# Speech-Plans: Generating Evaluative Responses in Spoken Dialogue

**M.A. Walker S. Whittaker A. Stent[†] P. Maloor J.D. Moore[‡] M. Johnston G. Vasireddy**

AT&T Labs - Research
Florham Park, NJ, USA, 07932
{walker,stevew,pmaloor,johnston}@research.att.com

[†]SUNY at Stony Brook
Stony Brook, NY, USA, 11794
stent@cs.sunysb.edu

[‡]University of Edinburgh
Edinburgh, Scotland, EH8 9LW
jmoore@cogsci.ed.ac.uk

## Abstract

Recent work on evaluation of spoken dialogue systems indicates that better algorithms are needed for the presentation of complex information in speech. Current dialogue systems often rely on presenting sets of options and their attributes sequentially. This places a large memory burden on users, who have to remember complex trade-offs between multiple options and their attributes. To address these problems we build on previous work using multiattribute decision theory to devise *speech-planning* algorithms that present user-tailored summaries, comparisons and recommendations that allow users to focus on critical differences between options and their attributes. We discuss the differences between speech and text planning that result from the particular demands of the speech situation.

## 1 Introduction

Recent work on evaluating spoken dialogue systems suggests that the information presentation phase of complex dialogues is often the primary contributor to dialogue duration (Walker et al., 2001). During this phase, the system returns from a database query with a set of options that match the user's constraints (Figure 1). The user then navigates through these options and refines them by offering new constraints. Depending on the number of options returned, and the facilities provided for refinement, this process may be quite onerous, leading to reduced user satisfaction.

| | |
|---|---|
| SYS: | I found 9 round trips from Hartford to Orlando on October 31st and returning Hartford on November 6th. For option 1, I have an outbound flight with U S Air departing at 7 AM and arriving at 9 49 AM non-stop. A return flight is with U S Air departing at 10 30 AM and arriving at 12 28 AM with 1 stopover. Ticket price is 165 dollars. Please say next option or flight details or I'll take it. |
| USER: | NEXT OPTION |
| SYS: | For option 2, I have an outbound flight with U S Air departing at 3 30 PM and arriving at 6 19 PM non-stop. A return flight is with U S Air departing at 8 25 PM and arriving at 12 28 AM with 1 stopover. Ticket price is 165 dollars. Say next option, or, flight details, or, I'll take it. |
| USER: | NEXT OPTION |
| | ..... |
| | Dialogue continues until user selects option |

Figure 1: Information presentation phase of a Communicator dialogue

Even in multimodal dialogue systems, the information presentation phase may still be laborious. For example, consider a user's request to *Show Italian Restaurants in the West Village* in the MATCH (Multimodal Access To City Help) dialogue system (Johnston et al., 2001). Figure 2 shows the large number of highlighted options generated in response. To make an informed choice, the user has to access information about each restaurant by clicking on its icon.

In both types of dialogue system, the complexity of having to sequentially access the set of returned options

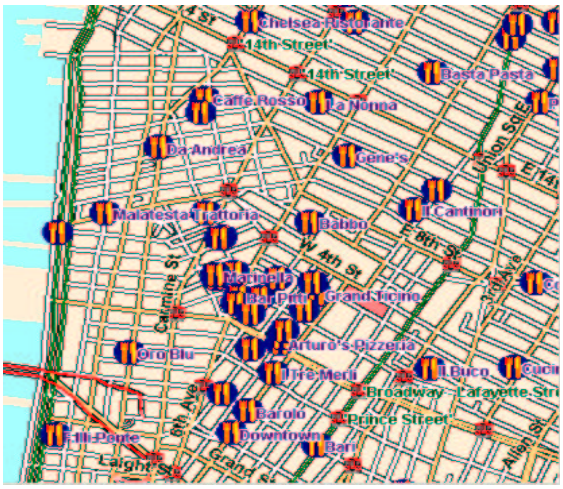


Figure 2: MATCH's graphical system response to *Show me Italian Restaurants in the West Village.*

makes it hard for the user to remember information relevant to making a decision. To reduce user memory load, we need alternative strategies to sequential presentation. In particular we require better algorithms for: (1) selecting the most relevant subset of options to mention, and (2) choosing what to say about them. We need methods to characterize the general properties of returned option sets, and to highlight the options and attributes that are most relevant to choosing between them.

