



# THE UNIVERSITY *of* EDINBURGH

## Edinburgh Research Explorer

### Information Presentation in Spoken Dialogue Systems

**Citation for published version:**

Demberg, V & Moore, JD 2006, 'Information Presentation in Spoken Dialogue Systems'. in 11th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 65-72.

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Author final version (often known as postprint)

**Published In:**

11th Conference of the European Chapter of the Association for Computational Linguistics

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# Information Presentation in Spoken Dialogue Systems

**Vera Demberg**

Institute for Natural Language Processing (IMS)  
University of Stuttgart  
D-70174 Stuttgart  
V.Demberg@gmx.de

**Johanna D. Moore**

School of Informatics  
University of Edinburgh  
Edinburgh, EH8 9LW, GB  
J.Moore@ed.ac.uk

## Abstract

To tackle the problem of presenting a large number of options in spoken dialogue systems, we identify compelling options based on a model of user preferences, and present tradeoffs between alternative options explicitly. Multiple attractive options are structured such that the user can gradually refine her request to find the optimal tradeoff. We show that our approach presents complex tradeoffs understandably, increases overall user satisfaction, and significantly improves the user's overview of the available options. Moreover, our results suggest that presenting users with a brief summary of the irrelevant options increases users' confidence in having heard about all relevant options.

## 1 Introduction

The goal of spoken dialogue systems (SDS) is to offer efficient and natural access to applications and services, such as email and calendars, travel and entertainment booking, and product recommendation. In evaluating nine SDS in the DARPA Communicator domain (flight, hotel, and rental car hire), Walker et al. (2001) found that (1) shorter task duration correlates with higher user satisfaction, and (2) the information presentation phase of dialogues is the primary contributor to dialogue duration. During this phase, the typical system enumerates the set of options that match the user's constraints, as shown in Figure 1. The user can then refine these options by offering new constraints. When the number of options to be presented is large, this process can be painstaking, leading to reduced user satisfaction. Moreover, as

S: *I found 28 round trips from Edinburgh to Brussels on October 31st and returning Edinburgh on November 6th. For option 1, I have an outbound flight with KLM departing at 7 am and arriving at 8:49 am non-stop. A return flight is with KLM departing at 10:30 am and arriving at 11:28 am with 1 stopover. Ticket price is £165. Please say "next option", or, "flight details", or, "I'll take it".*

U: Next option

S: *For option 2, I have an outbound flight with KLM departing at 3:30 pm and arriving at 5:19 pm nonstop. A return flight is with KLM departing at 8:25 pm and arriving at 12:28 pm with 1 stopover. Ticket price is £165 please say "next option", or, "flight details", or, "I'll take it."*

U: Next option

S: ...

Figure 1: Typical Information Presentation Phase of a Communicator Dialogue

Walker et al. (2004) observe, having to access the set of available options sequentially makes it difficult for the user to remember the various aspects of multiple options and to compare them in memory.

Clearly, alternative strategies to sequential presentation of information in SDS are needed. Recently, two approaches have been proposed. In the user-model (UM) based approach, the system identifies a small number of options that best match the user's preferences (Moore et al., 2004; Walker et al., 2004). In the summarize and refine (SR) approach, the system structures the large number of options into a small number of clusters that share attributes. The system summarizes the clusters based on their attributes and then prompts the user to provide additional constraints (Polifroni et al., 2003; Chung, 2004).

In this paper, we present an algorithm that combines the benefits of these two approaches in an approach to information presentation that integrates user modelling with automated clustering.

Thus, the system provides detail only about those options that are of some relevance to the user, where relevance is determined by the user model. If there are multiple relevant options, a cluster-based tree structure orders these options to allow for stepwise refinement. The effectiveness of the tree structure, which directs the dialogue flow, is optimized by taking the user’s preferences into account. Complex tradeoffs between alternative options are presented explicitly to allow for a better overview and a more informed choice. In addition, we address the issue of giving the user a good overview of the option space, despite selecting only the relevant options, by briefly accounting for the remaining (irrelevant) options.

In the remainder of this paper, we describe the prior approaches in more detail, and discuss their limitations (Section 2). In section 3, we describe our approach, which integrates user preferences with automated clustering and summarization in an attempt to overcome the problems of the original approaches. Section 4 presents our clustering and content structuring algorithms and addresses issues in information presentation. In Section 5, we describe an evaluation of our approach and discuss its implications.

