To appear in the Proceedings of FLAIRS 2004

Generating Tailored, Comparative Descriptions in Spoken Dialogue

Johanna Moore Mary Ellen Foster Oliver Lemon Michael White
School of Informatics, University of Edinburgh
2 Buccleuch Place
Edinburgh, Scotland EH8 9LW
{jmoore,mef,olemon,mwhite}@inf.ed.ac.uk

Abstract

We describe an approach to presenting information in spoken dialogues that for the first time brings together multi-attribute decision models, strategic content planning, state-of-the-art dialogue management, and realization which incorporates prosodic features. The system selects the most important subset of available options to mention and the attributes that are most relevant to choosing between them, based on the user model. It also determines how to organize and express descriptions of selected options and attributes, including determination of information structure and rhetorical structure at the level of content planning, resulting in descriptions which, we hypothesize, are both memorable and easy for users to understand.

Introduction

In evaluating the DARPA Communicator spoken dialogue systems, Walker, Passonneau, & Boland (2001) found that the information presentation phase of complex dialogues is often the primary contributor to dialogue duration. During this phase, the typical system sequentially presents the set of options that match the user's constraints, as shown in Figure 1. The user can then navigate through these options and refine them by offering new constraints. When multiple options are returned, this process can be painstaking, leading to reduced user satisfaction.

As Walker *et al.* (2002) observe, having to access the set of available options sequentially makes it hard for the user to remember information relevant to making a decision. Clearly, 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, as well as the attributes that are most relevant to choosing among them; and
2. determining how to organize and express the descriptions of the selected options and attributes, in ways that are both easy to understand and memorable.

To address the first point, we follow (Carenini & Moore 2000; Walker *et al.* 2002; Carberry, Chu-Carroll, & Elzer 1999) in applying decision-theoretic models of user preferences to the generation of tailored descriptions of the most relevant available options. Such preference models have been shown to enable systems to present information in ways that are both more concise and more tailored to the user's interests (Carenini & Moore 2001; Stent *et al.* 2002). Decision-theoretic models have also been commercially deployed in web systems.[1]

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*
SYS:	...

Figure 1: Typical information presentation phase of a Communicator dialogue

To address the second point, we note that tailoring to user preferences needs to be carried out at all levels of information presentation, so that not only is appropriate content selected, but it is presented appropriately in the current dialogue context, and with intonation that expresses contrasts intelligibly (Prevost 1995). If any of these features is missing, system output can be more difficult for users to process.

In the remainder of this paper, we present the FLIGHTS[2] system, which uses multi-attribute decision models to inform generation at all levels, from the selection of content to linguistic realization, including prosodic information. In addition, we present a new strategy for the presentation of

[1] See http://www.activebuyersguide.com/, for example.

[2] FLIGHTS stands for Fancy Linguistically Informed Generation of Highly Tailored Speech.

multiple options: first we present the best option (with respect to the user model), and then the most compelling remaining options, in terms of trade-offs between attributes which are important to the user.

Tailoring the Descriptions

To illustrate how decision-theoretic models of user preferences can be used to tailor descriptions of the available options at many points in the generation process, let us consider the following three sample users of the FLIGHTS system (see Figure 3 for details):

Student (S) a student who cares most about price, all else being equal

Frequent Flyer (FF) a business traveler who prefers business class, but cares most about building up frequent-flier miles on KLM

Business Class (BC) another business traveler who prefers KLM, but wants, above all, to travel in business class

Suppose that each user is interested in flying from Edinburgh to Brussels on a certain day, and would like to arrive by five in the afternoon. FLIGHTS begins the dialogue by gathering the details necessary to query the database for possible flights. Next, it uses the preferences encoded in the user model to select the highest ranked flight for each user, as well as those flights that offer interesting trade-offs; these flights are then described to the user, as shown in Figure 2. For the student (S), the BMI flight comes out on top, since it is a fairly cheap, direct flight near the desired time. The Ryanair flight is also mentioned as a possibility, as it has the best price; it ends up ranked lower overall than the BMI flight though, because it requires a connection and arrives well in advance of the desired arrival time. For the KLM frequent flier (FF), life is a bit more complicated: a KLM flight with a good arrival time is offered as the top choice, even though it is a connecting flight with no availability in business class. As alternatives, the direct flight on BMI (with no business class availability) and the British Airways flight with seats available in business class (but requiring a connection) are described. Finally, for the must-have-business-class traveler (BC), the British Airways flight with business class available is presented first, despite its requiring a connection; the direct flight on BMI is offered as another possibility.