Multi-attribute decision theory provides a detailed account of how models of user preferences can be used in decision making (Keeney and Raiffa, 1976; Edwards and Barron, 1994). By focusing on user preferences we can present information that is both more concise and more tailored to the user's interests. Our work is an extension of two lines of previous research that make direct use of decision theory. Walker (1996) describes dialogue strategies that (1) use decision theory to rank the options under consideration; and (2) include content expressing an option's utility in proposals. Carenini and Moore (2000) apply decision theoretic models of user preferences to the generation of textual evaluative arguments in the real-estate domain.

We extend these techniques to the generation of *speech-plans* intended to address the specific requirements of information presentation for complex spoken or multimodal dialogues. We define multi-attribute decision models of users' preferences and use them to devise speech-plans for SUMMARIZE, RECOMMEND and COMPARE strategies that abstract, highlight and compare small sets of user-relevant options and attributes. These strategies both ensure that the discussed content is relevant to users, and reduce the memory demands involved in making complex decisions with speech or multimodal data. These strategies are embodied in a speech-planning module called SPUR (Speech-Planning with Utilities for Restaurants) for the MATCH system.

In what follows we describe the development of multi-attribute decision models for the restaurant domain and the design and motivation of SPUR, and describe how SPUR is integrated into the MATCH dialogue system.

## 2 Multi-Attribute Decision Models in the Restaurant Domain

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).

The first step of the 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. 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 a one-level tree; 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.

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. In the restaurant database *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.

We next aggregate the vector of $u(x)$ values into a scalar in order to determine the overall utility $U_h$ of each option $h$. Heuristic tests applied to this domain suggest that an additive model is a good approximation (See (Edwards and Barron, 1994)). Thus, if $h$ ($h$ = 1,2, . . . H) is an index identifying the restaurant options being evaluated, $k$ ($k$ = 1,2, . . . K) is an index of the attributes, and $w_k$ is

| User | Food Quality | Service | Decor | Cost | Nbhd | FT | Nbhd Likes | Nbhd Dislikes | FT Likes | FT Dislikes |
|---|---|---|---|---|---|---|---|---|---|---|
| **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 | West Village, Chelsea, Chinatown, TriBeCa, East Village | Upper East Side, Upper West Side, Uptown, Bronx, Lower Manhattan | French, Japanese, Portugese, Thai, Middle Eastern | no-dislike |

Figure 3: Sample User Models (Nbhd = Neighborhood; FT = Food Type)

the weight assigned to each attribute:

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

The final step of decision model construction is the assignment of 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 judgements. SMARTER's main advantage over other elicitation procedures is that it only requires the user to specify the *ranking* of domain attributes. 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:

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

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, 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 foodtype likes or dislikes.

To date, 29 different user models have been elicited and stored in a database that SPUR accesses (see Figure 3, which shows attribute weightings and likes and dislikes for two users). For 25 of these users, we found that *cost* and *food quality* are always in the top three attributes, but their relative importance and that of other attributes, such as *decor, service, neighborhood* and *food type*, varies widely.

The user model reflects a user's *dispositional* biases about restaurant selection. These can be overridden by *situational* constraints specified in a user query. For example, as Figure 3 shows, some users express strong preferences for particular food types, but these can be overridden by simply requesting a different food type. Thus *dispositional* biases never eliminate options from the set of options returned by the database, they simply affect the *ranking* of options.

## 3 The SPUR Speech planner

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. The user model is used by SPUR to rank the options returned from a database query and select the content expressed about each option.