## 2 Previous Work in Information Presentation

### 2.1 Tailoring to a User Model

Previous work in natural language generation showed how a multi-attribute decision-theoretic model of user preferences could be used to determine the attributes that are most relevant to mention when generating recommendations tailored to a particular user (Carenini and Moore, 2001). In the MATCH system, Walker et al. (2004) applied this approach to information presentation in SDS, and extended it to generate summaries and comparisons among options, thus showing how the model can be used to determine which options to mention, as well as the attributes that the user will find most relevant to choosing among them. Evaluation showed that tailoring recommendations and comparisons to the user increases argument effectiveness and improves user satisfaction (Stent et al., 2002).

MATCH included content planning algorithms to determine what options and attributes to mention, but used a simple template based approach to realization. In the FLIGHTS system, Moore

et al. (2004) focussed on organizing and expressing the descriptions of the selected options and attributes, in ways that are both easy to understand and memorable. For example, Figure 2 shows a description of options that is tailored to a user who prefers flying business class, on direct flights, and on KLM, in that order. In FLIGHTS, coherence and naturalness of descriptions were increased by reasoning about information structure (Steedman, 2000) to control intonation, using referring expressions that highlight attributes relevant to the user (e.g., “the cheapest flight” vs. “a KLM flight” ), and signalling discourse relations (e.g., contrast) with appropriate intonational and discourse cues.

S: *You can fly business class on KLM, arriving at four twenty p.m., but you'd need to connect in London. There is a direct flight on BMI, arriving at four ten p.m., but it has no availability in business class.*

Figure 2: Tailored description by FLIGHTS

This prior work demonstrated that the user model-based approach can concisely present a relatively small number of options, pointing out the ways in which those options satisfy user preferences. It is an appropriate strategy for SDS when there are a small number of options to present, either because the number of options is limited or because users can supply sufficient constraints to winnow down a large set before querying the database of options.

However, there are several limitations to this approach. First, it does not scale up to presenting a large number of options. When there are hundreds of options to consider (e.g., when choosing among consumer products, hotels, or restaurants) there may be many options that are close in score. In addition, users may not be able to provide constraints until they hear more information about the space of options. This brings up a second problem with the UM-based approach, namely that it does not provide the user with an overview of the option space, because options scoring below a specified threshold are not mentioned. This is related to the third problem, which is that users might miss out on options they would have chosen if they had heard about them. These last two problems may reduce user confidence in the system, if users have the perception that the system is not telling them about all of the available options. This may ultimately lead to a decrease in user satisfaction.

## 2.2 Stepwise Refinement through Clustering and Summarization

Polifroni et al. (2003) developed an approach that structures large datasets for summarization and successive refinement. It supports the user in narrowing in on a suitable option by clustering the options in the database, summarizing the option clusters and suggesting further constraints. In this approach, the prompts presented to the user, and the order in which they appear are determined at run-time based on an algorithm that computes the most useful set of attributes, as dictated by the current data subset. For large data sets, attributes that partition the data into the minimal number of clusters are chosen, so that a concise summary can be presented to the user to refine, see Figure 3.

M: What restaurants do you have?
S: <i>I have found 983 restaurants. Most of them are located in Boston and Cambridge. There are 32 choices for cuisine. I also have information about price range.</i>
M: Okay tell me about the ones in Boston.
S: <i>I have found 401 restaurants in Boston. There are 29 choices for cuisine.</i>
M: Do you have any that serve seafood?
S: <i>I have found 19 seafood restaurants. They are predominantly in Back Bay, the North End, South Boston and the South End.</i>
M: In the South End.
S: <i>I have found 3 restaurants. Grill Fish on Columbus Avenue, McCormick &amp; Schmick's on Columbus Avenue and Joseph's on High Street.</i>

Figure 3: Dialogue between simulator (M) and Polifroni system (S)

Polifroni et al.'s approach was extended by Chung (2004), who proposed a constraint relaxation strategy for coping with queries that are too restrictive to be satisfied by any option. Qu and Beale (2003) had previously addressed the problem of responding to user queries with several constraints and used linguistic cues to determine which constraints had to be relaxed. Our discussion and evaluation of the SR approach is based on Chung's version.