While user preferences have an immediately apparent impact on content selection, they also have more subtle effects on many aspects of how the selected content is organized and expressed, as explained below:

Referring expressions Rather than always referring to the available flights in the same way, flights of interest are instead described using the attributes most relevant to the user: e.g., *direct flight*, *cheapest flight*, *KLM flight*.

Aggregation For conciseness, multiple attributes may be given in a single sentence, subject to the constraint that attributes whose values are positive or negative for the user should be kept together. For example, in *There's a KLM flight arriving Brussels at four fifty p.m., but business class is not available and you'd need to connect in Amsterdam*, the values of the attributes `airline` and `arrival-time` are considered good, and thus are grouped together to contrast with the values of the attributes `fare-class` and `number-of-legs`, which are considered bad.

User	Output
S	*There's a direct flight on BMI with a good price—it arrives at four ten p.m. and costs a hundred and twelve pounds. The cheapest flight is on Ryanair—it arrives at twelve forty-five p.m. and costs just fifty pounds, but it requires a connection in Dublin.*
FF	*There's a KLM flight arriving Brussels at four fifty p.m., but business class is not available and you'd need to connect in Amsterdam. If you want to fly direct, there's a BMI flight that arrives at four ten p.m., but it has no availability in business class either. There are seats in business class on the British Airways flight that arrives at four twenty p.m.—it requires a connection in Manchester though.*
BC	*You can fly business class on British Airways, arriving at four twenty p.m., but you'd need to connect in Manchester. There is a direct flight on BMI, arriving at four ten p.m., but it has no availability in business class.*

Figure 2: Tailored descriptions of the available flights for three different user models

Discourse cues Attributes with negative values for the user are acknowledged using discourse cues, such as *but* and *though*. Interesting tradeoffs are also signaled using cues such as *if*, or via prosodic emphasis, as in *There* ARE *seats in business class on the British Airways flight that arrives at four twenty p.m.*.

Scalar terms Scalar modifiers like *good*, as in *good price*, and *just*, as in *just fifty pounds*, are chosen to characterize an attribute's value to the user relative to values of the same attribute for other options.

Architecture

The system architecture uses OAA (Martin, Cheyer, & Moran 1998) as a communications hub, with the following agents responsible for specific tasks: the component described in the following section for user modelling; OPlan (Currie & Tate 1991) for content planning; Nuance (`http://www.nuance.com/`) for speech recognition; Gemini (Dowding *et al.* 1993) for parsing; DIPPER (Bos *et al.* 2003) for dialogue management; Xalan XSLT (`http://xml.apache.org/xalan-j/`) and OpenCCG (White & Baldridge 2003) for sentence planning and realization; and Festival (Taylor, Black, & Caley 1998) for speech synthesis. While some of these components can be viewed as "off the shelf" elements of the architecture, the integration of the user modelling component, OPlan, DIPPER, OpenCCG, and Festival is novel.

	Weights							Preferences		
	Arrival	**# Legs**	**Time**	**Price**	**Airline**	**Layover**	**Class**	**Airline**	**Layover**	**Class**
S	.1049	.1049	.1049	***.3704***	.1049	.1049	.1049	–	–	+economy
FF	.1641	.1641	.0728	.0323	***.3704***	.0323	.1641	+KLM	−LHR	+business
BC	.1641	.1641	.1641	.0323	.0728	.0323	***.3704***	+KLM	−LHR	+business

Figure 3: Sample user models

Multi-Attribute Decision Models in FLIGHTS

FLIGHTS uses multi-attribute decision models to represent the user's preferences, as in the MATCH restaurant recommendation system (Walker *et al.* 2002). These models are based on the notion that, if anything is valued, it is valued for multiple reasons, where the relative importance of different reasons may vary among users. To define a user model for the flight-booking domain, we must define the relevant attributes, and then set their relative importance for each user.