The aim of the speech-plan strategies is to filter the information presented to the user so that only user-relevant options and attributes are mentioned, contrasted and highlighted. We defined three types of strategy 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. Each strategy uses the overall utility $U_h$ to rank the options and the weighted attribute values $w_k u_k(x_{hk})$ to select the content for each option. For each response, SPUR outputs a:

- speech-plan: a semantic representation of the selected content items and the rhetorical relations holding between them.
- Template realization: a marked-up string to be passed to the text-to-speech module;

The template-based realizer lexicalizes each attribute value except for *cost* with 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*.

Carenini and Moore (2001; 2000) 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. Conciseness is highly desirable for speech. Because of the user's memory load, we want to restrict the set of mentioned options and attributes to those that are most important to the user.[1] The weighted utility values for each attribute are a precise prediction of how **convincing** an argument for an option would be that includes the attribute's content. 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 $V$. Depending on a threshhold for $z$, different numbers of options or attribute values are considered to be worth mentioning. In the examples below $z$ is 1.0. 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).

We demonstrate below the differences in option ranking and content selected for three different user models:

- No-Model: Options are in the order that the database returns them, and there is no ranking on the importance of attributes for each option.
- CK: The user model for CK in Figure 3.
- OR: The user model for OR in Figure 3.

The No-Model responses illustrate the default strategy. They are neither *tailored* nor *concise*. We compare the output for the CK and OR models because the different ranking of attributes in these models leads to very different responses. We first illustrate the effect of these user models on option ranking and next present the RECOMMEND, COMPARE and SUMMARY strategies.

### 3.1 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 Italian restaurants in the West Village* in Figures 4 and 5 for the users CK and OR respectively. (We do not give the No Model option here as this consists of a sequential recitation of the full option set presented in alphabetical order.) The first column gives the overall utility. The other columns give the attribute values and weighted utilities (WTD). Note that *food quality* contributes most strongly to the CK model ranking, while *cost* contributes most strongly to the OR model ranking. For example, consider the differences in overall ranking for Babbo and Uguale for CK and OR resulting from different attribute weightings. Babbo is fifth for OR because OR ranks *food quality* second. Babbo's 26 rating for *food quality* results in 36 utils for CK, but only 21 utils for OR. Also, Babbo's price of $60 per person results in only 14 utils for OR; all of the restaurants ranked higher by OR than Babbo's are less expensive. On the other hand, Uguale is more highly ranked for OR than CK. This is mainly because its modest price gets 28 utils for OR but only 11 for CK.

[1] Carenini and Moore use the predicate *notably-compelling* to describe such attributes.

### 3.2 Recommendations

The goal of 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 6 provides the algorithm for the RECOMMEND strategy. Sample responses for the No-Model, CK and OR models are in Figure 7.

Consider the algorithm's application with the OR model (see Figure 5 for relevant values). Uguale is the option with the highest utility ($U_h$ = 69). The weighted attribute values for Uguale are *17,4,9,28,9,2* for *food quality, service, decor, cost, neighborhood* and *food type* respectively. The *z-scores* (used for determining outliers) are 0.57,-0.78,-0.26, 1.72,-0.26,-0.99. Only *cost* is mentioned in the recommendation, because only the *z-score* of *cost* is greater than 1 (our setting for $z$ for these examples). A similar calculation leads to only *food quality* being mentioned for CK. Our algorithm parameters are tunable; we could redefine the value of $z$ to be 0.5 leading less extreme outliers to be mentioned.

As Figure 7 shows, the No-Model recommendation is verbose and complex because there is no method for highlighting relevant attributes. In contrast, the OR and CK recommendations are shorter and clearer. Because the system can identify outlier attributes, only attributes that significantly contribute to an option's utility are mentioned. The OR and CK recommendations differ because different attributes are important for these two users.

1. Select the restaurant option with highest overall utility from returned options.
2. Identify the outliers among all the weighted attribute values for that option, using the threshold we have set for $z$.
3. Lexicalize the real values for the outliers.
4. Generate the recommendation.

