A Strategy for Information Presentation in Spoken Dialog Systems

Vera Demberg*
Saarland University

Andi Winterboer**
University of Amsterdam

Johanna D. Moore†
University of Edinburgh

In spoken dialog systems, information must be presented sequentially, making it difficult to quickly browse through a large number of options. Recent studies have shown that user satisfaction is negatively correlated with dialog duration, suggesting that systems should be designed to maximize the efficiency of the interactions. Analysis of the logs of 2,000 dialogs between users and nine different dialog systems reveals that a large percentage of the time is spent on the information presentation phase, thus there is potentially a large pay-off to be gained from optimizing information presentation in spoken dialog systems.

This article proposes a method that improves the efficiency of coping with large numbers of diverse options by selecting options and then structuring them based on a model of the user's preferences. This enables the dialog system to automatically determine trade-offs between alternative options that are relevant to the user and present these trade-offs explicitly. Multiple attractive options are thereby structured such that the user can gradually refine her request to find the optimal trade-off.

To evaluate and challenge our approach, we conducted a series of experiments that test the effectiveness of the proposed strategy. Experimental results show that basing the content structuring and content selection process on a user model increases the efficiency and effectiveness of the user's interaction. Users complete their tasks more successfully and more quickly. Furthermore, user surveys revealed that participants found that the user-model based system presents complex trade-offs understandably and increases overall user satisfaction. The experiments also indicate that presenting users with a brief overview of options that do not fit their requirements significantly improves the user's overview of available options, also making them feel more confident in having been presented with all relevant options.

---

* Cluster of Excellence, Saarland University, Campus C7 4, 66041 Saarbrücken, Germany.
E-mail: `vera@coli.uni-saarland.de`.

** Intelligent Systems Lab Amsterdam, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands. E-mail: `A.Winterboer@uva.nl`.

† School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, UK.
E-mail: `j.moore@ed.ac.uk`.

Submission received: 18 June 2009; revised submission received: 4 October 2010; accepted for publication: 1 December 2010.

© 2011 Association for Computational Linguistics

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

1. Introduction

A common goal of many spoken dialog systems (SDSs) is to offer efficient and natural access to applications and services, such as e-mail, calendars, travel booking, navigation systems, and product recommendation, in situations where the user's hands and/or eyes are busy with another task, for example driving a car (Pon-Barry, Weng, and Varges 2006) or operating equipment (Hieronymus and Dowding 2007). The naturalness and usability of a spoken dialog interface depends not only on its ability to recognize and interpret user utterances correctly, but also on its ability to present information in ways that users can understand and that help them to achieve their goals.

One class of SDSs that has received considerable attention from both academic research and industry are information-seeking SDSs, which are designed to enable users to browse the space of available options (e.g., flights, hotels, movies) and choose a suitable option from a potentially large set of choices. Dialogs with such systems typically consist of two main types of activity: information gathering, in which the system tries to establish users' constraints and preferences, and information presentation, in which the system typically enumerates the set of options that match the user's constraints. An example is given in Figure 1. In some systems, these activities take place in strictly sequential phases: All of the information necessary to form a database query is gathered, and then the returned options are presented, one at a time or in small groups. In other systems, the activities are interleaved, with users refining their constraints after being presented with some options, or a summary of the option space. In either case, when the number of options to be presented is large, this process can be laborious, leading to reduced user satisfaction. Moreover, as 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 mentally.

Although much research has been conducted on the information gathering phase of spoken dialog systems, relatively little attention has been devoted to information presentation. An analysis of the Communicator corpus consisting of approximately 2,000 dialogs with nine different spoken dialog systems found that information presentation is the main contributor to dialog duration[1] (Moore 2006); see Table 1. Moreover, the DARPA Communicator evaluation showed that task duration is negatively correlated with user satisfaction ($r = -0.31, \mathrm{p} < 0.001$, see Walker, Passonneau, and Boland [2001]). Thus, there is reason to believe that improvements in information presentation will lead to improvements in spoken dialog systems.

Recently, two approaches to information presentation that present an alternative to sequential information presentation 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 then summarizes the clusters based on their attributes, thus prompting the user to provide additional constraints (Polifroni, Chung, and Seneff 2003; Chung 2004).

In this article, we propose an approach to information presentation which shortens dialog duration by combining the benefits of these two approaches (UMSR). Our

---

[1] This analysis was performed on the Communicator corpus which has been annotated extensively, including annotations for speech act types and timing information (Walker and Passonneau 2001; Georgila et al. 2009).

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

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 dialog.

Table 1
System contributions: Requesting and presenting information in Communicator systems.

System	Requesting info	Presenting info	Other
Utterances	43%	25%	32%
Time	31%	54%	15%
Words	28%	50%	22%

approach integrates user modeling with automated clustering such that information is structured in a way that enables users to more effectively and efficiently browse the option space. The system provides detail only about those options that are relevant 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 dialog flow, is optimized by taking the user's preferences into account. In order to give the user a good overview of the option space, trade-offs between alternative options are presented explicitly. In addition, despite selecting only the relevant options, the algorithm also briefly accounts for the remaining (irrelevant) options. We hypothesize that this approach will enable users to make more informed choices. Our approach to the problem has been implemented within FLIGHTS, a spoken dialog system for flight booking (Moore et al. 2004; White, Clark, and Moore 2010). Our results show that in addition to improving dialog efficiency (in terms of number of dialog turns) and effectiveness (in terms of successful task completion), our approach increases user satisfaction.

We hypothesize that user modeling in combination with content selection and structuring as implemented in our UMSR strategy can improve the information presentation phase of spoken dialog systems in the following ways:

1. UMSR leads to increased efficiency of information presentation.
2. UMSR makes information presentation more effective.
3. UMSR enables the system to provide users with a better overview of the option space and leads to higher confidence in having heard about all relevant options.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

4. Tailoring sentence realization to the user's preferences, through the use of discourse cues and comparisons (e.g., "the cheapest"), improves understandability.
5. UMSR ultimately leads to greater user satisfaction.

In the remainder of this article, we first describe prior approaches to the problem of information presentation in spoken dialog systems and discuss their advantages and limitations in more detail (Section 2). In Section 3, we describe our approach and its implementation within a spoken dialog system for flight information. Sections 4 through 7 present the user studies we have run to evaluate our approach. In Section 4, we describe two initial studies in which participants rated dialogs they read or overheard. Section 5 describes a modification to the UMSR system to control the length of system-generated dialog turns. Section 6 reports on an experiment in which participants interacted with the revised system. Results of all experiments are discussed in Section 7. Finally, we comment on the relation of this system to other systems from the literature in section 8, and discuss implications of our findings and future directions in Section 9.

2. Background on User Modeling and Content Structuring for Information Presentation

2.1 Tailoring to a User Model (UM)

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

MATCH included content planning algorithms to determine what options and attributes to mention, but used a simple template-based approach to realization. For the design of the FLIGHTS[2] system, Moore et al. (2004) focused on organizing and expressing the descriptions of the selected options and attributes in ways that were intended to make the descriptions both easy to understand and memorable. In addition, to increase coherence and naturalness of the descriptions, the system reasons about information structure (Steedman 2000) to control intonation, uses referring expressions that highlight attributes relevant to the user (e.g., *a direct flight* for a user who wants to minimize connections, vs. *the cheapest flight* for a user concerned about price), and signals discourse relations (e.g., contrast) with appropriate intonational and discourse cues. For example, Figure 2 shows a description of options tailored to a user who prefers flying business class, on direct flights, and on KLM, in that order.

The FLIGHTS system presents a small number of options that best match the user's constraints, and points out ways in which those options satisfy user preferences. Selecting a small number of options and presenting only these is an appropriate strategy for an SDS when the number of options to be presented is small, either because the number

---

[2] FLIGHTS expands as Fancy Linguistically Informed Generation of Highly Tailored Speech.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

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.

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 of this UM-based approach. First, selecting a small number of options from those that best match the user model does not scale up to situations where the number of relevant options is large. When there are hundreds of options to consider (e.g., when choosing among consumer products, hotels, or restaurants) there may be many options that fit the user's specification and interest. In addition, the user model may not contain enough information for new users, or users may not be able to provide constraints until they hear more information about the available 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 or below a certain rank are not mentioned. This is related to the third problem, which is the actual or perceived risk that users might miss out on options they would have chosen if they had heard about them. The 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, and a lack of confidence may ultimately lead to a decrease in user satisfaction.

Finally, the evaluation of the FLIGHTS system focused on the effectiveness of using information structure to generate more natural sounding utterances by controlling intonation, but did not include an explicit comparison to other information presentation strategies. The work presented here extends the FLIGHTS approach to overcome the weaknesses pointed out herein, and presents a series of experiments comparing this approach to an approach that does not employ user modeling techniques.

2.2 Stepwise Refinement through Clustering and Summarization (SR)

Polifroni, Chung, and Seneff (2003) developed an approach to information presentation that structures large data sets for SR. It supports the user in narrowing in on a suitable option by grouping the options that match the user's constraints into clusters of options with similar features. The system then summarizes the clusters based on the attribute values they share, thus suggesting further refinement constraints to the user (Figure 3).

