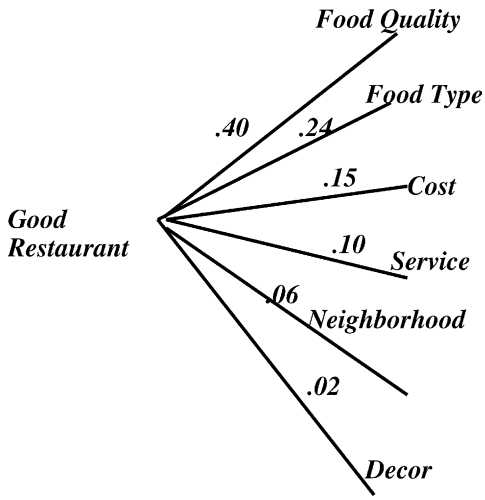

Fig. 6. Structure of objectives for MATCH.

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

Attribute	Range of values	Mapping of values to cardinal utilities
Food quality, Service, Decor	0–30	value × 3 1/3
Cost	0–90	100 − (10/9 × value)
Food type, neighborhood	e.g. Italian, French, West Village	Top values listed by user are mapped to 90, bottom ones to 10 and all others to 50

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.

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 & Hutton 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, u is the function for each attribute mapping attribute values to utilities, and w_k is the weight assigned to each 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, in which users allocate weights directly (Edwards & Hutton Barron, 1994; Srivastava & Connolly, 1995).

We elicit a user model when new users enroll with MATCH using the standard form of questions specified by SMARTER. The elicitation procedure is implemented as a sequence of web pages. The first web page says *Imagine that for whatever reason you have had the horrible luck to have to eat at the worst possible restaurant in the city. The price is $100 per head, you do not 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.

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:

$$w_k = \frac{1}{K} \sum_{i=k}^{K} \frac{1}{i}$$

3.4. Resulting user models

To date, 29 different user models have been elicited and stored in a database that MATCH uses. Fig. 7 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 ranks *service* differently as reflected by the different weights in the Service column. User CK 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 utilize a common measure of *distance*, the Manhattan or city-block distance (Mitchell, 1998), 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,

$$\text{distance}_{ij} = \sum_{k=1}^{K} (|w_{ki} - w_{kj}|)$$

For example, the distance between users CK and VM in Fig. 7 is .84, and the distance between users CK and BA in Fig. 7 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 dialogue 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 user tailored outputs in an interactive dialogue system.

User	FQ	SVC	Dec	Cost	Nbhd	FT	Nbhd Likes	Nbhd Dislikes	FT Likes	FT Dislikes
BA	0.10	0.16	0.06	0.24	0.03	0.41	Downtown, Midtown, E. Village, TriBeCa SoHo	The Bronx, Harlem	Cajun Creole, Greek, Italian, Japanese, Seafood	Coffeehouses Desserts, German, Steak
CK	0.41	0.10	0.03	0.16	0.06	0.24	Midtown, Chinatown, TriBeCa	Harlem, Bronx	Indian, Mexican, Chinese, Japanese, Seafood	Vegetarian, Vietnamese, Korean, Hungarian, German
OR	0.24	0.06	0.16	0.41	0.10	0.03	W. Village, Chelsea, Chinatown, TriBeCa, E. Village	Upper E. Side, Upper W. Side, Uptown, Bronx, Lower Manhattan	French, Japanese, Portugese, Thai, Middle Eastern	no-dislike
MSh	0.41	0.10	0.06	0.24	0.16	0.03	Flatiron, Chelsea, W. Village, Midtown East, Midtown West	Chinatown, Lower E. Side, E. Village, Upper E. Side, Upper W. Side	Indian, Mexican, Ethiopian, Thai, French	Steakhouse, Russian, Korean, Filipino, Diner
VM	0.24	0.10	0.03	0.41	0.06	0.16	Upper W. Side		Cajun Creole, Chinese, Coffeehouses, Indian, Tapas	

Fig. 7. Example user models: FQ, food quality; SVC, service; DEC, decor; Nbhd, neighborhood; FT, food type.

The content planning module in MATCH is called speech planning with utilities for restaurants (SPUR). The user model is used by SPUR for two aspects of content selection: (1) it ranks the options returned from a database query, and the ranking is used by SPUR to select a subset of restaurant options to recommend or compare; (2) it determines a subset of attributes that are mentioned for each option, with the size of the subset depending on the setting of a conciseness parameter.

SPUR takes as input: (1) a dialogue strategy goal; (2) a user model; (3) a conciseness parameter z, and (4) a set of restaurant options returned by the database that match situational constraints specified in the user's query. Given the options, and the conciseness setting, SPUR constructs a content plan specific to each strategy and user model. The resulting content plan serves 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 how the user model affects the recommend and compare content plans.

4.1. The effect of user model on option ranking

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*. Fig. 8 shows how the user models for CK, BA and VM rank these options. The third column gives the overall utility, U_h. 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

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

Fig. 8. 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.