Figure 6: Algorithm for recommendation generation

### 3.3 Comparisons

The goal of a comparison is to generate several potential candidate options (those with highest overall utility)

| Name | Utility | Food-Q (WTD) | Service (WTD) | Decor (WTD) | Cost (WTD) | Neighborhood (WTD) | Food Type (WTD) |
|---|---|---|---|---|---|---|---|
| Babbo | 66 | 26(36) | 24(8) | 23(2) | 60(5) | W. Village (3) | Italian (12) |
| Il Mulino | 66 | 27(38) | 23(7) | 20(2) | 65(4) | W. Village (3) | Italian (12) |
| Uguale | 64 | 23(29) | 22(7) | 18(2) | 33(11) | W. Village (3) | French, Italian (12) |
| Da Andrea | 60 | 22(26) | 21(6) | 17(1) | 28(12) | W. Village (3) | Italian (12) |
| John's Pizzeria | 59 | 22(26) | 15(3) | 13(1) | 20(14) | W. Village (3) | Italian, Pizza (12) |
| Vittorio Cucina | 57 | 22(26) | 18(4) | 19(2) | 38(10) | W. Village (3) | Italian (12) |
| Cent'anni | 56 | 22(26) | 20(5) | 15(1) | 45(9) | W. Village (3) | Italian (12) |
| Marinella | 56 | 21(24) | 20(5) | 17(1) | 35(11) | W. Village (3) | Italian (12) |
| Bar Pitti | 53 | 20(22) | 17(4) | 15(1) | 31(11) | W. Village (3) | Italian (12) |
| Grand Ticino | 50 | 19(19) | 19(5) | 17(1) | 39(10) | W. Village (3) | Italian (12) |
| Arlecchino | 48 | 18(17) | 17(4) | 15(1) | 34(11) | W. Village (3) | Italian (12) |

Figure 4: Restaurants ranked by CK model for query *Italian Restaurants in the West Village*; Domain attribute values are given along with WTD = Weighted Utility for that attribute for the CK user model.

| Name | Utility | Food-Q (WTD) | Service (WTD) | Decor (WTD) | Cost (WTD) | Neighborhood (WTD) | Food Type (WTD) |
|---|---|---|---|---|---|---|---|
| Uguale | 70 | 23(17) | 22(4) | 18(9) | 33(28) | W. Village (9) | French, Italian (2) |
| Da Andrea | 69 | 22(16) | 21(4) | 17(8) | 28(31) | W. Village (9) | Italian (1) |
| John's Pizzeria | 68 | 22(16) | 15(2) | 13(5) | 20(35) | W. Village (9) | Italian, Pizza (1) |
| Vittorio Cucina | 64 | 22(16) | 18(3) | 19(9) | 38(26) | W. Village (9) | Italian (1) |
| Babbo | 62 | 26(21) | 24(5) | 23(12) | 60(14) | W. Village (9) | Italian (1) |
| Marinella | 62 | 21(14) | 20(3) | 17(8) | 35(27) | W. Village (9) | Italian (1) |
| Trattoria Spaghetto | 62 | 19(11) | 17(2) | 14(6) | 25(33) | W. Village (9) | Italian (1) |
| Bar Pitti | 61 | 20(13) | 17(2) | 15(7) | 31(29) | W. Village (9) | Italian (1) |
| Cucina Stagionale | 61 | 18(10) | 15(2) | 12(5) | 23(34) | W. Village (9) | Italian (1) |
| Malatesta Trattoria | 60 | 18(10) | 16(2) | 16(7) | 28(31) | W. Village (9) | Italian (1) |
| Cent'anni | 58 | 22(16) | 20(3) | 15(7) | 45(22) | W. Village (9) | Italian (1) |
| Il Mulino | 58 | 27(23) | 23(4) | 20(10) | 65(11) | W. Village (9) | Italian (1) |

Figure 5: Restaurants ranked by OR model for query *Italian Restaurants in the West Village*; Domain attribute values are given along with WTD = Weighted Utility for that attribute for the OR user model.