This content structuring approach presents users with summaries at run time based on an algorithm that computes the most useful set of attributes, as dictated by the set of options that satisfy the current user query. For large data sets, attributes that partition the data into the minimal number of clusters should be chosen, so that a concise summary can be presented to the user to refine.[3]

---

[3] In the original implementation as reported in Polifroni, Chung, and Seneff (2003), however, the cluster attributes were specified in advance based on domain knowledge, not determined at run time based solely on the set of options returned. Our discussion and evaluation of the SR approach is therefore based on the "refined" refiner strategy from Polifroni and Walker (2008), where options are clustered based on attributes determined at run time.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

M: What restaurants do you have?

S: *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.*

M: Okay tell me about the ones in Boston.

S: *I have found 401 restaurants in Boston. There are 29 choices for cuisine.*

M: Do you have any that serve seafood?

S: *I have found 19 seafood restaurants. They are predominantly in Back Bay, the North End, South Boston and the South End.*

M: In the South End.

S: *I have found 3 restaurants. Grill Fish on Columbus Avenue, McCormick & Schmick's on Columbus Avenue and Joseph's on High Street.*

Figure 3
Dialog between simulator (M) and the Polifroni et al. (2003) system (S).

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 an SDS, it has several limitations. First, there may be long refinement paths in the dialog structure, that is, many dialog turns may be necessary to narrow in on a suitably small set of options. Because the system does not know about the user's preferences, the option clusters may contain irrelevant information which must be filtered out successively with each refinement step. In addition, the difficulty of summarizing options typically increases with their number and diversity, to the point where the summary becomes uninformative (e.g., *I found flights on 9 airlines.*).

Second, exploration of trade-offs is difficult with the SR approach in situations where there is no optimal option. If at least one option satisfies all of the user's requirements, this option can be found efficiently with the SR strategy. However, the system does not point out trade-offs among alternatives in cases where no optimal option exists. For example, in the flight booking domain, suppose the user wants a flight that is cheap and direct, but all the flights are either expensive and direct or cheap and indirect. In the SR approach, the user will have to ask for cheap flights and direct flights separately because one of these constraints must be relaxed in each case, and thus the user has to explore these refinement paths separately.

A third drawback of the SR approach is that the attribute(s) chosen for summarization may not be relevant to the user. The procedure for choosing the attributes for clustering the options is designed to select attributes that generalize well over the data (i.e., produce large clusters of options), and thus lead to efficient summarization. Hence attributes that partition the data set into a small number of clusters are preferred. If the attribute that is best for summarization is not of interest to a particular user, dialog duration is increased unnecessarily. This in turn may lead to reduced user satisfaction, as the results of our evaluation suggest (see Section 4.1.3).

3. Our Approach: User Model Based Summarize and Refine (UMSR)

Our approach, the UMSR approach first described in Demberg and Moore (2006), is intended to capture the complementary strengths of the two previous approaches. It exploits information from a user model to reduce dialog duration by selecting only options that are relevant to the user. In addition, we introduce a content structuring algorithm that supports stepwise refinement, as in Polifroni, Chung, and Seneff (2003),

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

but in which the structuring reflects the user's preferences. Thus our approach maintains the benefits of user tailoring, while also being capable of dealing with a large number of options.

We hypothesize that our approach will increase efficiency and effectiveness of the dialog and improve understandability for the user, as well as provide a better overview of the option space, ultimately leading to improved user satisfaction. We discuss these goals in more detail in the following paragraphs.

Increasing Efficiency. The integration of a user model with clustering and structuring 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 corresponding attributes are likely to be of interest to the user. The system can then select compelling options, and decide not to mention options which are likely to be irrelevant to the user, leading to shorter refinement paths, more relevant summaries, and increased efficiency.

Increasing Effectiveness. The user model also allows the system to determine trade-offs among options. For example, suppose the user wishes to book a flight, and the user model indicates that this user prefers to fly on KLM. If the database does not contain any KLM flights that also match the user's other preferences (such as preferring direct flights to connecting ones), the system can recognize this conflict and present an explicit trade-off, as in *I found a KLM flight but it requires a connection in Amsterdam. However, there is a direct flight on BMI.* The user model also allows the identification of the attribute that is most relevant at each stage in the refinement process, which is used to decide whether to present information about arrival time or price, for example.

Our hypothesis is that the explicit presentation of trade-offs enables the user to make a more informed choice and decreases the risk of the user missing out on relevant options. It thus improves the effectiveness of the spoken dialog system by helping users to select the most suitable option.

Improving Understandability. Our system strives to improve understandability of the presentation by tailoring the presentation to the user's interests. In the flight booking domain, this corresponds to clustering the available flights into ones on airlines that the user prefers versus flights on airlines that the user disprefers, or to talking about *flights that arrive by the requested time* as opposed to ones that *arrive later* than specified by the user.

Creating Confidence and Providing an Overview of the Available Options. In order to make the user feel more confident in the dialog system's option selection process, we also briefly summarize options that the user model determines to be irrelevant (see Section 3.5). By providing users with an overview of the whole option space, we reduce the risk of leaving out options the user may wish to choose in a specific situation (thus overriding her standard user model). The level of detail that the system chooses to present options which are likely to be irrelevant to the user is a trade-off between efficiency and quality of overview. If a situational user model with information about the degree of urgency is available, such overview summaries could be left out when the user is in a hurry (Komatani et al. 2003).

Increasing User Satisfaction. We hypothesize that a system that implements the features discussed here will lead to greater overall user satisfaction.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

3.1 Implementation

Our approach to information presentation was implemented within FLIGHTS, a spoken dialog system for flight booking (Moore et al. 2004). The options in the flight booking domain are flight connections with the attributes `arrival-time`, `departure-time`, `number-of-legs`, `travel-time`, `price`, `airline`, `fare-class`, and `layover-airport`. A user model contains a partial ordering of these attributes corresponding to the user's ranking, as shown in Table 2. Furthermore, the user model stores preferences (e.g., for a certain airline or flying business class). In a real-world scenario, the user model can be acquired by requiring the user to register with the system at first use (Moore et al. 2004), by building up a user model over time (Thompson, Goeker, and Langley 2004), or by classifying users into preference groups based on other information available about them, and using the group model (Rich 1979), as is frequently done in collaborative filtering. Once a user model exists, the user only needs to specify the current situational information, such as the destination, desired arrival time, and date of travel.

3.2 System Architecture

A sketch of our system's pipeline architecture focusing on the information presentation phase is given in Figure 4. In the version of our system that was used in evaluation, speech recognition and natural language understanding were performed by a wizard (see Section 6.1) who also chose from a set of canned queries during the initial information gathering phase. The first step in natural language generation (NLG) is content selection and structuring. The NLG subsystem takes as input an abstract communicative goal from the dialog manager. In the information presentation phase of the dialog, this goal is to describe the available flights that best match the user's constraints and preferences. This step is responsible for deciding what information should be communicated in the system's response, and structuring this information based on the user's query, the user model, and the set of options returned from the database. The core of this step is the algorithm for constructing and pruning the option tree, which structures all of the options that satisfy the user's query into the tree and selects the entities that should be mentioned.

The text planning step takes the pruned option tree as an input and transforms it into natural language. First, it determines how much information can be presented in one dialog turn, and how to structure the information in that turn. For example, in systems that aim to influence the user's choice, such as product recommendation systems, the ordering can be arranged to increase the effectiveness of the recommendation (Carenini and Moore 2001). There exists a full generation pipeline for this system, as described in Moore et al. (2004). However, for the experiments reported here, templates were used instead of the full generation pipeline, for reasons of robustness.

Table 2
Attribute ranking for business user.

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

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

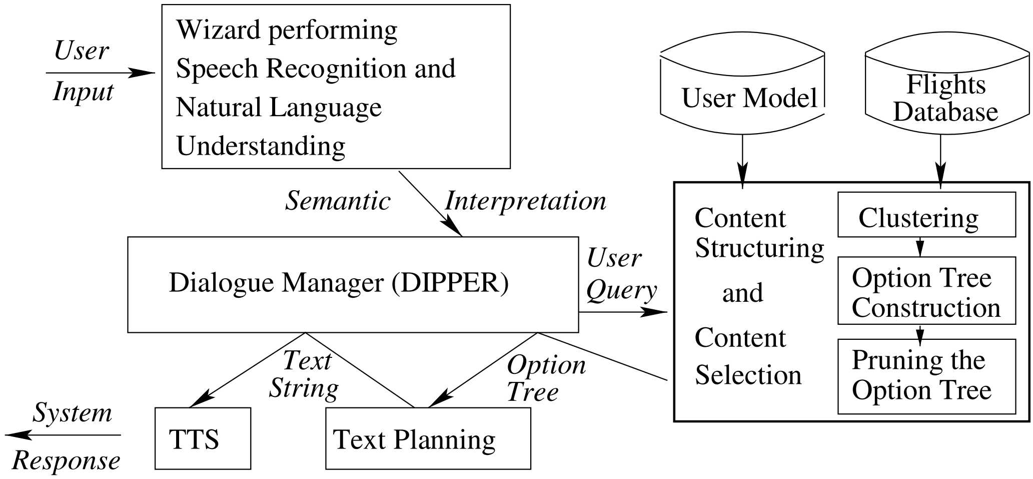


Figure 4
System architecture with emphasis on information presentation phase.

The system uses the Open Agent Architecture (OAA) framework (Martin, Cheyer, and Moran 1999) as a communication hub. All modules are implemented as agents, whose communication is managed by the DIPPER dialog manager agent (Bos et al. 2003), which invokes the different agents and stores the intermediate results from each component.