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 eight 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 dialogue strategies

We defined two types of strategy for SPUR: (1) RECOMMEND one of a selected set of restaurants; (2) COMPARE three or more selected restaurants. For each response, SPUR outputs a content plan to the template-based surface realizer (described below), using 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 to compare. 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 Fig. 5. The most concise recommendation for both CK and BA mentions one attribute. The weighted attribute values for each user in Fig. 8 predict how convincing a recommendation would be that includes just that attribute. Fig. 8 indicates that telling CK about Bond Street's *food quality* should provide 34 utils (units of utility) out of a possible 63. Similarly, telling BA about Komodo's *food type* is predicted to provide 36 utils 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 Fig. 5 should provide 46 (34 + 12) utils out of a possible 63 total utils.

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 content plan. Section 4.3 describes the templates used to realize the content plans.

4.2.1. Defining outliers

We 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 options or attribute values. We use the *z-score* (standard value)

(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 content 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 16.

Fig. 9. Algorithm for recommendation generation.

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 σ_V a value v is away from the mean μ_V of a population of values *V*. The population of values *V* that are used to calculate μ_V and σ_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 threshhold for outliers, *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 threshhold 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*, for the user models for VM, CK and BA in Fig. 7.

4.2.2. Recommendation dialogue strategies

The system's strategy for making a recommendation is to select the best option (based on overall utility) and provide convincing reasons for the user to choose it (based on weighted attribute values). Fig. 9 provides the algorithm for selecting the content for the RECOMMEND dialogue strategy.

First, consider the effect of the user model on recommendations at a fixed level of conciseness (*z*-value of .3). Fig. 10 shows sample responses, for the East Village Japanese task, for users CK, BA and VM. Because of differences in user model ranking, Bond Street is recommended to CK and Komodo to BA and VM, and different attributes are selected for the three recommendations.

User	Z value	Output
CK	0.3	Bond Street has the best overall value among the selected restaurants. Bond Street has excellent food quality.
BA	0.3	Komodo has the best overall value among the selected restaurants. Komodo's price is $29. It's a Japanese, Latin American restaurant.
VM	0.3	Komodo has the best overall value among the selected restaurants. Komodo's price is $29 and it has very good food quality.

Fig. 10. Recommendations for users CK, BA and VM, for the East Village Japanese task, for *z* = .3.

User	Z-value	Output
BA	1.5	Komodo has the best overall value among the selected restaurants. Komodo's a Japanese, Latin American restaurant.
BA	0.7	Komodo has the best overall value among the selected restaurants. Komodo's a Japanese, Latin American restaurant.
BA	0.3	Komodo has the best overall value among the selected restaurants. Komodo's price is $29. It's a Japanese, Latin American restaurant.
BA	-0.5	Komodo has the best overall value among the selected restaurants. Komodo's price is $29 and it has very good service. It's a Japanese, Latin American restaurant.
BA	-0.7	Komodo has the best overall value among the selected restaurants. Komodo's price is $29 and it has very good service and very good food quality. It's a Japanese, Latin American restaurant.
BA	-1.5	Komodo has the best overall value among the selected restaurants. Komodo's price is $29 and it has very good service, very good food quality and good decor. It's a Japanese, Latin American restaurant.

Fig. 11. Recommendations for user BA, for the East Village Japanese task, of varying levels of conciseness.

Now, consider the algorithm's implementation for the BA model for varying values of z as illustrated in Fig. 11. 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 at least one outlying attribute, even for the highest value of z. When there are no outlier attributes, the algorithm simply mentions the restaurants with highest overall value. Fig. 8 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. When z is 1.5 or .7, only the *food type* attribute is selected. When z is .3, the attributes *cost* and *food type* are selected. When z is $-.5$, the attributes *cost, service* and *food type* are selected. When z is $-.7$, 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.

4.2.3. Generating comparison content plans

The goal of a comparison is to mention several potential candidate options (those with highest overall utility) and provide the user with user tailored ways for choosing among them (expressed as different weighted attribute values).

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 Fig. 12 and the content for each option is selected using the algorithm in Fig. 13. Because comparisons are inherently contrastive, the algorithm in Fig. 13 describes a procedure whereby if a weighted attribute value is an outlier for any option, the attribute value is realized for all options. We use this approach 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

(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 content 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 content plan that the elements of the set R_i are the top 5 in overall value.

Fig. 12. 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 content 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 content plan with the s_i as joint nuclei.

Fig. 13. Algorithm for selecting content for subset of options to compare.

the realization, which supports the user's inference of contrast (Meteer, 1991; [Prevost, 1995; Prince, 1985]).

Fig. 14 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 .3 selects three options for CK, two for VM and only one for BA. The selected attributes are outliers with respect to the population of attribute values under consideration. For comparisons, the population of values are those for a particular attribute **across** a set of restaurant options.

User	Z value	Output
CK	0.3	Among the selected restaurants, the following offer exceptional overall value. Bond Street's price is $51. It has excellent food quality, good service and very good decor. It's a Japanese, Sushi restaurant. Japonica's price is $37. It has excellent food quality, good service and decent decor. It's a Japanese, Sushi restaurant. Komodo's price is $29. It has very good food quality, very good service and good decor. It's a Japanese, Latin American restaurant.
VM	0.3	Among the selected restaurants, the following offer exceptional overall value. Komodo's price is $29. It has very good food quality, very good service and good decor. Takahachi's price is $27. It has very good food quality, good service and decent decor.
BA	0.3	Among the selected restaurants, the following offer exceptional overall value. Komodo has very good service, very good food quality and good decor.