The first step is to create a tree-structured model of the attributes in the domain. The main objective is to choose a good flight for a particular origin, destination, and arrival or departure time. The following attributes contribute to this objective: `arrival-time`, `departure-time`, `number-of-legs`, `total-travel-time`, `price`, `airline`, `fare-class`, and `layover-airport`. As in MATCH, these attributes are arranged into a one-level tree. This overall structure is common to all user models; different user models are created by setting the weights of the branches, as described at the end of this section.

For each attribute, we define a function that maps from the features of a flight to a number between 0 and 1 representing the value of that flight for that attribute, where 0 is the worst and 1 is the best.

The functions for the `airline`, `layover-airport`, and `fare-class` attributes make use of user-specified preferred or dispreferred values for that attribute. In the current version of these functions, a preferred value is given a score of 0.8, a dispreferred value 0.2, and all other values 0.5.

The structure and weights of the user model represents a user's *dispositional* biases about flight selection. *Situational* features are incorporated in two ways. The requested origin and destination are used as a filter when selecting the set of available options by querying the database. On the other hand, the requested arrival or departure time—if specified—is used in the corresponding attribute's evaluation function to give a higher score to flights that are closer to the specified time. If an arrival or departure time is not specified, the corresponding attribute is disabled in the user model.

As in previous work, the overall evaluation of an option is computed as the weighted sum of its evaluation on each attribute. That is, if f represents the option being evaluated, N is the total number of attributes, and w_i and v_i are, respectively, the weight and the evaluation function for attribute i, then the evaluation $v(f)$ of option f is computed as follows: $v(f) = \sum_{i=1}^{N} w_i v_i(f)$.

Creating a Specific User Model

To create a user model for a specific user, two types of information are required. The user must rank the attributes in order of importance, and he or she must also specify any preferred or dispreferred attribute values for the `airline`, `layover-airport`, and `fare-class` attributes. In FLIGHTS, we also allow users to specify a partial ordering of the rankings, so that several attributes can be given equal importance. Users specify these rankings and preferences as part of registering to use the FLIGHTS system. Figure 3 shows the user models for the student (S), frequent-flier (FF), and business-class (BC) users discussed earlier; since no departure time is specified in the sample query, `departure-time` is not included in these examples.

Based on the user's ranking of the attributes, weights are assigned to each attribute. As in previous work, we use Rank Order Centroid (ROC) weights (Edwards & Barron 1994). This allows weights to be assigned based on rankings, guaranteeing that the sum will be 1. The nth ROC weight w_n^R of N total weights is computed as follows: $w_n^R = \frac{1}{N}\sum_{i=n}^{N}\frac{1}{i}$.

We extend these initial weights to the partial-ordering case as follows. If attributes $i \ldots j$ all have the same ranking, then the weight of each will be the mean of the relevant ROC weights; that is, $(\sum_{k=i}^{j} w_k^R)/(j-i+1)$. As a concrete example, if there is a single highest-ranked attribute followed by a three-way tie for second, then $w_1 = w_1^R$, while $w_2 = w_3 = w_4 = \frac{1}{3}(w_2^R + w_3^R + w_4^R)$.

Content Selection

Once a specific user model has been created as outlined in the preceding section, it can be used to select a set of flights to describe for that user, and to determine the features of those flights that should be included in the descriptions. We use a strategy that combines features of the *Compare* and *Recommend* strategies of Walker *et al.* (2002). As in the preceding examples, the examples below are based on a request for flights from Edinburgh to Brussels arriving by 5:00pm.