The approach proposed in this article concerns the content structuring and selection step of the system, and is a new design. It consists of three major steps: clustering, building the option tree, and pruning. The first step in our content structuring algorithm is to cluster the values of each attribute in order to group them such that labels like `cheap`, `moderate`, and `expensive` can be assigned to values of continuous categories such as `price`. This clustering means that options can also be summarized more easily later. Next, the system constructs the option tree. Each branch of the tree describes a possible refinement path and will thus direct the dialog flow. The construction of the option tree is driven by three factors: the user model, the options returned from the database, and the attribute–value clustering. The resulting option tree determines how different options relate to one another, and which ones are most attractive for the user. After the option tree structure has been built up, it is pruned based on the information from the user model, which enables the system to distinguish between options that are likely to be compelling to the user and those that are not. At this point, the content selection and structuring process is complete, and the option presentation phase follows, which consists of determining turn length and deciding on realizations for the information that is to be conveyed.

3.3 Clustering

We used agglomerative group-average clustering to automatically group values for each attribute; a similar clustering algorithm was used in Polifroni, Chung, and Seneff (2003). The algorithm begins by assigning each unique attribute value to its own cluster, and successively merging those clusters whose means are most similar. For example, Figure 5 shows the prices from six flights marked as dots on the price axis. In the first step, each flight is assigned to its own cluster (represented as a circle around the dots).

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

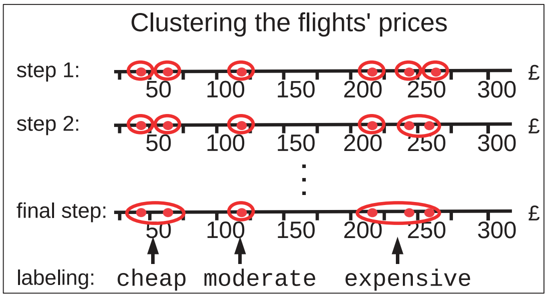


Figure 5
Agglomerative group-average clustering and labeling of options.

In the second step, the clusters of the two flights with the most similar prices (the ones close to £250, in our example) are merged. This procedure continues until a stopping criterion is met. In our implementation, we stop when we have reduced the number of clusters to three.[4] These clusters are then assigned predefined labels, for example, `cheap`, `average-price`, `expensive` for the `price` attribute. This clustering is used to group similar attribute values together and is only done once for each request (in the air travel domain, a request corresponds to one origin–destination pair on a specific date) on the basis of all database entries that satisfy the hard criteria. For further discussion of issues relating to this procedure, see Section 9. Categorical values are clustered using the user's valuation: For example, airlines are clustered into a group of `preferred` airlines, `dispreferred` airlines, and airlines the user does `not-care` about.

Clustering allows the algorithm to assess the similarity of options, namely, instead of talking about the *£51 flight* and the *£48 flight*, the system would refer to the *cheap flights*. This leads to more efficient summaries and enables the system to avoid presenting large numbers of options that are very similar in all respects. Furthermore, the clustering process enables the system to assign labels that are sensitive to the other options in the database. For example, a £300 flight is assigned 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 Amsterdam (for which there are many cheaper flights in the database).

3.4 Building the Option Tree

The tree building algorithm arranges the available options into a tree structure (Figure 6). Every branching point in the tree corresponds to a choice—for example, between economy versus business class flights. The nodes of the option tree correspond to sets of options that share a set of attribute values. The arcs going out from a node are labeled with the different attribute values. For example, in Figure 6, the root of the tree contains all options, and its left child contains all flights offering economy class tickets. The

---

[4] The choice of a maximum of three clusters as a stopping criterion is somewhat arbitrary. A clustering algorithm that automatically chooses a "natural" number of clusters given the data distribution could be used. Alternatively, domain knowledge could be employed to decide on an appropriate number of target clusters (and this number could of course be different for each attribute). However, choosing a larger number of clusters leads to bigger option trees, and thus there is a trade-off between the number of clusters and the complexity and verbosity of the summaries produced.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

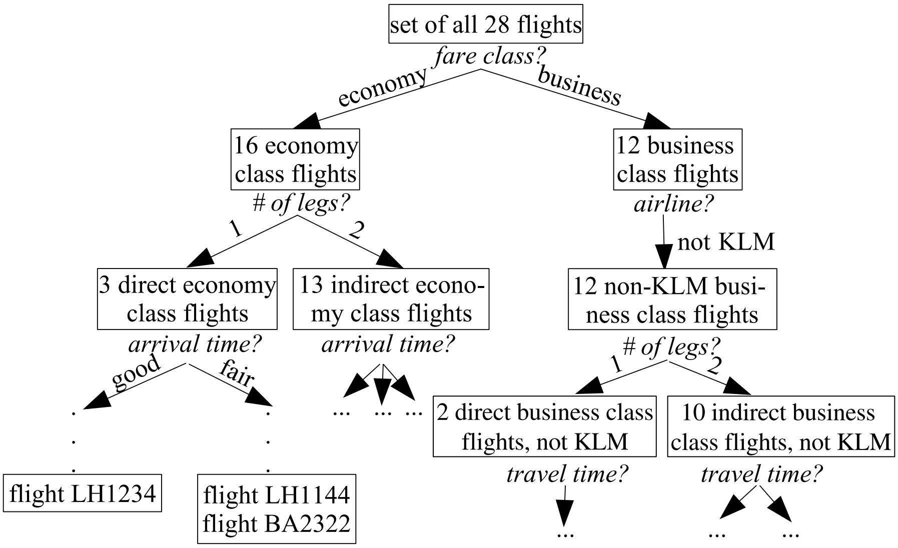


Figure 6
Option tree for business user.

children of this node contain complementary subsets of these options (i.e., all direct economy class flights vs. all indirect economy class flights). Leaf-nodes correspond either to a single flight or to a set of flights, where for each attribute of an option the values are either the same or fall within the same cluster (prices of all these flights are moderate, they all require one connection, they are all economy class, etc.). Each node can maximally have three children in our implementation, because the algorithm works on the clusters instead of the original values (e.g., it does not distinguish between similar prices, such as £48 and £52, if the clustering algorithm labeled both of them as `cheap`).

To maximize the efficiency and effectiveness of the dialog, the dialog structure is tailored to the user based on the user model. Table 2 shows the valuations of our prototypical "business user." Fare class is most important to this user, so it is ranked highest. Arrival time, number of legs, departure time, and travel time are considered next most important, and are therefore all assigned rank 2 (i.e., our algorithm does not require a total ordering of the user's preferences). The airline is next most important, and finally, price and layover airport are least important.

The user's ranking of attribute importance is crucial for dialog efficiency. If an irrelevant criterion is used as the branching criterion high up in the tree, interesting trade-offs risk being scattered across the different branches of the tree. For example, it would be suboptimal to ask a business user to make a choice about cheap versus expensive flights first, if she does not care about this aspect, as she would then have to try to identify interesting flights among both the cheap and the expensive flights. Our algorithm chooses the attribute that has the highest weight according to the user model as the branching criterion for the first level of the tree. For the business user, this would be fare class. The next decision is about the attributes that are second most important, such as the number of legs required (refer to Table 2 and Figure 6), and so on. The system therefore constructs the tree such that it presents the criteria which are most relevant for the specific user first, and leaves less relevant criteria for later in the dialog (i.e., further down in the tree). The advantage of this ordering is that it minimizes the probability that the user needs to backtrack.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

A special case occurs when an attribute is homogeneous for all options in an option set (for instance if none or all of the business class flights happen to be on the user's preferred airline). In that case, a unary node is inserted regardless of the rank of its attribute (see, for example, the right subtree with the attribute `airline`, which is inserted far up in the tree despite its low rank, in Figure 6). This special case allows for more efficient summarization, for example *None of the business class flights are on KLM*, instead of having to say this in subsequent dialog turns for each of the business class flights that the user explores.

In cases where several attributes have the same rank in the user model, we follow the approach taken in Polifroni, Chung, and Seneff (2003). The algorithm selects the attribute that partitions the data into the smallest number of sub-clusters. Consider again the tree in Figure 6: `number-of-legs` creates only two sub-clusters for the data set (`direct` and `indirect`) and is therefore further up in the tree than `arrival-time`, which splits the set of economy class flights into three subsets (`before 3 pm`, `3 pm to 5 pm`, `after 5 pm` for a user whose preferred arrival time is *by 5 pm*).

The tree-building algorithm constitutes one of the main differences between our structuring algorithm and Polifroni et al.'s (2003) refinement process. The SR system chooses the attribute that partitions the data into the smallest set of unique groups for summarization, whereas our UMSR algorithm takes the ranking of attributes in the user model into account. In the extreme case of a user who does not care about anything (the user model does not specify any valuations of any attributes over others, and indicates that the user does not care about price, whether it is a direct flight, etc.), our algorithm would end up only using the information theoretic criteria, just like the SR system.

3.5 Pruning the Tree Structure

After the tree-building step, the tree contains all the options in the database that satisfy the user's query. This tree can potentially be quite large and navigating through it would be very laborious for the user. At this point, the user model comes into play again: Because the system already knows which options are relevant to the user (and which ones are not), it can prune the option tree to retain only options that it classifies as being relevant to the user.

To determine the relevance of options, we define the notion 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 worse on at least one attribute and equal or worse in all other respects than some other option in the relevant partition of the database; it should therefore not be of interest to any rational user. When two options are equal in all respects and dominate other options, both are kept in the option tree. A similar notion of dominance was employed by Linden, Hanks, and Lesh (1997).[5] The notion of dominance is also related to the decision-theoretic concept of Pareto optimality.