Fig. 14. Comparisons for users CK, VM and BA, for the East Village Japanese task.

User	Z-value	Output
VM	1.5	Among the selected restaurants, the following offer exceptional overall value. Komodo has very good service.
VM	0.7	Among the selected restaurants, the following offer exceptional overall value. Komodo has very good service and good decor.
VM	0.3	Among the selected restaurants, the following offer exceptional overall value. Komodo's price is $29. It has very good food quality, very good service and good decor. Takahachi's price is $27. It has very good food quality, good service and decent decor.
VM	-0.5	Among the selected restaurants, the following offer exceptional overall value. Komodo's price is $29. It has very good food quality, very good service and good decor. Takahachi's price is $27. It has very good food quality, good service and decent decor. Japonica's price is$37. It has excellent food quality, good service and decent decor
VM	-0.7	Among the selected restaurants, the following offer exceptional overall value. Komodo's price is $29. It has very good food quality, very good service and good decor. Takahachi's price is $27. It has very good food quality, good service and decent decor. Japonica's price is $37. It has excellent food quality, good service and decent decor. Shabu-Tatsu's price is $31. It has very good food quality, good service and decent decor.
VM	-1.5	Among the selected restaurants, the following offer exceptional overall value. Komodo's price is $29. It has very good food quality, very good service and good decor. Takahachi's price is $27. It has very good food quality, good service and decent decor. Japonica's price is $37. It has excellent food quality, good service and decent decor. Shabu-Tatsu's price is $31. It has very good food quality, good service and decent decor. Bond Street's price is $51. It has excellent food quality, good service and very good decor. Dojo's price is $14. It has decent food quality, mediocre service and mediocre decor.

Fig. 15. Comparisons for user VM, for the East Village Japanese task, at varying levels of conciseness.

The outlier attributes here are *food quality, service* and *decor*, which are realized as in Fig. 14. The *food type* attribute is selected for user CK because of the larger set of restaurants from which outliers are calculated.

Now, consider the algorithms in Figs. 12 and 13 applied with z-values ranging from -1.5 to 1.5. Fig. 15 shows comparisons for user VM for varying levels of conciseness. Fig. 8 shows the relevant values for overall utility U_h and weighted attribute values.

4.3. Realization of dialogue strategies

We developed a template-based realizer that takes as input the content plans that SPUR produces using the algorithms described above and generates a marked-up string to be passed to the text-to-speech module.

Fig. 16 illustrates a content plan for recommendations, for user BA for z of $-.7$, that the template-based realizer takes as input. The representation of the plans is based on previous work (Marcu, 1997; Mellish, Knott, Oberlander, & O'Donnell, 1998), where each plan consists of a set of *assertions* that 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 & Thomp-

strategy:	recommend
items:	Komodo, Japonica, Takahachi, Shabu-Tatsu, Bond Street, Dojo
relations:	justify(nuc:1;sat:2); justify(nuc:1;sat:3); justify(nuc:1,sat:4); justify(nuc:1,sat:5)
content:	1. assert(best(Komodo))
	2. assert(has-att(Komodo, cost(29)))
	3. assert(has-att(Komodo, foodquality(verygood)))
	4. assert(has-att(Komodo, service(verygood)))
	5. assert(has-att(Komodo, foodtype(Japanese,Latin American)))

Fig. 16. A content plan representation for a recommendation for user BA for a Japanese restaurant in the East Village for z of $-.7$.

son, 1987). The content plan in Fig. 16 specifies that the nucleus is the assertion that *Komodo has the best overall value*, and that the satellites are the evidential support for this assertion.

Following guidelines from argumentation theory, the strategy for realizing recommendations is to order the nucleus first followed by the satellites. The satellites are ordered to maximize the opportunity for aggregation - to produce the most concise recommendations given the content to be communicated, phrases with identical verbs and subjects are grouped, so that lists and coordination can be used to aggregrate the assertions about the subject. Figs. 5 , 10, and 11 provide examples.

The realizer also 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 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.

Fig. 17 illustrates a content plan for comparisons, for user VM for z of .3, that the template-based realizer takes as input. The option selection algorithm in Fig. 12 determines that Takahachi and Komodo are outliers for overall utility, thus the nucleus is the assertion that Komodo and Takahachi are exceptional restaurants and the satellites are assertions about the selected attributes for each restaurant. Contrast relations hold between pairs of assertions about attributes. The realization template for comparisons focuses on 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 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 Figs. 14 and 15.