Selecting the Options to Describe

In determining whether an option is worth mentioning, we make use of two measures. Firstly, we use the z-score of each option; this measures how far the evaluation $v(f)$ of an option f is from the mean evaluation. Formally, it is defined as follows (where μ_V is the mean of all evaluations and σ_V is the standard deviation): $z(f) = (v(f) - \mu_V)/\sigma_V$.

We also make use of the *compellingness* measure described by Carenini & Moore (2000), who provide a formal definition. Informally, the compellingness of an attribute measures its strength in contributing to the overall difference between the evaluation of two options, all other things being equal. For options f, g and threshold value k_c, we define the set $comp(f, g, k_c)$ as the set of attributes that have a higher

- Set $Sel := \{f_0\}$, where f_0 is the top-ranked option.
- For each flight f, in decreasing order of evaluation (after f_0):
  - If $z(f) < k_z$, stop.
  - Otherwise, for each attribute a in the user model:
    - If $\forall g \in Sel, a \in comp(f,g,k_c)$, set $Sel := Sel \cup \{f\}$ and continue.

Figure 4: Algorithm for selecting the options to describe

Let *Sel* be the set of selected options, and f_0 the top-ranked option.

- Set $Atts := s\text{-}comp(f_0,k_c)$.
- For all options $f,g \in Sel$:
  - Set $Atts := Atts \cup comp(f,g,k_c)$.

Figure 5: Algorithm for selecting the attributes to include

score for f than for g, and for which the compellingness is above k_c.

The set *Sel* of options to describe is constructed as follows. First, we include the top-ranked option. Next, for all of the other options whose z-score is above k_z, we check whether there is an attribute of that option that offers a compelling trade-off over the already selected options; if so, we add that option to the set. This algorithm is presented formally in Figure 4.

For the BC user model, for example, this algorithm proceeds as follows. First, it selects the top-ranked flight: a connecting flight on British Airways with availability in business class. The next-highest-ranked flight is a morning flight, which does not have any attributes that are compellingly better than than those of the top choice, and is therefore skipped. However, the third option presents an interesting trade-off: even though business class is not available, it is a direct flight, so it is also included. None of the other options above the threshold present any interesting trade-offs, so only those two flights are included.

The selected flights for the other user models show similar trade-offs, as described in the discussion of Figure 2.

Selecting the Attributes to Include

When selecting the attributes to include in the description, we make use of the additional measure of *s-compellingness*. Informally, the s-compellingness of an attribute represents the contribution of that attribute to the evaluation of a single option; again, the formal definition is given by Carenini & Moore (2000). Note that an attribute may be s-compelling in either a positive or a negative way. For an option f and threshold k_c, we define the set $s\text{-}comp(f,k_c)$ as the set of attributes whose s-compellingness for f is greater than k_c.

The set *Atts* of attributes is constructed in two steps. First, we add the most compelling attributes of the top choice. Next, we add all attributes that represent a trade-off between any two of the selected options; that is, attributes that are compellingly better for one option than for another. Figure 5 presents this algorithm formally.

For the BC user model, the s-compelling attributes of the top choice are `arrival-time` and `fare-class`; the latter is also a compelling advantage of this flight over the second option. The advantage of the second option over the first is that it is direct, so `number-of-legs` is also included.

A similar process on the other user models results in `price`, `arrival-time`, and `number-of-legs` being selected for S, and `arrival-time`, `fare-class`, `airline`, and `number-of-legs` for FF.

Content Planning

Based on the information returned by the content selection process, together with information from the user model and the current dialogue context, the OPlan agent develops a strategy for presenting the available options. A distinguishing feature of the resulting content plans is that they contain specifications of the *information structure* of sentences (Steedman 2000), including sentence *theme* (roughly, the topic the sentence addresses) and sentence *rheme* (roughly, the new contribution on a topic). The presentation strategy thus performs several functions:

- marking the status of items as *given/not given*, *definite/indefinite* and *theme/rheme* for information structure;
- *grouping* and *ordering* of similar options and attributes (e.g. presenting the top scoring option first vs. last);
- choosing *referring expressions* to use for options (e.g. referring to a particular option by airline);
- determining *contrast* between options and attributes, or between groups of options and attributes; and
- *decomposing* strategies into basic dialogue acts and rhetorical relations.