Pruning dominated options is crucial to our structuring process. The algorithm uses information from the user model to prune all dominated options. Paths from the root to a given option are thereby shortened considerably and thus dialogs with our system

---

[5] In their work, dominance is used to avoid presenting options that are dominated by an option that has already been mentioned in the interaction with the user. During the user–system interaction sequences, a user makes a request and the system then presents an option. If the user then specifies or modifies his request, the system presents more options given the new specifications, but never ones that are dominated by a previously mentioned option (i.e., worse or equivalent in all respects).

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

can be expected to be on average shorter than dialogs with a system employing the SR strategy, which does not exploit information from a user model.

The pruning algorithm operates directly on the option tree, and exploits the tree structure in order to efficiently determine dominance relations. We first briefly outline the algorithm, before describing each step in detail. The first step of the algorithm is to order the tree such that the best options are leftmost.[6] The algorithm then traverses the tree in depth-first order and generates constraints during this process. These constraints encode the properties that options to the right of the current position in the tree would need to satisfy in order not to be classified as being dominated by any of the options considered so far. A branch must fulfill the constraints that apply to it, otherwise it is pruned. If an option (or a cluster of options) satisfies a constraint, the property that satisfied the constraint is marked as the options' justification. If some, but not all, of the constraints can be satisfied by an option, the constraints are propagated to the options that are further to the right in the ordered option tree. Once all the dominated options have been pruned from the option tree, there is a homogeneity check to ensure that attributes which have the same value among a set of options are annotated at a node that is a common ancestor of all of these options.

Tree Ordering. The first step of the pruning algorithm is to order the tree. This step is very important, because it imposes a total ordering on the available options and arranges them such that the best option of every node becomes that node's leftmost child. For example, the tree in Figure 6 is not ordered because the business user prefers business flights to economy flights, and thus the two subtrees under the root node must be exchanged (see Figure 7). The total ordering is enforced firstly by placing the attributes that are most relevant to the user at the top of the tree during tree construction, and secondly, by sorting the attribute values from best to worst within each node.

Constraint Generation. After ordering the tree, the globally best option is described by the leftmost branch in the option tree. In our example in Figure 7, this is flight LH1554, in node 7. If the globally best option in node 7 was perfect (i.e., if it was exactly what the user was looking for), the option in node 7 would dominate all other options, and the rest of the tree would be pruned. However, if there is an aspect of the globally best option which does not match the user's ideal, the user will have to make some kind of trade-off. This is what happens in our example, because the arrival time of the flight in node 7 was only classified as `fair`, not as `good`, whereas there exist some connections with arrival times that were classified as `good`. A flight with a good arrival time constitutes a possibly interesting alternative. In order to find such an option and filter out the others, the constraint `arrival-time:good` is generated.

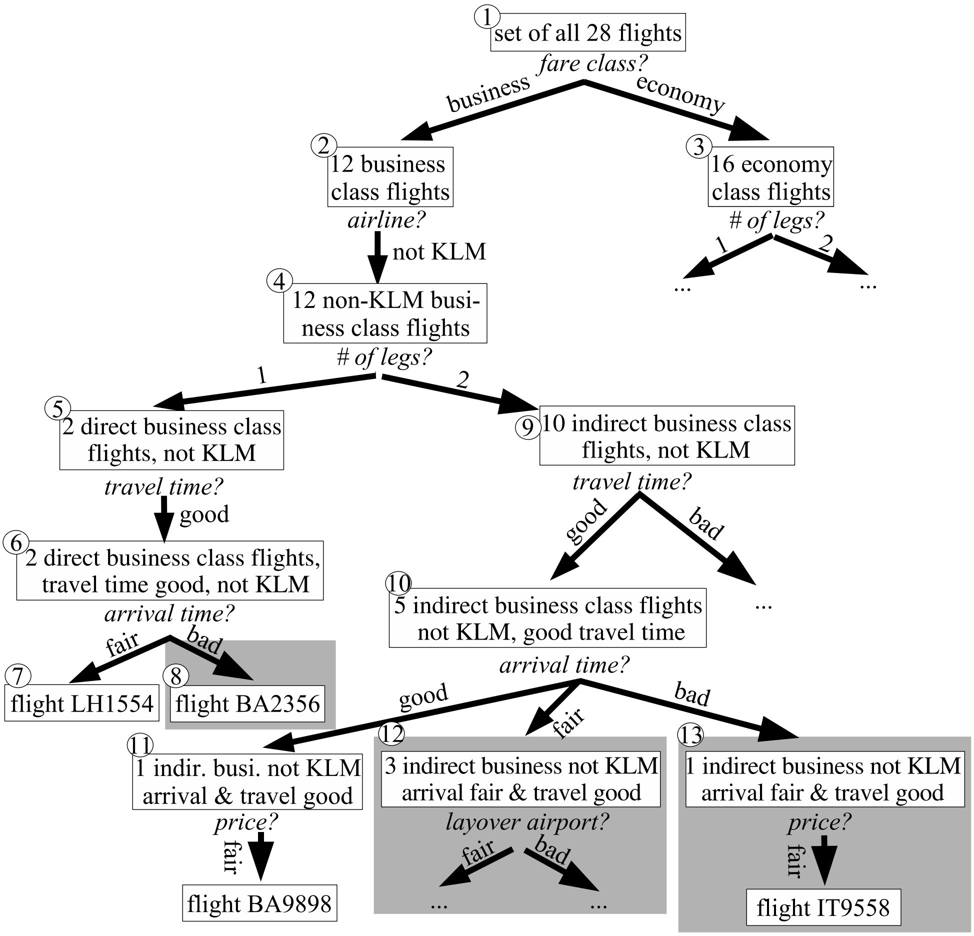
Pruning Options from the Tree. When node 8 is reached by the depth-first traversing algorithm, a constraint (`arrival-time:good`) has been generated by node 7. Node 8 does not satisfy this constraint; this means that it is dominated by node 7 and therefore is pruned from the option tree (as indicated by shading in Figure 7).

Constraint Propagation. Once the status of a node's children has been determined, any unsatisfied constraints that were generated by the child nodes are propagated to the

---

[6] Alternatively, the tree construction algorithm could be designed to insert all options such that the resulting tree is already ordered.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022



1 set of all 28 flights
fare class?
business
economy
2 12 business class flights
airline?
not KLM
3 16 economy class flights
# of legs?
1
2
...
...
4 12 non-KLM business class flights
# of legs?
1
2
5 2 direct business class flights, not KLM
travel time?
good
9 10 indirect business class flights, not KLM
travel time?
good
bad
...
6 2 direct business class flights, travel time good, not KLM
arrival time?
fair
bad
7 flight LH1554
8 flight BA2356
10 5 indirect business class flights not KLM, good travel time
arrival time?
good
fair
bad
11 1 indir. busi. not KLM arrival & travel good
price?
fair
flight BA9898
12 3 indirect business not KLM arrival fair & travel good
layover airport?
fair
bad
...
...
13 1 indirect business not KLM arrival fair & travel good
price?
fair
flight IT9558

Figure 7
This figure shows the ordered version of the option tree from Figure 6. The shaded subtrees are dominated because they do not fulfill constraints generated by nodes to the left in the tree, and are therefore pruned.

parent. In our example, the constraint generated by node 7 is propagated up to parent node 6. Because node 6 has no siblings, the constraint is again propagated up to its parent, node 5. The sibling of node 5, node 9, is then tested against the constraint `arrival-time:good`. Because there is no information about arrival time available at node 9, the constraint is passed down to its leftmost child (node 10). If that child node does not have information about arrival time, the constraint is passed down again. In our example, the constraint is passed down to node 11, and we find that this flight satisfies the constraint. We next repeat the constraint generation step. Flight BA9898 generates the constraint `price:good` because its own price is only classified as `fair`. At nodes 12 and 13, both constraints `arrival-time:good` and `price:good` have to be satisfied. However, they are not satisfied and therefore these two nodes are pruned. The depth-first traversal continues through the tree trying to find options that satisfy the constraints. When node 2 is traversed on the way up in the tree, it generates the constraint `airline:KLM`. This constraint, as well as any constraints that were generated by the subtree below it and have not yet been satisfied (in our example, the complex

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

constraint `price:good AND arrival-time:good`) are propagated to the right branch of the tree, at node 3.

Note that the constraints allow for efficient pruning: It is not necessary to look at the exact instances or properties of nodes 12 and 13 or their children. One only has to consider the properties which are relevant to the constraints because the tree is ordered. This allows us to conclude that all options in a specific subtree are dominated by the options in branches to the right of that subtree.

Justifications. An important by-product of the pruning algorithm is the identification of attributes that make an option cluster compelling with respect to alternative clusters. For example, the flights in node 11 were considered compelling because they had a better arrival time than the flight in node 7. We call such an attribute the **justification** for a cluster, as it justifies its existence—that is, it is the reason it is not pruned from the tree. Node 5 in turn is kept in the tree because it is the leftmost child, which means that its attribute values best match the user's preferences. When compared to the flights in node 9, its compelling property is that it is direct (i.e., `number-of-legs=1`). The default justification for a node is the attribute value on which the branch is based (e.g., `fare-class` for node 2 in Figure 7). This justification is used for nodes on the leftmost branch. Justifications are used by the generation algorithm to present trade-offs between alternative options explicitly (see Section 3.6.2).