strategy:	compare
items:	Komodo, Takahachi, Japonica, Shabu-Tatsu, Bond Street, Dojo
relations:	elaboration(nuc:1, sat:2); elaboration(nuc:1, sat:3); elaboration(nuc:1, sat:4); elaboration(nuc:1, sat:5); elaboration(nuc:1, sat:6); elaboration(nuc:1, sat:7); elaboration(nuc:1, sat:9); elaboration(nuc:1, sat:9); contrast(nuc:2, nuc:3); contrast(nuc:4, nuc:5); contrast(nuc:6, nuc:7); contrast(nuc:8, nuc:9)
content:	1. assert(exceptional(Komodo's, Takahachi's))
	2. assert(has-att(Komodo, cost(29)))
	3. assert(has-att(Takahachi's, cost(27)))
	4. assert(has-att(Komodo, service(verygood)))
	5. assert(has-att(Takahachi's, service(good)))
	6. assert(has-att(Komodo, decor(good)))
	7. assert(has-att(Takahachi's, decor(decent)))
	8. assert(has-att(Komodo, foodquality(verygood)))
	9. assert(has-att(Takahachi's, foodquality(good)))

Fig. 17. A content plan representation for a comparison for a Japanese restaurant in the East Village for user VM for $z = .3$.

5. Experimental evaluation

Our experiments evaluated four main hypotheses, concerning tailoring, conciseness, mode and the interaction between tailoring and mode.

- *Tailoring*: We expected users to prefer tailored to untailored system responses.
- *Mode*: We expected users to prefer text to speech responses.
- *Tailoring/mode*: We expected tailoring to have a greater effect on judgements of speech as opposed to text responses because speech imposes a greater cognitive load.
- *Conciseness*: We expected users to be sensitive to the amount of information provided in system responses, and to prefer concise to verbose responses.

We test the first three hypotheses in the tailoring experiment described in Sections 5.1 and 5.2. We test the final 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.

In both experiments, the user models were collected in a separate process that took place before the experiments. We also carried out a pre-experimental survey where subjects provided information about where they live, the frequency of eating in restaurants in general, and their familiarity with Manhattan.

There were six experimental tasks altogether, 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; (e) cheap restaurants; (f) Japanese restaurants in the East Village. Only tasks a–d were used in the tailoring and mode experiment whereas all six tasks were used in the conciseness experiment. 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.

The experimental procedure for both experiments treats the subject as an "overhearer" of a series of dialogues, each involving one restaurant-selection task (Walker et al., 2001a; Whittaker & Walker, 2002). Each dialogue about a task consists of a sequence of dialogue exchanges between the user and the system, with each exchange 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*. Prior research indicated that a typical dialogue structure in this domain is for users to identify promising candidate restaurants and request more information about these (compare) and then request detailed information about a single specific option (recommend) (Whittaker et al., 2002). Therefore, for all subjects and tasks, the following pages show the user first circling some subset of the restaurants and asking the system to compare them, and then to recommend options from the circled subset. The sample dialogue in Fig. 2 illustrates this dialogue structure. The main advantage of the "overhearer" method is that it allows users to give specific feedback about alternative system responses in the context in which they are provided.

5.1. Experimental design for tailoring and mode experiment

The first experiment tests the tailoring, mode and tailoring/mode hypotheses and consists of dialogues involving tasks a–d, as described above. To test the tailoring hypothesis, the subject sees two types of responses on each web-page for each dialogue exchange, one tailored to her user model, and the other tailored to the user model of another randomly selected subject. We then compare subjects' judgments of the two responses. By randomly selecting another user model (Random), and using the distance between two user models, we can both test whether having one's own model is better than someone else's, and 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 presentation of subject-tailored and other-tailored responses is randomized from page to page.

For each instance of a recommend, or compare strategy, the subject is asked to state her degree of agreement on a five-point Likert scale (a standard technique for mapping subjective responses to scalar values (Likert, 1932) 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 statement refers to both comprehensibility and informativeness. We asked about both these dimensions in a single compound statement because our algorithms were intended to simultaneously optimize both the exact information presented (the second part of the statement), and the format in which it was presented (the first part of the statement). Extensive piloting showed a statement about comprehensibility alone favored a short response and a statement about informativeness alone favored a long response. Responses to this compound statement are measured as InformationQuality.

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 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 (again on a five-point Likert scale): *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.

In order to test the mode and tailoring/mode hypotheses 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 familiarize subjects with proper names in the restaurant domain. Prior text 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 min apart means that users cannot remember their judgments for the previous instance of the task.

We test the mode hypothesis by asking users to judge the same responses in the same context first in text and later in speech. We test the mode/tailoring hypothesis by comparing the effects of tailoring on speech with its effects on 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 InformationQuality judgments for each dialogue each time made up of (a) one recommendation and two comparisons tailored to the subject's user model; and (b) one recommendation and two comparisons tailored to a randomly selected user model. The total number of InformationQuality judgments per subject is 48. The subject makes four RankingConfidence 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.

5.2. Tailoring experimental results

We first tested whether differences in the user model affected subjects' rankings of the InformationQuality of the system's responses. A paired *t*-test confirmed the tailoring hypothesis that people prefer responses generated with their own model than with a randomly assigned model ($t(383) = 1.76$; $P < .05$, for a one-tailed test).

However, 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 .3, to exclude

these similar cases. This removed 9% of judgments from the original data set. To test our hypotheses, we conducted two analyses of variance with model type (Own, Random) × mode (Speech, Text) × strategy (Recommend, Compare) as independent variables and judgments of InformationQuality or RankingConfidence as the dependent variables. As predicted, there were main effects for model type, both for InformationQuality ($F = 5.6$; d.f. = 1,674; $P < .02$) and RankingConfidence ($F = 4.3$; d.f. =1,674; $P < .05$), showing that using the User's Own model significantly improved system responses, and confirming the tailoring hypothesis.