| User | Recommend Output |
|---|---|
| No Model | *Bar Pitti has the best overall value among the selected restaurants. Bar Pitti's price is 31 dollars. It has very good food quality and good service. It's in the West Village. It's an Italian restaurant.* |
| CK | *Babbo has the best overall value among the selected restaurants. Babbo has superb food quality.* |
| OR | *Uguale has the best overall value among the selected restaurants. Da Andrea's price is 33 dollars.* |

Figure 7: Recommendations for three different user models for a selection of Italian West Village restaurants

and weigh up the different reasons (expressed as different weighted attribute values) for choosing each of them. 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 8. Then the content for each option is selected using the algorithm in Figure 9. Because comparisons are inherently contrastive, the algorithm in Figure 9 describes a procedure whereby if a weighted attribute value is an outlier for any option, the attribute value is realized for all options.

Consider the algorithm in Figure 9 using the CK user model (see Figure 4 for relevant values). The option selection algorithm in Figure 8 determines that Babbo, Il Mulino and Uguale are outliers for overall utility. Then *z-scores* for the weighted attribute values are calculated for each attribute **across** these options. The only attributes whose values show significant variability are *food quality, service* and *cost*. *Food quality* is a negative outlier for Uguale, *service* a positive outlier for Babbo, and *cost* a positive outlier for Uguale. Thus these three attributes are selected for the comparison, and their real values are realized as in Figure 10. A similar calculation for the OR model leads to the realization of the *cost*, *food quality*, and *decor* attribute values.

Figure 10 illustrates the effect of user model on comparisons. The NoModel comparison has no utility model, and hence no method for selecting outlier options or attributes to be mentioned. Thus the first five restaurants options returned are described, along with all their attributes. In consequence the No-Model comparison is lengthy and complex. The OR and CK descriptions are shorter and focus on attributes that are both salient and significant for the specific user.

### 3.4 Summaries

The goal of a summary is to provide an overview of the range of overall utility of the option set. It also describes 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 rea-

- If the number of restaurants is greater than 5 then
  - Output only those restaurants that are positive outliers for overall utility (outstanding restaurants)
  - If there are no outstanding restaurants, output the top 5 in terms of utility value.

Figure 8: Algorithm for selecting a subset of options to compare

1. For each option, for each attribute
   (a) If the weighted attribute value is an outlier when compared against the weighted attribute value for other options, then add attribute to $OUTLIER-LIST.
2. For each attribute in $OUTLIER-LIST,
   (a) Lexicalize real values for the attribute;
   (b) Order attributes to allow for aggregation;
   (c) Generate description.

Figure 9: Algorithm for selecting content for subset of options to compare

| User | Compare Three or More Strategy |
|---|---|
| No Model | *The first five restaurants follow. Bar Pitti's price is 31 dollars. It has very good food quality, good service and decent decor. Arlecchino's price is 34 dollars. It has good food quality, good service and decent decor. Babbo's price is 60 dollars. It has superb food quality, excellent service and excellent decor. Cent'anni's price is 45 dollars. It has very good food quality, very good service and decent decor. Cucina Stagionale's price is 23 dollars. It has good food quality, decent service and mediocre decor.* |
| CK | *Among the selected restaurants, the following offer exceptional overall value. Babbo's price is 60 dollars. It has superb food quality, excellent service and excellent decor. Il Mulino's price is 65 dollars. It has superb food quality, excellent service and very good decor. Uguale's price is 33 dollars. It has excellent food quality, very good service and good decor.* |
| OR | *Among the selected restaurants, the following offer exceptional overall value. Uguale's price is 33 dollars. It has good decor and very good service. It's a French, Italian restaurant. Da Andrea's price is 28 dollars. It has good decor and very good service. It's an Italian restaurant. John's Pizzeria's price is 20 dollars. It has mediocre decor and decent service. It's an Italian, Pizza restaurant.* |

Figure 10: Comparisons for three different user models for a selection of Italian West Village restaurants

sons for making those choices. SPUR therefore examines the user-selected set of restaurants and determines which attributes have the *same* values and which attributes have *different* values. Then it simply states the ways in which the restaurants are similar or different. Figure 11 shows the algorithm for summary generation. Unlike the other strategies, summaries do not use the weighted utilities for calculations because of the potential for inconsistent lexicalizations of similarities and differences. Figure 12 illustrates the effects of user model on summaries when *Da Andrea, Malatesta Trattoria,* and *Uguale* are the selected options.