Although the SR approach provides a solution to the problem of presenting information when there are large numbers of options in a way that is suitable for SDS, it has several limitations. First, there may be long paths in the dialogue structure. Because the system does not know about the user's preferences, the option clusters may contain many irrelevant entities which must be filtered out successively with each refinement step. In addition, the difficulty of summarizing options typi-

cally increases with their number, because values are more likely to be very diverse, to the point that a summary about them gets uninformative ("I found flights on 9 airlines.").

A second problem with the SR approach is that exploration of tradeoffs is difficult when there is no optimal option. If at least one option satisfies all requirements, this option can be found efficiently with the SR strategy. But the system does not point out alternative tradeoffs if no "optimal" option exists. For example, in the flight booking domain, suppose the user wants a flight that is cheap and direct, but there are only expensive direct and cheap indirect flights. In the SR approach, as described by Polifroni, the user has to ask for cheap flights and direct flights separately and thus has to explore different refinement paths.

Finally, the attribute that suggests the next user constraint may be suboptimal. The procedure for computing the attribute to use in suggesting the next restriction to the user is based on the considerations for efficient summarization, that is, the attribute that will partition the data set into the smallest number of clusters. If the attribute that is best for summarization is not of interest to this particular user, dialogue duration is unnecessarily increased, and the user may be less satisfied with the system, as the results of our evaluation suggest (see section 5.2).

## 3 Our Approach

Our work combines techniques from the UM and SR approaches. We exploit information from a user model to reduce dialogue duration by (1) selecting all options that are relevant to the user, and (2) introducing a content structuring algorithm that supports stepwise refinement based on the ranking of attributes in the user model. In this way, we keep the benefits of user tailoring, while extending the approach to handle presentation of large numbers of options in an order that reflects user preferences. To address the problem of user confidence, we also briefly summarize options that the user model determines to be irrelevant (see section 4.3). Thus, we give users an overview of the whole option space, and thereby reduce the risk of leaving out options the user may wish to choose in a given situation.

The integration of a user model with the clustering and structuring also alleviates the three problems we identified for the SR approach. When a

user model is available, it enables the system to determine which options and which attributes of options are likely to be of interest to the particular user. The system can then identify compelling options, and delete irrelevant options from the refinement structure, leading to shorter refinement paths. Furthermore, the user model allows the system to determine the tradeoffs among options. These tradeoffs can then be presented explicitly. The user model also allows the identification of the attribute that is most relevant at each stage in the refinement process. Finally, the problem of summarizing a large number of diverse attribute values can be tackled by adapting the cluster criterion to the user's interest.

In our approach, information presentation is driven by the user model, the actual dialogue context and the available data. We allow for an arbitrarily large number of alternative options. These are structured so that the user can narrow in on one of them in successive steps. For this purpose, a static option tree is built. Because the structure of the option tree takes the user model into account, it allows the system to ask the user to make the most relevant decisions first. Moreover, the option tree is pruned using an algorithm that takes advantage of the tree structure, to avoid wasting time by suggesting irrelevant options to the user. The tradeoffs (e.g., cheap but indirect flights vs. direct but expensive flights) are presented to the user explicitly, so that the user won't have to "guess" or try out paths to find out what tradeoffs exist. Our hypothesis was that explicit presentation of tradeoffs would lead to a more informed choice and decrease the risk that the user does not find the optimal option.

## 4 Implementation

Our approach was implemented within a spoken dialogue system for flight booking. While the content selection step is a new design, the content presentation part of the system is an adaptation and extension of the work on generating natural sounding tailored descriptions reported in (Moore et al., 2004).

### 4.1 Clustering

The clustering algorithm in our implementation is based on that reported in (Polifroni et al., 2003). The algorithm can be applied to any numerically ordered dataset. It sorts the data into bins that

roughly correspond to small, medium and large values in the following way. The values of each attribute of the objects in the database (e.g., flights) are clustered using agglomerative group-average clustering. The algorithm begins by assigning each unique attribute value to its own bin, and successively merging adjacent bins whenever the difference between the means of the bins falls below a varying threshold. This continues until a stopping criterion (a target number of no more than three clusters in our current implementation) is met. The bins are then assigned predefined labels, e.g., `cheap`, `average-price`, `expensive` for the `price` attribute.