For example, consider planning the presentation of the second option in example S. We want to suggest the second option to the user, and to identify it as *the cheapest flight*. The first plan operator shown in Figure 6 illustrates the planning of information structure for this `identify` speech act. In general, the second and subsequent options are identified via their compelling attributes, which are marked as theme because they have been selected to address issues which are known to be *salient* for the user (rheme marking is the default). This operator, marking definiteness, is chosen by the content planner when the attribute is unique for the option. Thus, in example S we can plan to say *the cheapest flight* because we already know that the price attribute is salient for the user and there is only one cheapest flight.[3]

The second plan operator shown in Figure 6 illustrates how the remaining information for the second option of example S of Figure 2 is structured. Here, the option is to be presented in terms of a contrast between its positive and negative attributes, as determined by the user model. Where there are no negative attributes for an option (as in the first option for user S), a different operator is chosen, which presents only the positive attributes.

[3]Note that the type of the option (`flight`) is marked as *given* because we can assume that this is the topic of the dialogue. Information is *given* if it is not new or contrastive; *not given* is the default.

```
schema do_identify1;
 ;;; OPlan variable type declarations omitted
  expands {identify  ?option};
  nodes 1 action {inform ?opt_type given def theme ?option},
        2 action {inform ?pos_att theme ?option};
  orderings 1 ---> 2;
  conditions
          compute {user_model (get_pos_att ?option)} = {?pos_att},
          only_use_if {unique ?pos_att}  = true,
          only_use_if {not_best ?option} = true,
          compute {type ?option} = ?opt_type;
  effects {identified ?option} = true at end_of 2;

schema contrast_pos_neg;
  ;;; OPlan variable type declarations omitted
   expands {describe  ?option};
   nodes  1 action {list ?pos_atts},
          2 action {list ?neg_atts};
   orderings 1 ---> 2;
   conditions
          compute {user_model (get_atts ?option)} = {?pos_atts ?neg_atts},
          only_use_if {empty_set ?neg_atts} = false at begin_of self;
   effects {contrasted_pos_neg ?option} = true at end_of 2;
```

Figure 6: Sample content-plan operators

```
sequence
  elaborate
    suggest f1
      identify f1
        inform [pred=flight arg=f1 status=given]
        inform [pred=direct arg=f1]
      inform [pred=airline arg=f1 val=BMI]
      inform [pred=price arg=f1 eval=good]
    list
      inform [pred=arrival-time arg=f1 val=4.10pm]
      inform [pred=cost arg=f1 val=112GBP]
  elaborate
    suggest f2
      identify1 f2
        inform [pred=flight arg=f2 status=given def=true info=theme]
        inform [pred=cheapest arg=f2 info=theme]
      inform [pred=airline arg=f2 val=Ryanair]
    contrast-pos-neg
      list
        inform [pred=arrival-time arg=f2  val=12.45pm]
        inform [pred=cost  arg=f2 val=50GBP eval=just]
      list
        inform [pred=connection arg=f2 val=Dublin]
```

Figure 7: Sample content planner output for example S

This approach to content planning is significant in that it presents a principled approach to determining information structure. In previous systems these decisions have typically been made at an intermediate sentence planning stage. Note also that the plan operators are not domain-specific, although they are specific to a particular genre of information presentation. Figure 7 shows the content planner output for example S of Figure 2 (in a more readable version than the actual XML).