Our results partially confirm the mode hypothesis. For InformationQuality, as predicted, mode was significant, ($F = 3.8$; d.f. = 1,674; $P < .05$), with text responses being rated more highly than speech. Mode has no significant effect on users' RankingConfidence, however ($F = .1$; d.f. = 1,674; $P = .9$).

Finally, and contrary to our predictions, there was no interaction between model type and mode ($F = .02$; d.f. = 1,674; $P > .05$), so the mode/tailoring Hypothesis was not confirmed. One possible explanation is that there were floor effects for Speech judgments and this in turn reduced the variance in these judgments. Neverthless, there were no differences between judgments of text with the random model and tailored speech ($t(397) = 1.8$; $P > .05$). This offers some evidence that with previous exposure to restaurant names and proper name priming it is possible to overcome limitations of speech by the use of tailoring.

5.3. Conciseness experimental design

We now describe our evaluation of the conciseness hypothesis. 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. We first 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 focus was the relation between conciseness and information presentation strategy. Contrast the recommendations in Fig. 11 with the comparisons in Fig. 15 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 than recommendations.

This experiment used all six tasks described above. As before, an initial web page set up the task by showing the MATCH system's graphical response for a user query, and then the page showed the user circling some subset of the restaurants and asking the system to first compare and then recommend options from the circled subset. Subjects saw one page each for recommend and compare, for each task. In this case, on each page, they saw multiple system responses of differing conciseness. We operationalized *concise* responses as a z-value of .3, *sufficient* responses as a z-value of $-.7$, and *verbose* responses as a z-value of -1.5. As before, 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 five-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.*

Twenty-one subjects completed the experiment in approximately half an hour per subject. All were fluent English speakers. Most eat out moderately often (11 eat out 3–5 times per month, 10 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 were not very familiar with it.

5.4. Conciseness experimental results

We analyzed the user data using ANOVA. Independent measures were algorithmic conciseness (verbose, sufficient, concise), and strategy (recommend, compare). Using standard Likert scale procedures we first transformed the elicited conciseness judgments into 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.

Fig. 18 indicates the relationship between algorithmic conciseness and user judgments of conciseness. 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 ($F(2, 750) = 220.8$; $P < .0001$), with post hoc tests showing judged differences between algorithmically concise and sufficient, and between algorithmically sufficient and verbose (both $P < .0001$). These data clearly show that we have algorithmic control over conciseness.

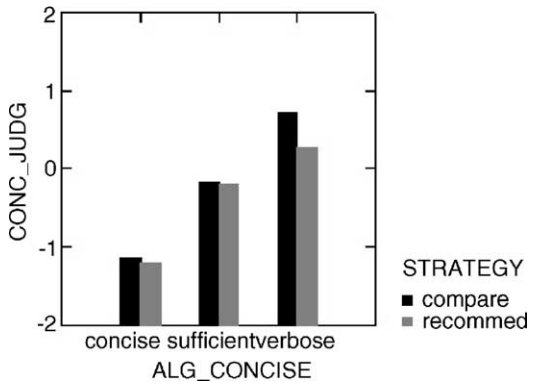

Fig. 18. Relationship between algorithmic conciseness and user evaluations by strategy.

Nevertheless, Fig. 18 also indicates the need for further calibration of the algorithm. If our algorithmic calibration had been correct, we would have expected verbose outputs to have been judged as providing too much information (scored as +1 or greater on the Likert scale), sufficient outputs judged as 0, and concise as −1 or less. Results show that sufficient outputs require little further calibration as they are judged at −.2 (where "0" indicates exactly the right amount of information), but those generated to be verbose are judged as .5, and those generated to be concise are generated as −1.2. 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. Fig. 18 also shows as predicted that recommendations are judged to be more concise than comparisons ($F(1, 750) = 19.7$; $P < .0001$). Furthermore, there is an interaction between strategy and judgments ($F(2, 750) = 10.0$; $P < .0001$), with the main difference being accounted for by users' tendency to judge verbose comparisons as containing more information than verbose recommendations (post hoc test, $P < .05$). 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, there are large individual differences between users in terms of their perception of conciseness, suggesting that the conciseness parameter itself should be user-tailored.

6. Conclusions and future work

This paper describes an approach to user tailored generation of evaluative responses for multimodal dialogue 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 Walker et al. (2002a). 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 two different dialogue strategies: recommendations and comparisons. We have implemented the algorithms for generating content plans for these strategies in SPUR, a content planner for the MATCH dialogue system. Furthermore our theoretical framework allows parameters of the content plans to be highly configurable: allowing us to generate differently concise content plans, that highlight and compare different sets of attributes and options.

Our results show that user models based on multi-attribute decision theory generalize across domains. Techniques that work for other domains are also effective in the restaurant domain, as well as being effective for multi-modal dialogue, where the requirements for interactive information presentation are different from those for text presentations. We have also extended user-tailored generation to include comparisons as well as recommendations. Our results show the effects of user-tailoring on these strategies. Users rated responses generated using their Own Model much more highly than those generated with the Random Model.