Clustering attribute values with the above algorithm allows for database-dependent labelling. A £300 flight gets the label `cheap` if it is a flight from Edinburgh to Los Angeles (because most other flights in the database are more costly) but `expensive` if it is from Edinburgh to Stuttgart (for which there are a lot of cheaper flights in the data base). Clustering also allows the construction of user valuation-sensitive clusters for categorical values, such as the attribute `airline`: They are clustered to a group of `preferred airlines`, `dispreferred airlines` and `airlines the user does not-care about`.

### 4.2 Building up a Tree Structure

The tree building algorithm works on the clusters produced by the clustering algorithm instead of the original values. Options are arranged in a refinement tree structure, where the nodes of an option tree correspond to sets of options. The root of the tree contains all options and its children contain complementary subsets of these options. Each child is homogeneous for a given attribute (e.g., if the parent set includes all direct flights, one child might include all direct cheap flights whereas another child includes all direct expensive flights). Leaf-nodes correspond either to a single option or to a set of options with very similar values for all attributes.

This tree structure determines the dialogue flow. To minimize the need to explore several branches of the tree, the user is asked for the most essential criteria first, leaving less relevant criteria for later in the dialogue. Thus, the branching criterion for the first level of the tree is the attribute that has the highest weight according to the user model. For example, Figure 5 shows an option tree structure

rank	attributes
1	fare class (preferred value: business)
2	arrival time, # of legs, departure time, travel time
6	airline (preferred value: KLM)
7	price, layover airport

Figure 4: Attribute ranking for business user

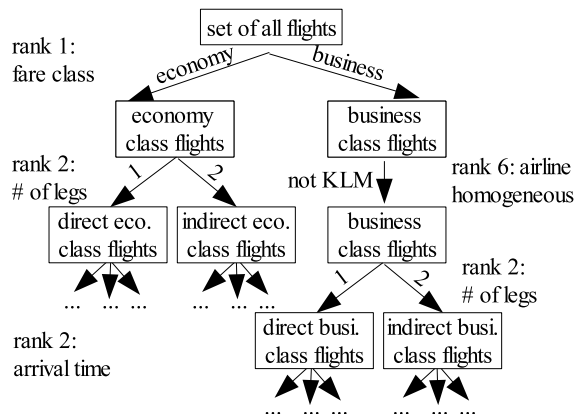


Figure 5: Option tree for business user

for our “business” user model (Figure 4).

The advantage of this ordering is that it minimizes the probability that the user needs to backtrack. If an irrelevant criterion had to be decided on first, interesting tradeoffs would risk being scattered across the different branches of the tree.

A special case occurs when an attribute is homogeneous for all options in an option set. Then a unary node is inserted regardless of its importance. This special case allows for more efficient summarization, e.g., “There are no business class flights on KLM.” In the example of Figure 5, the attribute `airline` is inserted far up in the tree despite its low rank.

The user is not forced to impose a total ordering on the attributes but may specify that two attributes, e.g., `arrival-time` and `number-of-legs`, are equally important to her. This partial ordering leads to several attributes having the same ranking. For equally ranked attributes, we follow the approach taken by Polifroni et al. (2003). The algorithm selects the attribute that partitions the data into the smallest number of sub-clusters. For example, in the tree in Figure 5, `number-of-legs`, which creates two sub-clusters for the data set (direct and indirect), comes before `arrival-time`, which splits the set of economy class flights into three subsets.

The tree building algorithm introduces one of the main differences between our structuring and

Polifroni’s refinement process. Polifroni et al.’s system chooses the attribute that partitions the data into the smallest set of unique groups for summarization, whereas in our system, the algorithm takes the ranking of attributes in the user model into account.

### 4.3 Pruning the Tree Structure

To determine the relevance of options, we did not use the notion of compellingness (as was done in (Moore et al., 2004; Carenini and Moore, 2001)), but instead defined the weaker criterion of “dominance”. *Dominant* options are those for which there is no other option in the data set that is better on all attributes. A *dominated* option is in all respects equal to or worse than some other option in the relevant partition of the data base; it should not be of interest for any rational user. All dominant options represent some tradeoff, but depending on the user’s interest, some of them are more interesting tradeoffs than others.