The reasons why options have been pruned from the tree are also registered. These reasons contain information about which constraints the options failed to satisfy; in our example, the flight in node 8 is deleted because of its `bad` arrival time. These pruning reasons are later used to provide information for the summarization of poor options in order to give the user a better overview of the option space (e.g., *All other flights arrive too late or are more expensive*). 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.

Homogeneity Check. After deleting branches from the option tree, it may be the case that several options have the same attribute value, but are located in different branches in the tree. For example, imagine there are three economy class flights, two direct ones (1 leg), and one which requires a connection (2 legs). Among the two direct ones, one has a good price, and the other one is more expensive. The 2-leg flight also has a good price. If the more expensive direct flight is pruned, both of the remaining options have a good price, and thus this property should be above the `number-of-legs` branching level in the tree. This is important for efficient information presentation and summarization of options.

3.6 Option Presentation

The user model also comes into play when determining the wording of the option presentation. Because the system has a model of the user's preferences, it can effectively compare and contrast alternatives by highlighting compelling aspects of an option (e.g., a *direct* flight, the KLM flight), by using intonation and comparatives (e.g., the *cheapest* flight, the *only* KLM flight) and by acknowledging drawbacks through the use of discourse markers (e.g., *but*, *however*, *although*) when generating descriptions of options. For the options that were considered unattractive for the particular user, the system can provide an overview to cover the option space (e.g., *All other flights arrive later than 3 pm*).

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

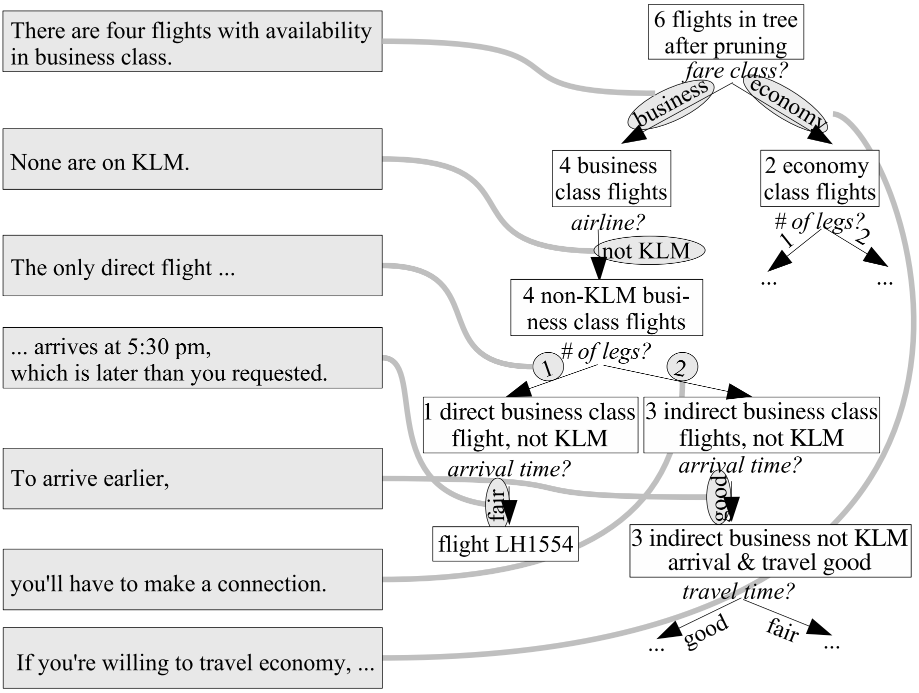

There are four flights with availability in business class.
None are on KLM.
The only direct flight ...
... arrives at 5:30 pm, which is later than you requested.
To arrive earlier,
you'll have to make a connection.
If you're willing to travel economy, ...
6 flights in tree after pruning
fare class?
business
economy
4 business class flights
2 economy class flights
airline?
not KLM
# of legs?
1
2
...
...
4 non-KLM business class flights
# of legs?
1
2
1 direct business class flight, not KLM
3 indirect business class flights, not KLM
arrival time?
fair
flight LH1554
arrival time?
good
3 indirect business not KLM arrival & travel good
travel time?
good
fair
...
...

Figure 8
Diagram showing how the pruned option tree is mapped onto language. The tree on the right hand side corresponds to the example trees in Figures 6 and 7. The complete system utterance is shown in Figure 10.

Figure 8 shows how the nodes in the pruned option tree translate to the system's utterances. The different design decisions underlying sentence planning and realization will be explained in the following sections.

3.6.1 Turn Length. In any spoken dialog system, it is important not to present too much information in a single turn in order to keep the memory load on the user manageable (Seneff 2002). Thus, our system aims at presenting no more than two or maximally three options at once. However, the pruned option tree sometimes contains more than this critical number of options, and therefore needs to be broken down into smaller chunks. We thus divide the pruned option tree into several smaller dialog-turn-sized subtrees. Typically not all of these subtrees will be presented, but only the ones between the root of the tree and the chosen subset of flights that the user wishes to hear more about.

In addition to determining the number of options to present in a single turn, the system must decide how many and which of their attributes to mention. Arguably, mentioning too many attributes of options will also lead to memory overload, which may ultimately reduce user satisfaction. However, the system must provide enough information to fully account for what constitutes the trade-off, that is, it must give the reasons why an option is potentially relevant.

For instance, in our example, the system mentions that the direct business class flight arrives later than requested and contrasts this against another business class flight that arrives earlier but requires a connection. The pruning process provided the system with information about the relevant differences between alternative options (`arrival-time` and `number-of-legs`).

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

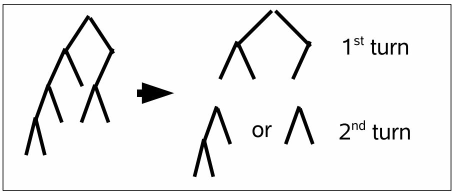


Figure 9
The option tree is cut into subtrees which determine turn length.

U:	Hello. I'd like to book a flight from Edinburgh to Brussels for tomorrow arriving at around 3 pm.
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 3:30 pm.* *There's a cheaper flight on Lufthansa. It arrives at 3:10 pm 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 10
Example dialog with our WoZ System based on the tree shown in Figure 7 and mapping to natural language sentences as shown in Figure 8.

In order to segment the pruned tree into turn-sized subtrees, we chose a very simple heuristic segmentation algorithm. Dialogs generated with this heuristic were evaluated in our early experiments,[7] which we report in Sections 4.1 and 4.2. The heuristic cutoff point is visualized in Figure 9, and defined as being no deeper than two branching nodes[8] and their children.

This heuristic produces a limited set of options to be presented in a single turn. The target size is two to three options. This strategy yields a maximum of nine options (three options per branching level to the power of two branching levels). However, in practice there are typically three or fewer options in any two branching levels left after pruning. We chose to include two layers in order to allow for informative trade-offs: If information from only one layer were available at a time, it would not be possible to contrast the most relevant advantages and disadvantages of alternative options, which is needed to make the trade-off(s) explicit. For example, if only the first level of Figure 9 were to be presented, the system could only talk about fare classes, and would not be able to point out that there is a disadvantage with the business class flights, which the economy class flights do not have.

At the end of the turn, the user is expected to make a choice indicating which of the options she would like to hear more about (for illustration see Figure 10).

---

[7] Later experiments used a more sophisticated way of determining turn length, which we describe in Section 5.

[8] Branching nodes as opposed to unary nodes. For example, in Figure 6, the unary node in the right subtree would not count as a separate level.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

This definition of turn length was employed in the reading and overhearer experiments (see Sections 4.1 and 4.2). However, it became clear in related experiments (Winterboer et al. 2007) that this method of deciding on dialog turn length sometimes led to very long system turns, which were then difficult for the user to remember. We thus modified the algorithm in order to ease comprehension. We will return to this point later, in Section 5.

3.6.2 Referring to Sets of Options. Each branch in the pruned tree corresponds to a set of options. These options should be referred to in an effective way. We do this by taking into account both the dialog structure (i.e., structure of the argumentation) and the user's interest: The description of a set of options is based on their justification. For example, the justification of the flights in the left branch of the tree in Figure 8 is their fare class. Therefore, they are described as flights *with availability in business class*. On the other hand, the justification for the indirect business class flights is that they have an arrival time that matches the user query better. They are thus referred to by their justification *to arrive earlier*.

If a node is justified by several attributes, only one of them is selected for reference. If one of these multiple justifications is a contextually salient attribute, this one is preferred over the justifications that are not salient. For example, if a node is justified both by its arrival time and its price, it would be referred to by the price attribute in a context that just mentioned the price of another flight as being expensive:

$$[\ldots \textit{but it costs 1000 pounds.}]_{\textit{context}}\ \textit{A}\ [\textit{cheaper}]_{\textit{salient}}\ \textit{flight}\ldots$$

If none of the attributes are particularly salient, the options in the cluster are referred to by the highest ranked attribute, that is, arrival time in the example from Figure 8. In cases where all options in a set only share a common negative attribute, we acknowledge this situation using a concessive formulation: for example, *If you're willing to travel economy...*; see the last sentence in Figure 8.

3.6.3 Presenting Additional Attributes to Explain Trade-Offs. In order to present trade-offs between options, it is necessary to provide information about the properties of options that constitute the trade-off(s). Any of these additional properties that are not already mentioned as part of the referring expression are ordered to optimize coherence. First, all positive attributes are enumerated and contrasted against all average or negative attributes. These negative attributes, which are presented last, are then salient and will be used in the description of an alternative option.