We also demonstrated effects of presentation mode. As we expected, text responses were rated more highly than speech responses. Despite our attempts to prime text-to-speech pronunciation, users complained about difficulty understanding restaurant names, which are often foreign words. Contrary to our expectations, we found that the effect of tailoring 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 so that effects may be observed in a domain that is less demanding for text-to-speech. In support of this, 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 understanding text-to-speech output.

We also successfully implemented a strategy for controlling presentation conciseness, and showed that outputs from our algorithm were consistent with user judgments of conciseness, although some fine tuning of the algorithm will be necessary to exactly map onto absolute user judgments.

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., 2002b). Another area of additional experimentation is in the mapping between selected content and dialogue strategy. For example, we did not vary the content plan templates for each strategy in this experiment, although in our own exploration we identified various possibilities for each strategy. Additional experiments could alter different aspects of the template, 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 (Bangalore & Rambow, 2000; Stent, Prasad, & Walker, 2004; Walker, Prasad, & Stent, 2003; Walker, Rambow, & Rogati, 2001b). We also hope to conduct field trials of people using the system in a mobile environment.

Notes

1. Users were allowed to select up to five preferred and five dispreferred food types. 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.

Acknowledgements

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

References

Bangalore, S., & Johnston, M. (2000). Tight coupling of multimodal language processing with speech recognition. In *Proceedings of ICSLP* Beijing, China.

Bangalore, S., & Rambow, O. (2000). Exploiting a probabilistic hierarchical model for Generation. In *COLING* Saarbucken, Germany.

Bell, L., Boye, J., Gustafson, J., & Wirn, M. (2000). Modality convergence in a multimodal dialogue system. In *Proceedings of Gotalog 2000, Fourth Workshop on the Semantics and Pragmatics of Dialogue* Saarbucken, Germany, (pp. 29–34).

Beutnagel, M., Conkie, A., Schroeter, J., Stylianou, Y., & Syrdal, A. (1999). The AT&T Next-Generation Text-to-Speech System. In *Meeting of ASA/EAA/DAGA in Berlin* Germany.

Brennan, S. E. (1991). Conversations with and through computers. Germany, *User Modeling and User-Adapted Interaction*, *1*, 67–86.

Brennan, S. E. (1996). Lexical entrainment in spontaneous dialog. In *International Symposium on Spoken Dialogue* (pp. 41–44).

Brennan, S. E., & Clark, H. H. (1996). Lexical choice and conceptual pacts in conversation. *Journal of Experimental Psychology: Learning, Memory And Cognition*.

Carenini, G., Generating and evaluating evaluative arguments. Ph.D. thesis, University of Pittsburgh, 2000.

Carenini, G., & Moore, J. (2000a). A strategy for generating evaluative arguments. In *Proceedings of the 1st International Conference on Natural Language Generation (INLG-00)* Mitzpe Ramon, Israel.

Carenini, G., & Moore, J. D. (2000b). An empirical study of the influence of argument conciseness on argument effectiveness. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-00)*.

Carenini, G., & Moore, J. D. (2001). An empirical study of the influence of user tailoring on evaluative argument effectiveness. In *IJCAI* Mitzpe Ramon, Israel, (pp. 1307–1314).

Cawsey, A. (1993). Planning interactive explanations. Mitzpe Ramon, Israel, *International Journal of Man–Machine Studies*, *38*, 169–199.

Chin, D. (1989). KNOME: Modeling what the user knows in UC. In A. Kobsa & & W. Wahlster (Eds.), *User Models in Dialog Systems* (pp. 74–107). Springer-Verlag.

Clark, H. H., & Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition*, *22*, 1–39.

Cogentex. Cogentex recommender system, 2003.

Cohen, P. R., Johnston, M., McGee, D., Oviatt, S. L., Pittman, J., Smith, I., et al. (1998). Multimodal interaction for distributed interactive simulation. In M. Maybury & W. Wahlster (Eds.), *Readings in Intelligent Interfaces*. Morgan Kaufmann Publishers.

Corbett, E. P. J., & Connors, R. J. (1999). *Classical Rhetoric for the Modern Student*. Oxford University Press.

Coulston, R., Oviatt, S., & Darves, C. (2002). Amplitude convergence in children's conversational speech with animated personas. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP'2002)* (pp. 2689–2692).

Darves, C., & Oviatt, S. (2002). Adaptation of users' spoken dialogue patterns in a conversational interface. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'2002)*.

Edwards, W., & Hutton Barron, F. (1994). Smarts and smarter: Improved simple methods for multiattribute utility measurement. *Organizational Behavior and Human Decision Processes*, *60*, 306–325.

Garrod, S., & Anderson, A. (1987). Saying what you mean in dialogue: A study in conceptual and semantic coordination. *Cognition*, *27*, 181–218.

Goecks, J., & Shavlik, J. (2000). Learning users' interests by unobtrusively observing their normal behavior. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces* (pp. 129–132).