Pruning dominated options is crucial to our structuring process. The algorithm uses information from the user model to prune all but the dominant options. Paths from the root to a given option are thereby shortened considerably, leading to a smaller average number of turns in our system compared to Polifroni et al.’s system.

An important by-product of the pruning algorithm is the determination of attributes which make an option cluster compelling with respect to alternative clusters (e.g., for a cluster containing direct flights, as opposed to flights that require a connection, the justification would be `#-of-legs`). We call such an attribute the “justification” for a cluster, as it justifies its existence, i.e., is the reason it is not pruned from the tree. Justifications are used by the generation algorithm to present the tradeoffs between alternative options explicitly.

Additionally, the reasons why options have been pruned from the tree are registered and provide information for the summarization of bad options in order to give the user a better overview of the option space (e.g., “All other flights are either indirect or arrive too late.”). To keep summaries about irrelevant options short, we back off to a default statement “or are undesirable in some other way.” if these options are very heterogeneous.

## 4.4 Presenting Clusters

### 4.4.1 Turn Length

In a spoken dialogue system, it is important not to mention too many facts in one turn in order to keep the memory load on the user manageable. Obviously, it is not possible to present all of the options and tradeoffs represented in the tree in a single turn. Therefore, it is necessary to split the tree into several smaller trees that can then be presented over several turns. In the current implementation, a heuristic cut-off point (no deeper than two branching nodes and their children, which corresponds to the nodes shown in Figure 5) is used. This procedure produces a small set of options to present in a turn and includes the most relevant advantages and disadvantages of an option. The next turn is determined by the user's choice indicating which of the options she would like to hear more about (for illustration see Figure 6).

### 4.4.2 Identifying Clusters

The identification of an option set is based on its justification. If an option is justified by several attributes, only one of them is chosen for identification. If one of the justifications is a contextually salient attribute, this one is preferred, leading to constructions like: "...you'd have to make a connection in Brussels. If you want to fly direct,..."). Otherwise, the cluster is identified by the highest ranked attribute e.g., "There are four flights with availability in business class.". If an option cluster has no compelling homogeneous attribute, but only a common negative homogeneous attribute, this situation is acknowledged: e.g., "If you're willing to travel economy / arrive later / accept a longer travel time, ...".

### 4.4.3 Summarizing Clusters

After the identification of a cluster, more information is given about the cluster. All positive homogeneous attributes are mentioned and contrasted against all average or negative attributes. An attribute that was used for identification of an option is not mentioned again in the elaboration. In opposition to a single flight, attributes may have different values for the entities within a set of flights. In that case, these attribute values need to be summarized.

There are three main cases to be distinguished:

1. The continuous values for the attributes `price`, `arrival-time` etc. need to be

summarized, as they may differ in their values even if they are in the same cluster. One way to summarize them is to use an expression that reflects their value range, e.g. "between  $x$  and  $y$ ". Another solution is to mention only the evaluation value, leading to sentences like "The two flights *with shortest travel time*" or "The *cheapest* flights."

2. For discrete-valued attributes with a small number of possible values, e.g., `number-of-legs` and `fare-class`, summarization is not an issue, because when homogeneous for a cluster, the attribute values of its options are identical.
3. The third group are attributes with categorical values, e.g., "airline". If there are no more than three different values, we summarize using quantifications like "none/all/both of them", as done in (Polifroni et al., 2003). If the values are more diverse, the user model comes back into play to produce a tailored summary based on user preferences (e.g., liking KLM). For example, we would generate "None are on KLM.", which takes into account the user's preference and is shorter than mentioning all airlines the flights are on.

An issue arising from summarization with negation is that the negated value has to be salient, otherwise the utterance might be irritating. For example, it would be better to say "These flights are not direct." in a neutral context, but "You would not need to connect in London Heathrow." if London Heathrow had already been mentioned.

A sample dialogue produced by our system, when given the business user model (see Figure 4), is shown in Figure 6.