CCG Generation

Following Prevost (1995) and Steedman (2000), we use Combinatory Categorial Grammar (CCG) to convey the information structure of sentences via intonation. To illustrate, consider again the flights suggested for the student user (where small caps indicate stress, and brackets indicate intonational phrase boundaries):

> There's a direct flight on BMI with a good price—it arrives at four ten p.m. and costs a hundred and twelve pounds. [The CHEAPEST flight]$_{theme}$ [is on RYANAIR]$_{rheme}$—it arrives at twelve forty-five p.m. and costs just fifty pounds, but it requires a connection in Dublin.

```
be [tense=pres info=rh id=n1]
<Arg> flight [num=sg det=the info=th id=f2]
      <HasProp> cheapest [kon=+ id=n2]
<Prop> has-rel [id=n3]
       <Of> f2
       <Airline> Ryanair [kon=+ id=n4]
```

Figure 8: Sample OpenCCG Input Logical Form

In the annotated clause, intonation helps to convey the relations of contrast among the alternatives on offer, as well as the division of the clause into its theme and rheme. The OpenCCG realizer uses these aspects of content to determine the types and locations of pitch accents and boundary tones. For example, the pitch accent on *cheapest*—which distinguishes the Ryanair flight from the BMI one—is determined to be of the variety identified by Pierrehumbert (1980) as L+H*, since it is part of the theme; when combined with the L-H% boundary tone marking the end of the theme, it produces the distinctive rise-fall-rise tune that signals a topic shift to the listener. The pitch accents and boundary tones are encoded in the input to Festival using APML.[4]

To transform the output of the content planner to the input logical forms expected by the OpenCCG realizer, we currently employ a lightweight, domain-specific sentence planner implemented as a set of XSLT templates. The sentence planner is primarily responsible for transforming the input dialogue acts and rhetorical relations into lexicalized predications. For example, the sentence planner translates `inform [pred=connection arg=f2 val=Dublin]` into the logical forms for either *you'd need to connect in Dublin* or *(it) requires a connection in Dublin.* In principle, the sentence planner could be made domain independent, though doing so would require encodings of the detailed knowledge of lexical semantics implicit in such tranformations.

In addition to choosing lexicalized predicates, the sentence planner may also refine the groupings and specifications of referring expressions provided by the content planner, for improved fluency. For example, the sentence planner may combine two acts to inform the user of the destination and the arrival time into the logical form for a single clause, e.g. *arriving Brussels at four fifty p.m.* However, it is important to emphasize that such refinements must only be made if they are consistent with the major decisions made by the content planner, which has access to the user model.

An example output of sentence planning (i.e., OpenCCG input logical form) appears in Figure 8, for the highlighted clause in the example above (in a more readable format than the actual XML). This logical form specifies a present tense clause headed by *be*, with *the* CHEAPEST *flight* as its subject, and *on* RYANAIR as its predicative complement. The `info=rh` feature indicates that the clause has rheme status, with the exception of the subject, where the `info=th` feature indicates its theme status. The `kon=+` feature indicates

[4] Affective Presentation Markup Language; see `http://www.cstr.ed.ac.uk/projects/festival/apml.html`.

new or contrastive predicates whose lexicalizations should be marked with pitch accents (Steedman 2004).

Festival Synthesis

In order to synthesize contextually appropriate intonation using a voice whose segmental quality approaches the level now available in commercial synthesizers, we have built a limited domain voice which permits the specification of appropriate pitch accents and boundary tones, using the techniques developed in (Baker 2003). In Baker's pilot study, she found that such prosodic specifications can improve the perceived naturalness of synthesized utterances, especially when the limited domain voice must be capable of producing the same word sequences with different intonation contours.

Related and Future Work

The FLIGHTS system combines and extends earlier approaches to user-tailored generation in spoken dialogue, and addresses 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. The most similar system to ours is MATCH (Walker *et al.* 2002); however, it employs simpler content planning strategies and uses quick-and-dirty templates for realization. Carenini and Moore's (2000) system is also closely related, but it does not make comparisons, and generates text rather than speech. Carberry *et al.*'s (1999) system likewise employs additive decision models in recommending courses, though their focus is on dynamically acquiring a model of the student's preferences, and their recommendations only come into the picture at a later stage of the dialogue where the system may recommend a single option considered better than the user's current one. In addition, they only address the problem of selecting positive attributes to justify the recommendation, and do not consider how to plan and prosodically realize the positive and negative attributes of multiple suggested options. Finally, Prevost's (1995) generator has directly informed our approach to information structure and prosody; his system does not make use of quantitative user models though, and only describes single options.