3.6.4 Summarizing Properties of Options. When describing a set of flights that are in the same cluster (e.g., because they have `good` arrival times), the specific attribute values of the options in this cluster may vary (e.g., one flight might arrive at 3 pm and the other at 3:30 pm). In that case, it is necessary to generate a summarizing expression for these attribute values.

There are three main cases to be distinguished:

1. The continuous values for attributes such as `price` and `arrival-time` must 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*) or that contrasts them with some previously mentioned value (e.g., *earlier* or *the cheapest flights*).

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

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 the 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 such as *none/all/both of them*, as done in Polifroni, Chung, and Seneff (2003). If the values are more diverse, the user model comes back into play to produce a tailored summary based on user preferences. For a user who prefers to fly on KLM and a cluster which contains no KLM flights, as in Figure 8, we generate *None are on KLM*, which takes into account the user's preference for KLM and is more concise than mentioning all airline names.

A sample dialog highlighting these realizations is shown in Figure 10. The turns marked "S" in this dialog were generated by our system, employing the business user model (see Figure 2). Note, however, that interpreting the user's request (i.e., speech recognition and natural language interpretation) was done by a wizard in our experiments.

3.7 Adaptability to Other Domains

To adapt our system to assist users finding options in another domain (e.g., restaurants, digital cameras), the following steps would have to be taken:

1. Creation of a domain model: Identify entity attributes and the values they can take.
2. Adaptation of the generation pipeline: Although sentence structures for presenting trade-offs between options would remain very similar, a domain-specific vocabulary would have to be created.
3. Acquisition of user models: Accomplished by explicit questioning of users or automatic inference as in, for example, Linden, Hanks, and Lesh (1997).

To apply the clustering algorithm in another domain, it is necessary to define a function that maps attribute values onto numeric values. This can be done by either acquiring the user's preferences, or by choosing a distance function that describes the relation between values of an attribute.

For the air travel domain, we used three non-relaxable constraints (i.e., constraints for which we do not use a distance function but only accept options that fully match these requirements): departure airport, arrival airport, and travel date. If our database had contained distances between airports, our system would have been able to suggest flights that start from or arrive at a nearby airport (e.g., departing from Glasgow instead of Edinburgh). Similarly, in a restaurant recommendation system with information about streets, an external resource could be used to determine which streets are in the same neighborhood, in order to cluster them according to distance or the time it takes to travel to them using a specific means of transport.

To effectively summarize categorical data, it is often necessary to use an ontology, or some kind of ordering that is meaningful for the user. Consider for example our clustering of airlines: There is no information in our database that would make a KLM flight more similar to a Lufthansa flight than to a RyanAir flight based on the airline.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Therefore, we use information from the user model to arrange options into clusters that are meaningful to the user by grouping together airlines that the user prefers to use, and those airlines that the user prefers not to use. Finally, when clusters have been generated, a set of labels must be defined for the resulting clusters.

4. Evaluation I – Two Pilot Experiments

We conducted three experiments that compared our approach to information presentation, employing content structuring and a user model (UMSR), to a system that uses only content structuring (SR). The experiments were designed to test our hypotheses that the UMSR approach increases user satisfaction, as well as dialog efficiency and effectiveness when compared to the SR approach. In this section, we discuss the first two experiments, during which our participants read or overheard dialogs, as opposed to our second evaluation phase during which they interacted with the system in a Wizard-of-Oz set-up. Although the user's turns were scripted, the SR and UMSR system turns were generated by implemented information presentation components in these two experiments.

4.1 Reading Task Experiment

In the first experiment, participants were asked to read a number of dialogs, some of which were designed according to the description of the SR algorithm and some of which were generated using our UMSR algorithm. Participants were thus in the position of external observers, and could not interact with the system.

4.1.1 Experimental Design. The experiment was performed with 38 participants, mostly students of the University of Edinburgh, who were either native speakers of English or had a near-native proficiency level in English. They were naive with respect to dialog systems and the purpose of the experiment. The experiment was conducted in the lab under controlled conditions. Each subject was presented with six dialog pairs. The first dialog pair was used for training, and was thus not included in the analysis. Each dialog pair consisted of one dialog with the UMSR system and one dialog designed following the SR approach (see Section 2.2). The order of the dialogs in a pair was randomized to prevent the risk of favoring a system due to a constant ordering (in particular, because the content of the database is already known when reading the second dialog, this dialog would be easier to understand).

The user models (a student, a frequent flyer, and a business traveler) were designed to be highly prototypical (see Figure 11) to make it easier for the participants to memorize the characteristics of the users.

Before each dialog, the hypothetical users who appeared in the dialogs were described briefly in order to communicate their preferences to the participants and put them in a position to assess whether the hypothetical users were able to find the flight that was optimal with respect to their specifications and preferences.

The dialogs with the SR system were designed manually based on the description in Polifroni, Chung, and Seneff (2003). All dialogs were provided as transcripts.

4.1.2 Evaluation Criteria. The order of the dialogs in a pair was randomized. After reading each dialog transcript, participants were asked to provide feedback about system responses on four Likert-type scales (see Figure 12). After reading each pair of dialogs, the participants were also asked the forced choice question: "Which of the two systems would you recommend to a friend?" to assess user satisfaction.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

the business traveler	He wants, above all, to travel in business class. A short travel time and good match to his desired arrival time are important to him. Furthermore, he has a preference for KLM over other airlines, see also Table 2.
the student	He cares most about price, everything else is equal.
the frequent flier	She collects business miles on KLM and therefore cares most about airline.

Figure 11
Descriptive versions of example user models used in our experiments.

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 X 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 that the system did not tell X 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.
4. How quickly did the system allow X to find the optimal flight?
   1: slowly
   3: quickly

Figure 12
The set of questions that users were asked after each dialog in order to evaluate the system. Participants provided their answers using Likert-type scales. X was instantiated by the name of our hypothetical example users.

4.1.3 Results. User Satisfaction. A significant preference for the UMSR system was observed. From a total of 190 forced choices in the experiment (38 participants × 5 dialog pairs), UMSR was preferred 120 times ($\approx 63\%$), whereas SR was preferred only 70 times ($\approx 37\%$). This difference is highly significant ($p < 0.001$) according to a two-tailed binomial test. Thus, the null-hypothesis that the systems are preferred equally often can be rejected with high confidence.

Overall preference for a system is likely to be the product of many different factors. The defining differences between the two systems are tailoring to the user first by presenting only options that are of some relevance to the user, second by explicitly comparing options to one another by pointing out trade-offs, and third by briefly summarizing the full option space. The goal of the experiment is to elicit the individual contribution of these factors to user satisfaction by asking the questions in Figure 12. Figure 13 shows the average Likert scores for the UMSR and SR systems provided by the users, and will be discussed in the following paragraphs.

Understandability. The UMSR and SR dialogs received almost identical scores for understandability (Question 1 in Figure 12). There is no statistically significant difference between the scores of the two systems ($t(189) = -0.049, p = 0.97$, using a two-tailed

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

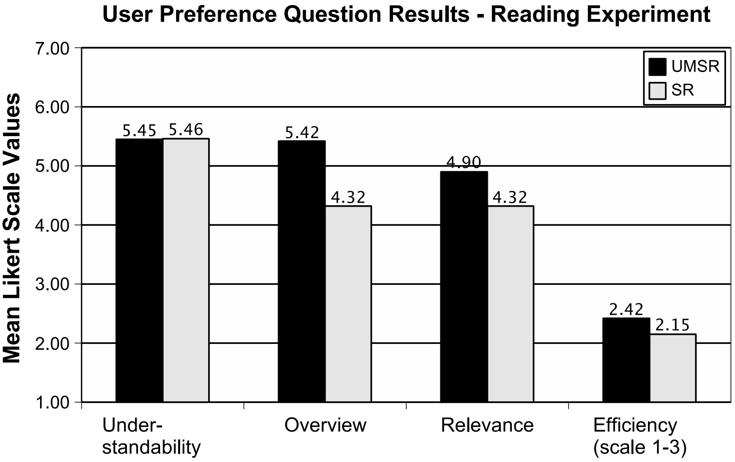


Figure 13
Results for Likert-type scale questions (see Figure 12) from the reading experiment.

paired t-test). On the one hand, understandability is likely to be enhanced by UMSR because it provides explicit trade-offs, but on the other hand UMSR produces longer and more complex dialog turns which may hinder understandability. We therefore control for dialog turn length between the two conditions in our later experiments; see Section 5.

Overview. The evaluation confirmed our hypothesis that the UMSR system provides a better overview of the option space than the SR system. The difference in ratings, more than one Likert scale point, was statistically highly significant ($t(189) = 7.47, p < 0.0001$, using a paired t-test); see Figure 13. The better overview achieved by the UMSR system is attributable to the explicit trade-offs, comparative qualifications of options (e.g., *the cheapest flight*), and summaries about dominated options (e.g., *all other flights arrive later than you specified*).

Relevance. The third question assessed the participant's confidence that all relevant options were mentioned by the system. The UMSR system was again rated significantly higher on this question in comparison to the SR system ($t(189) = 2.88, p < 0.01$), according to a two-tailed paired t-test. Presenting only options that are deemed to be relevant by the system as well as providing summaries about the whole option space are likely to have led to the observed higher confidence ratings.