Hastie, H., Ehlen, P., & Johnston, M. (2002). Context-sensitive multimodal help. In *Proceedings of the 4th International Conference on Multimodal Interfaces*.

Jameson, A., Schafer, R., Simons, J., & Weis, T. (1995). Adaptive provision of evaluation-oriented information: Tasks and techniques. In *IJCAI* (pp. 1886–1895).

Johnston, M. (1998). Multimodal language processing. In *Proceedings of ICSLP* Sydney, Australia.

Johnston, M., & Bangalore, S. (2000). Finite-state multimodal parsing and understanding. In *Proceedings of COLING 2000* Saarbrücken, Germany.

Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A., & Smith, I. (2002). Unification-based Multimodal Integration. In *Annual Meeting of the Association for Computational Linguistics, ACL*.

Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., et al. (2002). MATCH: An architecture for multimodal dialogue systems. In *Annual Meeting of the Association for Computational Linguistics, ACL*.

Johnston, M., & Bangalore, S. (2004). Finite-state multimodal integration and understanding. Saarbrücken, Germany, *Natural Language Engineering*.

Joshi, A. K. (1982). Mutual Beliefs in Question-Answer Systems. In N. V. Smith (Ed.), *Mutual knowledge* (pp. 181–199). Academic Press.

Joshi, A. K., Webber, B., & Weischedel, R. M. (1984). Preventing False Inferences. In *Proceedings of COLING 1984* (pp. 134–138).

A.K. Joshi, B. Webber, R.M. Weischedel, Some aspects of default reasoning in interactive discourse. Technical Report MS-CIS-86–27, University of Pennsylvania, 1986.

Keeney, R., & Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons.

Kempen, G., & Hoenkamp, E. (1987). An incremental procedural grammar for sentence formulation. *Cognitive Science*, *11*, 201–258.

Klein, D. (1994). *Decision Analytic Intelligent Systems: Automated Explanation and Knowledge Acquisition*. Lawrence Erlbaum Associates.

S. Larsson, P. Bohlin, J. Bos, D. Traum, Trindikit manual. Technical report, TRINDI Deliverable D2.2, 1999.

Levelt, W. J. M. (1989). *Speaking: From Intention to Articulation*. MIT Press.

Levelt, W. J., & Kelter, S. (1982). Surface form and memory in question answering. *Cognitive Psychology*, *14*(1), 78–106.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, *140*(1).

Linden, G., Hanks, S., & Lesh, N. (1997). Interactive assessment of user preference models: The automated travel assistant. In *Proceedings of User Modeling '97*.

Luchok, J. A., & McCroskey, J. C. (1978). The effect of quality of evidence on attitude change and source credibility. *The Southern Speech Communication Journal*, *43*(1), 371–383.

Mann, W. C., & Thompson, S. A. (1987). Rhetorical structure theory: Description and construction of text structures. In G. Kempen (Ed.), *Natural Language Generation* (pp. 83–96). Martinus Nijhoff.

Marcu, D. (1997). From local to global coherence: a bottom-up approach to text planning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'97)*.

Mayberry, K. J., & Golden, R. E. (1996). *For Argument's Sake: A Guide to Writing Effective Arguments*. Harper Collins.

McGuire, W. J. The nature of attitudes and attitude change. In: G. Lindzey, E. Aronson (Eds.), The Handbook of Social Psychology, 3:136–314. Addison-Wesley, 1968.

McKeown, K., Feiner, S., Dalal, M., & Chang, S.-P. (1998). Generating multimedia briefings; coordinating language and illustrations. *Artificial Intelligence Journal*, *103*(1), 95–116.

McLemore, C. A. (1992). Prosodic variation across discourse types. In *Workshop on Prosody in Natural Speech*. Institute for Research in Cognitive Science, University of Pennsylvania.

Mellish, C., Knott, A., Oberlander, J., & O'Donnell, M. (1998). Experiments using stochastic search for text planning. In *Proceedings of International Conference on Natural Language Generation* (pp. 97–108).

Meteer, M. (1991). Bridging the generation gap between text planning and linguistic realization. *Computational Intelligence*, *7*(4), 296–304.

Miller, M. D., & Levine, T. R. (1996). Persuasion. In M. B. Salwen & D. W. Stack (Eds.), *An integrated approach to Communication Theory and Research* (pp. 261–276).

Mitchell, T. (1998). *Machine Learning*. MIT Press.

Mittal, Vibhu O., Roth, S., Moore, J. D., Mattis, J., & Carenini, G. (1995). Generating explanatory captions for information graphics. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI '95)* (pp. 1276–1283).

Moore, J. D., & Paris, Cécile L. (1993). Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, *19*(4).

Morik, K. (1989). User models and conversational settings: Modeling the user's wants. In A. Kobsa & W. Wahlster (Eds.), *User Models in Dialog Systems* (pp. 364–385). Springer-Verlag.

Nass, C., Steuer, J., & Tauber, E. (1995). Computers are social actors. In *Proceedings of the Conference on Computer Human Interaction, CHI-94* (pp. 72–78).

Oviatt, S. L. (1997). Multimodal interactive maps: Designing for human performance. In *Human-Computer Interaction* (pp. 93–129).