## 5 Evaluation

A within-participants laboratory experiment was conducted in order to determine whether user model-based clustering leads to increased overall user satisfaction, a better overview of the available options, quicker accessibility to the optimal option and higher confidence of having heard all relevant options. The experiment furthermore assessed whether the options were presented in a way that users found understandable and recorded the time users took to read a dialogue turn.

U: Hello. I'd like to book a flight from Edinburgh to Brussels for tomorrow arriving at around 3pm.

S: *There are four flights with availability in business class. None are on KLM. The only direct flight arrives at 5:30 pm, which is later than you requested. To arrive earlier, you'll have to make a connection. If you're willing to travel economy, there is a direct flight on BMI, arriving at 1:10 pm. There's also a KLM flight arriving at 1:50 pm, but it requires a connection in Amsterdam. All other economy flights take longer or arrive after 4 pm.*

U: Can you tell me more about the business class flights that arrive earlier?

S: *The two flights with shortest travel time are both on British Airways and cost 308 pounds. The first takes 4 hours and 20 minutes and arrives at 1:20 pm. The other takes 4 hours and 30 minutes and arrives at 3:30pm. There's a cheaper flight on Lufthansa. It arrives at 3:10pm and costs just 270 pounds, but it takes 5 hours and 35 minutes. All other business class flights arrive very early or too late.*

Figure 6: Example Dialogue with our System

## 5.1 Experimental Design

Each of the 38 subjects who completed the experiment was presented with six dialogue pairs, the first of which was used for training and was thus not included in the analysis. Each dialogue pair consisted of one dialogue between a user and our system and one dialogue between the same user and a system designed as described in (Polifroni et al., 2003; Chung, 2004) (cf. Section 2.2). Some of the dialogues with our system were constructed manually based on the content selection and structuring step, because the generation component did not cover all linguistic constructions needed. The dialogues with the Chung system were designed manually, as this system is implemented for another domain. The order of the dialogues in a pair was randomized. The dialogues were provided as transcripts.

After reading each dialogue transcript, participants were asked four questions about the system's responses. They provided their answers using Likert scales.

1. Did the system give the information in a way that was easy to understand?  
1: very hard to understand  
7: very easy to understand
2. Did the system give you a good overview of the available options?  
1: very poor overview  
7: very good overview
3. Do you think there may be flights that are better options for  $X^1$  that the system did not tell  $X^1$  about?

- 1: I think that is very possible
- 7: I feel the system gave a good overview of all options that are relevant for  $X^1$ .

4. How quickly did the system allow  $X^1$  to find the optimal flight?  
1: slowly  
3: quickly

After reading each pair of dialogues, the participants were also asked the forced choice question: "Which of the two systems would you recommend to a friend?" to assess user satisfaction.

## 5.2 Results

A significant preference for our system was observed. (In the diagrams, our system which combines user modelling and stepwise refinement is called UMSR, whereas the system based on Polifroni's approach is called SR.) There were a total of 190 forced choices in the experiment (38 participants \* 5 dialogue pairs). UMSR was preferred 120 times ( $\approx 0.63\%$ ), whereas SR was preferred only 70 times ( $\approx 0.37\%$ ). This difference is highly significant ( $p < 0.001$ ) using a two-tailed binomial test. Thus, the null-hypothesis that both systems are preferred equally often can be rejected with high confidence.

The evaluation results for the Likert scale questions confirmed our expectations. The SR dialogues received on average slightly higher scores for understandability (question 1), which can be explained by the shorter length of the system turns for that system. However, the difference is not statistically significant ( $p = 0.97$  using a two-tailed paired t-test). The differences in results for the other questions are all highly statistically significant, especially for question 2, assessing the quality of overview of the options given by the system responses, and question 3, assessing the confidence that all relevant options were mentioned by the system. Both were significant at  $p < 0.0001$ . These results confirm our hypothesis that our strategy of presenting tradeoffs explicitly and summarizing irrelevant options improves users' overview of the option space and also increases their confidence in having heard about all relevant options, and thus their confidence in the system. The difference for question 4 (accessibility of the optimal option) is also statistically significant ( $p < 0.001$ ). Quite surprisingly, subjects reported that they felt they could access options more quickly even though the dialogues were usually longer. The average scores (based on 190 val-

<sup>1</sup> $X$  was instantiated by name of our example users.