## 4 Integration of Speech-Plans into MATCH

We have integrated SPUR into the MATCH multimodal speech-enabled dialogue system. MATCH runs standalone on a Fujitsu PDA, providing users with mobile access to information for New York City, and enabling experimentation in realistic mobile settings (Figure 13).

Users interact with a multimodal user interface client which displays a pan-able zoomable dynamic street map of NYC. Users specify inputs via speech, gesture, handwriting or by a combination of these. Outputs are generated in speech, using a graphical display, or with a combination of both these modes. Speech recognition is provided by AT&T's Watson engine, and TTS by AT&T's Natural Voices. MATCH uses a finite-state approach to parse, integrate, and understand multimodal and unimodal inputs (Johnston and Bangalore, 2000). The multimodal dialogue manager (MDM) is based on the notion that each conversational move functions to transform the information state. The state consists of a set of variables, whose bindings are updated as the dialogue progresses.

Figure 14 shows a sample dialogue with MATCH using SPUR. The summary, comparison and recommendation examples are those presented above for OR. The aim is to show these strategies in context in our working system. In U1 the user specifies the query *West Village Italian* in speech. The system responds in S1 by presenting a map of New York, zooming to the West Village and highlighting Italian restaurants. At this point, the user has too many options to decide between and so s/he circles some highlighted restaurants (Figure 15) and says *summarize* (U2). The system then produces a summary (S2), highlighting options and attributes relevant to the user OR. The user decides to select a different set with a gesture (Figure 16) 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

1. For each attribute, for each option,
   - 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. Output similar attributes ordered by user rank.
3. Output different attributes ordered by user rank.

Figure 11: Algorithm for summary generation

| User | Summary Output |
|---|---|
| No Model | The 3 restaurants have the same price range of 28 to 33 dollars. They differ in food quality, service, and decor. |
| CK | The 3 restaurants have the same price range of 28 to 33 dollars. They differ in food quality. |
| OR | The 3 restaurants have the same price range of 28 to 33 dollars. They differ in food quality and decor. |

Figure 12: Summaries for three different user models when *Da Andrea, Malatesta Trattoria,* and *Uguale* are selected.

selected set (from U3). This example shows how SPUR allows users to finesse the problem of having too many complex options to evaluate by presenting compact descriptions, highlighting only those options and attributes that are directly relevant to the user.

## 5 Conclusions and Future Work

This paper describes an approach to user-tailored generation of evaluative responses for spoken dialogue that is based on quantitative user models. We address a pressing problem for current dialogue systems, namely that sequential information presentation strategies overload users, and do not effectively support them in making decisions between 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.

Much prior work that has applied models of user expertise or knowledge to the generation of tailored texts (Cawsey, 1993; Paris, 1988) *inter alia*. Other previous work has utilized models of users' preferences or biases, and methods for automatically inferring such models from user actions (Morik, 1989; Klein, 1994; Carenini and Moore, 2000; Carenini and Moore, 2001; Jameson et al., 1995). Our work differs from these in focusing on the particular demands of speech output, using directly elicited models [2].

Figure 13: MATCH running on Fujitsu PDA

| | |
|---|---|
| U1 | *"show Italian restaurants in the West Village"* |
| S1 | [zooms to the West Village, displays Italian restaurants. (Figure 2] |
| U2 | [circles some highlighted restaurants and says *"summarize"*. (Figure 15)] |
| S2 | *"The 3 restaurants have the same price range of 28 to 33 dollars. They differ in food quality and decor."* |
| U3 | [circles a selection of restaurants and says *"compare"* (Figure 16)] |
| S3 | *"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."* |
| U4 | [writes word "recommend"] |
| S4 | *"Uguale has the best overall value among the selected restaurants. Uguale's price is 28 dollars."* |

Figure 14: Example dialogue with MATCH

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; while (2) motivating 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 the real-estate domain. However, this is the first application of this approach to a speech-planning

[2] This direct elicitation procedure is appropriate for our application as it is common for users to enroll with a spoken dialogue service.