Oviatt, S. L. (1999). Mutual disambiguation of recognition errors in a multimodal architecture. In *CHI '99* (pp. 576–583). ACM Press.

Paris, C. L. (1988). Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, *14*(3), 64–78.

S. Prevost, A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation. PhD thesis, University of Pennsylvania, 1995.

Prince, E. F. (1985). Fancy syntax and shared knowledge. *Journal of Pragmatics*, *9*(1), 65–81.

Rafter, R., Bradley, K., & Smyth, B. (2000). Personalized retrieval for online recruitment services. In *Proceedings of the 22nd Annual Colloquium on Information Retrieval*.

Rich, E. (1979). User modeling via stereotypes. *Cognitive Science*, *3*(1), 329–354.

Rogers, S., & Fiechter, C. (1999). An adaptive interactive agent for route advice. In *Proceedings of the Third International Conference on Autonomous Agents*.

Schober, M. F. (1998). Different kinds of conversational perspective-taking. In S. R. Fussell & R. J. Kreuz (Eds.), *Social and cognitive psychological approaches to interpersonal communication* (pp. 145–174). Lawrence Erlbaum.

Sharp, R. D., Bocchieri, E., Castillo, C., Parthasarathy, S., Rath, C., Riley, M., et al. (1997). The watson speech recognition engine. In *Proceedings of the ICASSP97* (pp. 4065–4068).

Solomon, M. R. (1998). Consumer Behavior: Buying. In *Having and Being*. Prentice Hall.

Srivastava, J., & Connolly, T. (1995). Do ranks suffice? a comparison of alternative weighting approaches in value elicitation. *Organization Behavior and Human Decision Processes*, *63*(1), 112–116.

Sharp, R. D., Bocchieri, E., Castillo, C., Parthasarathy, S., Rath, C., Riley, M., & Rowland, J. (1997). The Watson speech recognition engine. In *Proceedings of the ICASSP97* (pp. 4065–4068).

Stent, A., Prasad, R., & Walker, M. (2004). Trainable Sentence Planning for Complex Information Presentation in Spoken Dialogue Systems. In *Meeting of the Association for Computational Linguistics*.

Thompson, C. A., & Goker, M. H. (1999). Learning to suggest: The adaptive place advisor. In *Proceedings of the Third International Conference on Autonomous Agents*.

Wahlster, W., & Kobsa, A. (1989). User models in dialogue systems. In *User Models in Dialogue Systems* (pp. 4–34). Springer Verlag.

Walker, M. A. (1996). The Effect of Resource Limits and Task Complexity on Collaborative Planning in Dialogue. *Artificial Intelligence Journal*, *85*(1–2), 181–243.

Walker, M., Aberdeen, J., Boland, J., Bratt, E., Garofolo, J., Hirschman, L., et al. (2001a). DARPA Communicator dialog travel planning systems: The June 2000 data collection. In *EUROSPEECH 2001*.

Walker, M., Rambow, O., & Rogati, M. (2001b). Spot: A trainable sentence planner. In *Proceedings of the North American Meeting of the Association for Computational Linguistics*.

Walker, M., Rudnicky, A., Prasad, R., Aberdeen, J., Bratt, E., Garofolo, J., et al. (2002). DARPA communicator: Cross-system results for the 2001 evaluation. In *ICSLP 2002*.

Walker, M. A., Whittaker, S. J., Stent, A., Maloor, P., Moore, J. D., Johnston, M., et al. (2002). Speech-Plans: Generating evaluative responses in spoken dialogue. In *Proceedings of INLG-02*.

Walker, M., Prasad, R., & Stent, A. (2003). A Trainable Generator for Recommendations in MultiModal Dialog. In *EUROSPEECH*.

Ward, N., & Nakagawa, S. (2002). Automatic user-adaptive speaking rate selection for information delivery. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'2002)*.

Webber, B., & Joshi, A. (1982). Taking the initiative in Natural Language Database Interaction: Justifying Why. In *Proceedings of COLING 1982* (pp. 413–419).

Whittaker, S., Brennan, S., & Clark, H. (1991). Coordinating activity: an analysis of interaction in computer supported cooperative work. In *CHI 91* (p. 1).

Whittaker, S., & Walker, M. (2002). Evaluating dialogue strategies in multimodal dialogue systems. In *IDS02*.

Whittaker, S., Walker, M., & Moore, J. (2002). Fish or fowl: A Wizard of Oz evaluation of dialogue strategies in the restaurant domain. In *Language Resources and Evaluation Conference*.

Wu, L., Oviatt, S. L., & Cohen, P. R. (1999). Multimodal integration – a statistical view. *IEEE Transactions on Multimedia, 1*(4), 334–341.

Zukerman, I., & Litman, D. (2001). Natural language processing and user modeling: Synergies and limitations. *User Modeling and User-Adapted Interaction*, *11*(1–2), 129–158.

Zukerman, I., & McConachy, R. (1993). Generating concise discourses that addresses a user's inferences. In *IJ-CAI'93*.

Zukerman, I., McConachy, R., & Korb, K. (2000). Using argumentation strategies in automated argument generation. In *Proceedings of the 1st International Natural Language Generation Conference* (pp. 55–62).