Future work will evaluate the effectiveness of our approach. We plan to follow the experimental design of (Stent *et al.* 2002), where the subject is an "overhearer" of a series of dialogues—first in text, then in speech—involving two conditions. In one condition, the system responses are tailored to the subject's user model, whereas in the other condition, system responses are tailored to another (randomly chosen) subject's user model; the subject's ratings of the quality of system responses are then compared by condition. Our focus will be on the effectiveness of our strategy for presenting multiple options; our hypothesis is that we will find a significant preference for appropriately tailored responses.

Acknowledgements

Thanks to Mark Steedman, Colin Matheson, and the anonymous reviewers for helpful discussion. This work was supported in part by EPSRC grant GR/R02450/01.

References

Baker, R. E. 2003. Using unit selection to synthesise contextually appropriate intonation in limited domain synthesis. Master's thesis, Department of Linguistics, University of Edinburgh.

Bos, J.; Klein, E.; Lemon, O.; and Oka, T. 2003. DIPPER: Description and formalisation of an information-state update dialogue system architecture. In *4th SIGdial Workshop on Discourse and Dialogue*, 115–124.

Carberry, S.; Chu-Carroll, J.; and Elzer, S. 1999. Constructing and utilizing a model of user preferences in collaborative consultation dialogues. *Computational Intelligence Journal* 15(3):185–217.

Carenini, G., and Moore, J. 2000. A strategy for generating evaluative arguments. In *Proceedings of INLG-00*.

Carenini, G., and Moore, J. 2001. An empirical study of the influence of user tailoring on evaluative argument effectiveness. In *Proceedings of IJCAI-01*.

Currie, K., and Tate, A. 1991. O-Plan: the open planning architecture. *Artificial Intelligence* 52:49–86.

Dowding, J.; Gawron, J. M.; Appelt, D.; Bear, J.; Cherny, L.; Moore, R.; and Moran, D. 1993. GEMINI: a natural language system for spoken-language understanding. In *Proceedings 31st Annual Meeting of the ACL*.

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

Martin, D.; Cheyer, A.; and Moran, D. 1998. Building distributed software systems with the open agent architecture. In *Proceedings of the Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*.

Pierrehumbert, J. 1980. *The Phonology and Phonetics of English Intonation*. Ph.D. Dissertation, MIT.

Prevost, S. 1995. *A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation*. Ph.D. Dissertation, University of Pennsylvania.

Steedman, M. 2000. Information structure and the syntax-phonology interface. *Linguistic Inquiry* 31(4):649–689.

Steedman, M. 2004. Information-structural semantics for English intonation. In Gordon, M.; Büring, D.; and Lee, C., eds., *LSA Summer Institute Workshop on Topic and Focus, Santa Barbara July 2001*. Dordrecht: Kluwer. To appear.

Stent, A.; Walker, M.; Whittaker, S.; and Maloor, P. 2002. User-tailored generation for spoken dialogue: An experiment. In *Proceedings of ICSLP-02*.

Taylor, P.; Black, A.; and Caley, R. 1998. The architecture of the the Festival speech synthesis system. In *Third International Workshop on Speech Synthesis, Sydney, Australia*.

Walker, M.; Whittaker, S.; Stent, A.; Maloor, P.; Moore, J.; Johnston, M.; and Vasireddy, G. 2002. Speech-plans: Generating evaluative responses in spoken dialogue. In *Proceedings of INLG'02*, 73–80.

Walker, M. A.; Passonneau, R.; and Boland, J. E. 2001. Quantitative and qualitative evaluation of DARPA Communicator spoken dialogue systems. In *Proceedings of ACL-01*.

White, M., and Baldridge, J. 2003. Adapting chart realization to CCG. In *Proceedings of EWNLG9*.