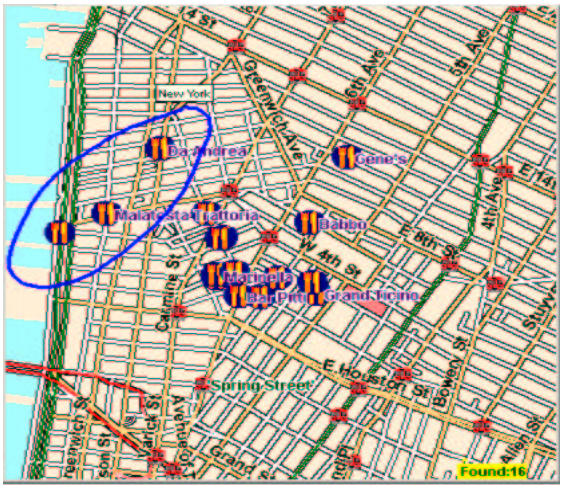


Figure 15: User circles subset for summary using a pen gesture.

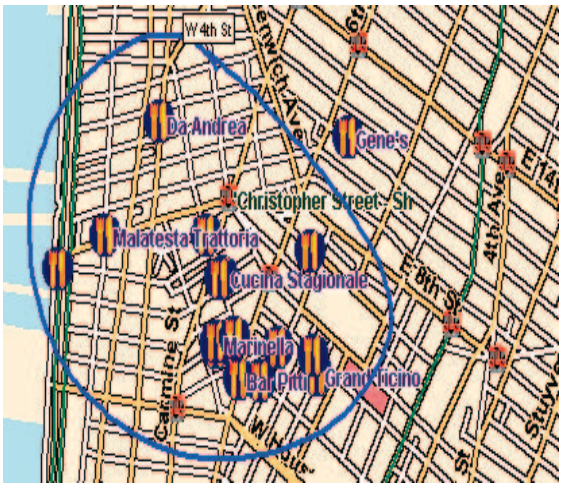


Figure 16: User circles subset for comparison.

module for a dialogue system where the requirements for information presentation are different from when presenting text.

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.

Our initial results suggest that users should be able to find a desirable option more efficiently using SPUR than with existing methods. We are currently conducting a user evaluation of SPUR to test this. We also hope to enrich SPUR's ability to structure the selected content, and to interface SPUR to a sentence planner and surface realizer.

## References

G. Carenini and J.D. Moore. 2000. An empirical study of the influence of argument conciseness on argument effectiveness. In *Proceedings of ACL 2000*.

G. Carenini and J.D. Moore. 2001. An empirical study of the influence of user tailoring on evaluative argument effectiveness. In *IJCAI*, pages 1307–1314.

A. Cawsey. 1993. Planning interactive explanations. *International Journal of Man-Machine Studies*, 38:169–199.

W. Edwards and F.H. Barron. 1994. SMARTS and SMARTER: Improved simple methods for multiattribute utility measurement. *Organizational Behavior and Human Decision Processes*, 60:306–325.

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

M. Johnston and S. Bangalore. 2000. Finite-state multimodal parsing and understanding. In *Proceedings of COLING 2000*.

M. Johnston, S. Bangalore, and G. Vasireddy. 2001. MATCH: Multimodal access to city help. In *Automatic Speech Recognition and Understanding Workshop*, Trento, Italy.

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

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

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

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

M. Walker, R. Passonneau, and J. Boland. 2001. Quantitative and qualitative evaluation of DARPA communicator spoken dialogue systems. In *Proceedings of ACL 2001*.

M. Walker. 1996. The effect of resource limits and task complexity on collaborative planning in dialogue. *Artificial Intelligence Journal*, 85(1–2):181–243.