Efficiency. Question 4 assessed the efficiency with which the hypothetical user was able to access the flight that best matched her interests. Again, the UMSR system was rated significantly higher ($t(189) = 3.77, p < 0.001$ in two-tailed paired t-test) than the SR system. Surprisingly, subjects reported that they felt they could access options more quickly with a system employing the UMSR strategy even though the dialog turns were longer on average.

4.2 Overhearer Experiment

A potential criticism of the reading experiment is that reading dialogs is an artificially simplified task because participants can read at their own pace and go back in the text to

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

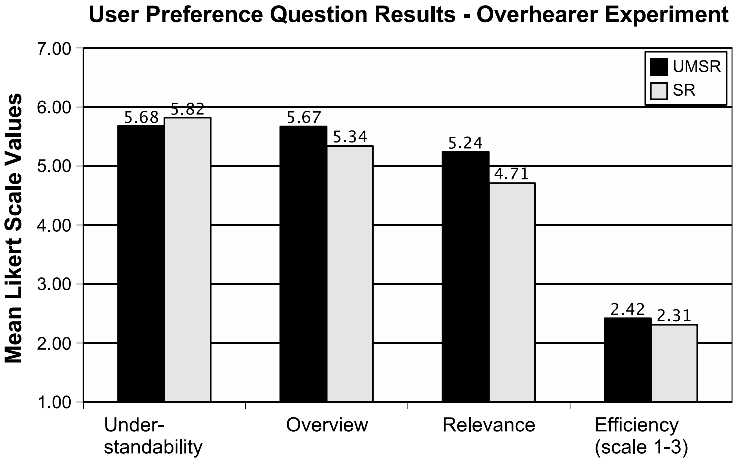


Figure 14
Results for Likert-type scale questions, overhearer experiment.

read difficult parts again. Furthermore, there is no extra difficulty caused by imperfect speech synthesis. The goal of this second experiment was therefore to quantify any facilitating effect introduced through the reading modality, and to compare and validate results.

4.2.1 Experimental Design. For our second lab experiment, we recruited 40 participants, who were mainly Edinburgh university students, native or near-native speakers of English, and were studying for a range of different degrees. They had not taken part in the previous experiment and were naive with respect to the task. The participants were presented with exactly the same dialog pairs as in the first experiment, and were asked the identical set of questions (see Section 4.1.2). The only difference was that this time, participants overheard the dialogs instead of reading them: We used speech recordings from three different speakers to impersonate our three users—the student, the frequent flyer, and the business traveler. System turns were produced by a text-to-speech system provided by CereProc Ltd.,[9] using a voice with a Scottish accent ("Heather").

4.2.2 Results. User Satisfaction. Results for overall user satisfaction were consistent with the results from the reading experiment: Participants chose the UMSR system more often than the SR system when asked which system they would recommend to a friend. However, the difference did not reach significance level in the overhearer condition (UMSR was preferred 109 times out of 200 times). The more detailed analysis based on the Likert-type scale questions (Figure 14) shows that while ratings tended to be higher in general, the UMSR system benefited less from the oral presentation mode than the SR system.

Understandability. We specifically asked participants not to assess the systems on the speech synthesis quality but rather on the structure and content of the presented material. The SR system was rated slightly higher on the understandability of the dialogs in comparison with UMSR. The observed slight preference for the SR system did not reach significance level ($t(199) = -1.17, p \approx 0.24$ in a two-tailed paired t-test), however.

---

[9] `http://www.cereproc.com/`.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Overview and Relevance. The second experiment confirmed the findings from the reading task experiment: Users thought that the UMSR system gave them a better overview of the option space (Question 2). In addition, they felt more confident that they had not missed out on any relevant options (Question 3). The differences in ratings were again highly statistically significant in both cases ($\mathrm{t}(199) = 2.61, \mathrm{p} < 0.01; \mathrm{t}(199) = 3.11, \mathrm{p} < 0.01$), just as in the reading experiment.

Efficiency. The UMSR system was rated better than the SR system in terms of how quickly the hypothetical user was able to access the optimal flight (Question 4). However, this difference was smaller in the overhearer experiment, and did not reach significance level ($\mathrm{t}(199) = 1.65, \mathrm{p} \approx 0.1$).

4.2.3 Discussion. All the tendencies in ratings for experiments 1 and 2 were the same, the only differences being that effect sizes were slightly smaller in the overhearer condition (see Figure 14) and that participants rated the systems consistently higher in the overhearer experiment. We hypothesize that these higher scores across the board must be due to the presentation mode, that is, participants were possibly more impressed with the systems when they overheard the interaction as opposed to reading it, because it was closer to experiencing a working system.

Although these two experiments provide valuable support for our hypotheses concerning overview and relevance, and indicate better perceived efficiency of the system, we are ultimately interested in how people cope when *interacting* with the different types of dialog systems, and how user modeling affects task efficiency and effectiveness. We therefore designed a third experiment, in which participants were assigned flight booking tasks and directly interacted with the dialog system (see Section 6).

Because UMSR employs information from a user model and content structuring to present only those options that are likely to be relevant to the user, and explicitly points out trade-offs among options, we hypothesized that it would also consume less cognitive resources, that is, presentations based on UMSR would impose less cognitive load than those based on SR. Therefore, in parallel to our work on the interaction task experiment, a companion project conducted experiments using a dual-task setting (driving a simulated car while using the dialog system to book flights) to investigate the effects of the two strategies to information presentation (UMSR vs. SR) on cognitive load (Hu et al. 2007). Results indicated that, contrary to expectations, the cognitive load placed on users by the UMSR system was in some cases higher than the cognitive load placed on users in the SR condition, and that this affected performance on the primary task (driving). By examining logs of the conversations, we found that the UMSR approach sometimes generated turns that were considerably longer than turns produced by the SR approach. Because we are primarily interested in whether the content selection and structuring aspects of UMSR led to its benefits, we removed the confounding factor of turn length by controlling for turn length across the two conditions in our third experiment.

5. Controlling Dialog Turn Length

Determining how much information to include in a dialog turn is an important factor in designing information presentation strategies. The results from Hu et al. (2007) suggested that our initial UMSR algorithm for determining how much information to include in a turn needed improvement. In addition, controlling for dialog turn length and

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

the number of information units conveyed in each dialog turn is important for making a fair comparison between the aspects of information presentation we are interested in, that is, content selection (which attributes are included in a summary) and structuring. Presenting much complex information in a single turn can cause a significant memory burden on the user, and thus disturb the variables we want to measure. Furthermore, we hypothesize that a large memory burden will not only overwhelm the user, and thus affect user satisfaction, but can also cause a degradation in efficiency (because users may have to ask for repetition or clarification of information) and/or effectiveness (because users may forget pieces of information that are critical for making the best choice).

Dialog turn length in our system is defined by the algorithm that divides the tree into separate chunks, as explained in Section 3.6.1. The original naive heuristic can sometimes cause long system turns if the branching factor of an option tree is large even after the pruning step. In order to balance message length between the two conditions, we modified the content selection method to control the amount of information presented in each turn, as follows:

- The strategy for determining turn length (see Section 3.6.1) was revised so that the initial option tree is divided into smaller chunks. Recall that the tree was cut after a maximum of two branching nodes and their corresponding children in the original implementation. In the new implementation, the tree is now split into smaller trees if a threshold number of attribute values conveyed is exceeded.
- Fewer attributes are mentioned per presentation (for instance, by using a very general formulation *that will get you there on time* instead of *that arrives between 3 pm and 6 pm* once the user had specified a specific time she wished to arrive, because this mentions a smaller number of facts).
- The generation component was modified so that a maximum of four sentences were presented per turn.
- Information that can be inferred from other pieces of information that have already been presented (e.g., travel time, once departure and arrival time are known) was left out to minimize the number of pieces of information that have to be remembered.
- Presenting the details of more than two flights in one system turn was generally considered not manageable for a user, because each flight presentation must contain at least the airline name, the arrival time, the travel time, and the number of legs. If there are more than two flights in a cluster, we only present attributes that distinguish the flights (e.g., *The three direct flights are on Continental, Lufthansa, and Delta. They arrive at 9:55 am, 10:15 am, and 11:15 am*).

Finally, we compared the dialogs for the SR and UMSR approaches to make sure that at each step, the turn length and information density would be roughly the same for both conditions.

6. Evaluation II – User Interaction Study

Due to the complexity of building a working end-to-end SDS, our previous experiments employed an "overhearer" evaluation methodology, in which participants read

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

or listened to pre-recorded dialogs with a scripted user. This limits the evaluation criteria that can be applied to users' perceptions (e.g., understandability, goodness of overview of options, and so on). Our third experiment, a short summary of which was first reported in Winterboer and Moore (2007), compares the two systems in a more interesting setting, in which users actively interact with the dialog systems. This setting allows us to measure the systems' impact on effectiveness of the interaction (as measured by task success) and efficiency (as measured by dialog duration), in addition to user satisfaction. In this experiment, we compare the revised UMSR approach, which controls the amount of information presented in each turn, to the SR approach as described previously.

6.1 The Wizard-of-Oz Paradigm and Experimental Environment

The interaction task experiment presented here followed the Wizard-of-Oz (WoZ) paradigm (Dahlbäck, Jönsson, and Ahrenberg 1993) which enables us to test hypotheses about not yet implemented or not sufficiently robust systems by simulating them. The systems used in this experiment have fully implemented information presentation components, including content selection and structuring, text planning, and template-based realization. The wizard's role was limited to language understanding, making sure all obligatory slots (e.g., departure airport, destination airport, travel dates) were filled, and keeping the dialog going if the user was silent.