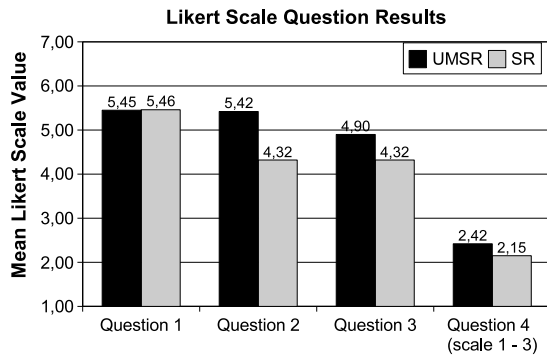


Figure 7: Results for all Questions

ues) are shown in Figure 7.

To get a feel for whether the content given by our system is too complex for oral presentation and requires participants to read system turns several times, we recorded reading times and correlated them to the number of characters in a system turn. We found a linear relation, which indicates that participants did not re-read passages and is a promising sign for the use of our strategy in SDS.

## 6 Conclusions and Future Work

In this paper, we have shown that information presentation in SDS can be improved by an approach that combines a user model with structuring of options through clustering of attributes and successive refinement. In particular, when presented with dialogues generated by a system that combines user modelling with successive refinement (UMSR) and one that uses refinement without reference to a user model (SR), participants reported that the combined system provided them with a better overview of the available options and that they felt more certain to have been presented with all relevant options. Although the presentation of complex tradeoffs usually requires relatively long system turns, participants were still able to cope with the amount of information presented. For some dialogues, subjects even felt they could access relevant options more quickly despite longer system turn length.

In future work, we would like to extend the clustering algorithm to not use a fixed number of target clusters but to depend on the number of natural clusters the data falls into. We would also like to extend it to be more sensitive to the user model when forming clusters (e.g., to be more sensitive at lower price levels for a user for whom price is very important than for a user who does not care

about price).

The explicit presentation of tradeoffs made by the UMSR system in many cases leads to dialogue turns that are more complex than typical dialogue turns in the SR system. Even though participants did not report that our system was harder to understand, it would be interesting to investigate how well users can understand and remember information from the system when part of their concentration is absorbed by another task, for example when using the system while driving a car.

## Acknowledgments

We would like to thank the anonymous reviewers for their comments. The research is supported by the TALK project (European Community IST project no. 507802), <http://www.talk-project.org>. The first author was supported by Evangelisches Studienwerk e.V. Villigst.

## References

- G. Carenini and J.D. Moore. 2001. An empirical study of the influence of user tailoring on evaluative argument effectiveness. In *Proc. of IJCAI 2001*.
- G. Chung. 2004. Developing a flexible spoken dialog system using simulation. In *Proc. of ACL '04*.
- V. Demberg. 2005. Information presentation in spoken dialogue systems. Master's thesis, School of Informatics, University of Edinburgh.
- J.D. Moore, M.E. Foster, O. Lemon, and M. White. 2004. Generating tailored, comparative descriptions in spoken dialogue. In *Proc. of the 17th International Florida Artificial Intelligence Research Society Conference, AAAI Press*.
- J. Polifroni, G. Chung, and S. Seneff. 2003. Towards automatic generation of mixed-initiative dialogue systems from web content. In *Proc. of Eurospeech '03*, Geneva, Switzerland, pp. 193–196.
- Y. Qu and S. Beale. 1999. A constraint-based model for cooperative response generation in information dialogues. In *AAAI/IAAI 1999* pp. 148–155.
- M. Steedman 2000. Information structure and the syntax-phonology interface. In *Linguistic Inquiry*, 31(4): 649–689.
- A. Stent, M.A. Walker, S. Whittaker, and P. Maloor. 2002. User-tailored generation for spoken dialogue: an experiment. In *Proc. of ICSLP-02*.
- M.A. Walker, S. Whittaker, A. Stent, P. Maloor, J.D. Moore, M. Johnston, and G. Vasireddy. 2004. Generation and evaluation of user tailored responses in dialogue. In *Cognitive Science* 28: 811–840.
- M.A. Walker, R. Passonneau, and J.E. Boland. 2001. Quantitative and qualitative evaluation of DARPA communicator spoken dialogue systems. In *Proc of ACL-01*.