We used a database-driven Web interface which generated system responses on-the-fly based on either the SR or the UMSR strategy to presenting information. The wizard sat on the opposite side of the room, hidden behind a partition that prevented participants from seeing or hearing the wizard during the experiment. The integrated SQL-based database system contained actual flight information as provided by airlines. The wizard used drop-down menus to perform stepwise queries according to a participant's requests until a satisfying flight was found and booked. The generated textual information provided by the Web interface was copied-and-pasted to Speechify, a text-to-speech (TTS) application provided by Nuance Communications, Inc. All participants heard a synthetic voice of their own gender speaking British English. This choice was motivated by the work of Nass and Brave (2005), which shows that the gender of the voice used for TTS plays a significant role in terms of user preferences and trust in the information provided (users prefer a voice that matches their gender). Participants were encouraged to speak naturally rather than merely to respond to system prompts. The wizard used very few questions as prompts (e.g., *What would you like me to book for you?*) and would only add additional questions (e.g., *Do you have a preferred arrival time?*) if the participant remained silent for more than five seconds after a round of information presentation by the system.

6.2 Experimental Design

A total of 34 participants were paid to participate in the experiment. About half of participants were students of the University of Edinburgh (but not with a computer science background), the other half were recruited via a Web site for one-off jobs. The average age of the 17 female and 17 male participants was 24 years. All participants were naive to the purpose of the experiment and native English speakers.

Each participant was asked to book four flights. In order to enable reliable and rigorous comparisons, all participants were briefed to act as a business traveler for the flight booking task. As described in Section 4.1.1, the business traveler, in descending

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

order of importance, (1) prefers flying `business class`, (2) is concerned about `arrival time`, `travel time`, and `number of legs`, and (3) wants to fly on `KLM` if possible.

After being briefed about the role they were to play, participants were assigned the task of booking flights and received detailed instructions concerning the two flights to be booked in the first part of the experiment. To make the booking process more realistic, the four routes (i.e., pairs of cities) were carefully chosen in order to guarantee that each participant experienced four different scenarios:

1. no KLM flight was available
2. one KLM flight matched all the criteria
3. one KLM flight in business class was available but required a connection
4. one KLM flight was found but it was in economy class

Both the order in which the four flights were booked and the order of the information presentation strategies (UMSR and SR) were randomized to counter-balance possible order effects. For the first two dialogs, half of the participants obtained flight information presented from the system adopting the SR approach; the other half received search results as determined by the UMSR approach. In the second part of the experiment, participants who received flight information with SR in the first half of the experiment were provided with information based on UMSR and vice versa. After completing the dialogs with a specific system (i.e., after the second and fourth dialog), participants were asked to fill in a questionnaire to provide judgments on the four user satisfaction questions (see Figure 12).

6.3 Results

Dialog recordings and questionnaire answers were analyzed. For the questionnaire data, we used seven-point Likert-type scales.

Efficiency. There was a highly significant difference in the average number of turns participants required for booking a flight when the system adopted the SR approach as compared to the system adopting the UMSR approach. Using a two-tailed paired t-test we found that participants interacting with the UMSR-based system took significantly fewer turns than when using the SR-based system ($t(31) = -5.57; p < 0.0001$; indicated with ** in Table 3).

Moreover, a two-tailed paired t-test revealed that dialog duration was significantly shorter for bookings made with presentations based on UMSR as opposed to SR ($t(29) = -6.39; p < 0.0001$; Table 3).

Table 3
Number of turns per booking and average dialog duration for two bookings (in seconds) for SR and UMSR.

	SR	UMSR
Turns	14.53**	10.53**
Duration (sec)	391.65**	252.55**

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Effectiveness. We hypothesized that UMSR, which explicitly points out trade-offs among options, would lead to improved task success. To this end, we extracted information about flights (e.g., flight times, airlines, number of layovers) from `http://www.expedia.co.uk` and chose our scenarios carefully to ensure that there was one flight that best matched the requirements we presented to participants. We then counted how often the flight best matching the business traveler's profile was chosen in each condition. Using the UMSR-based system, participants succeeded in booking the best-matching flight significantly more often. They booked the optimal flight 62 times out of 68 (91.18% success rate) with the UMSR system, but they only succeeded in doing so 50 out of 68 times (73.53% success rate) when interacting with the SR system.

User Satisfaction Ratings. No statistically significant differences in terms of system preference were found between genders. In the questionnaire data (see Figure 15) we found a general preference for UMSR-based recommendations on all four evaluation criteria. However, only differences between answers to Questions 1 and 4 ("Did the system give the information in a way that was easy to understand?", $t(33) = -2.85; p < 0.05$; "How quickly did the system allow the user to find the optimal flight?", $t(33) = -2.98; p < 0.005$) were statistically significant. We believe that the lack of a significant difference between systems on Questions 2 (overview) and 3 (relevance) is related to the fact that the new version of UMSR controlled for turn length. In the revised version, information about alternative or competing options was only included if the turn length allowed for it. Thus, there is trade-off between understandability and presenting information that is intended to improve the user's overview of the option space. Further investigation of this trade-off should be carried out as part of future research.

6.4 Limitations on Evaluation

Although the WoZ method allows us to evaluate our information presentation strategy in the context of an actual interaction, there are still a number of shortcomings. An ideal user evaluation would use actual automatic speech recognition and language

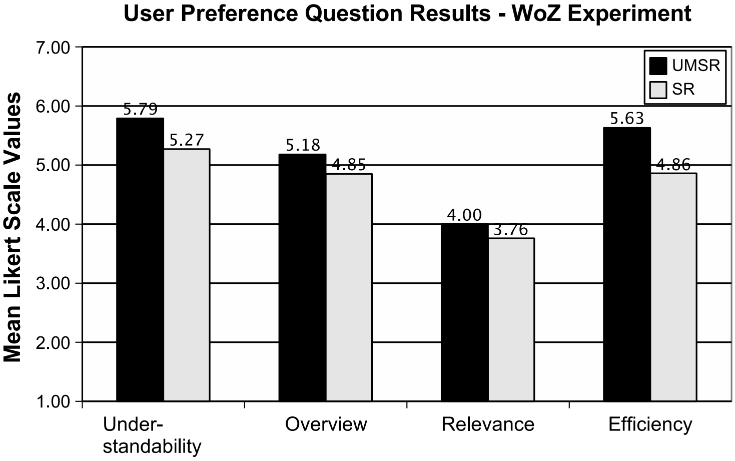


Figure 15
Results for all questions, interaction experiment.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

understanding instead of the wizard performing these actions. In addition, it would use the actual model of each user rather than ask them to role-play.

Although role-playing is an often used methodology in SDS evaluations (Polifroni and Walker 2008) and corpus collection (Carletta 2007), we recognize that there are concerns with respect to assigning subjects a user model as opposed to using the participants' own models. Possible problems with role-playing include the fact that participants may have incorrect expectations about the needs and preferences of the role they are supposed to play, and that they might not adopt the role sufficiently. Furthermore, a scenario where participants are role-playing cannot test how good the user model is, or how robust a user-adaptation system is.

We chose role-playing as a method despite these concerns because it allows us to more closely control conditions and guarantee that every participant will be exposed to the same number and constellation of trade-offs (see Section 6.2).

In future work, we wish to perform an evaluation where the user's own model is used.

7. Discussion

We ran over 100 participants in three experiments and obtained very stable results in favor of the UMSR approach to information presentation. In this section, we summarize our findings from the three experiments with respect to our initially stated hypotheses.

Did the UMSR Approach Increase the Efficiency of Information Presentation? Overall, both in terms of the subjectively perceived efficiency, and in terms of the objectively measurable efficiency, the UMSR system significantly outperformed the SR system. Task duration was significantly lower in the UMSR condition than in the SR condition in the interaction experiment, with the UMSR system also requiring fewer dialog turns. Furthermore, participants reported that they felt the optimal option was accessed more quickly in the UMSR condition than in the SR condition in all three experiments.

Did the UMSR Approach Make Information Presentation More Effective? This hypothesis was tested in the interaction task experiment by measuring task success. We found that users chose the option that was best for the role they were playing more often in the interaction experiment. These findings also agree with results obtained in other work, using a dual-task method (Hu et al. 2007; Winterboer et al. 2007). Thus, we conclude that information access with the UMSR approach is more effective than with the SR approach.

Did UMSR Provide a Better Overview of the Option Space and Was Higher Confidence Achieved with UMSR? Questions 2 ("Did the system give X a good overview of the available options?") and 3 ("Do you think there may be flights that are better options for X that the system did not tell X about?") of the questionnaire were designed to assess this aspect. The first two experiments showed a clear preference for the UMSR system in comparison with the SR system. Participants in the interaction experiment also exhibited a preference for the UMSR system, but the difference in ratings did not reach significance level. The smaller effect size in the third experiment is likely due to controlling dialog length: The overview of options deemed uninteresting to the user was sometimes omitted to reduce turn length. This omission seems to negatively affect the user's overview of the option space and their confidence in having heard about all relevant options.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Was the Understandability of the System Improved? Understandability was the single aspect in which the SR system was not outperformed by the UMSR system for the first two experiments (differences in ratings were marginal). Our hypothesis had been that the user model improves understandability by tailoring the presentation of the options to the user. This seemed to be outweighed by the larger amount of information presented in the original implementation of UMSR, however. In the interaction task experiment, in which dialog turn length was kept constant across the SR and UMSR conditions, turns in the two conditions contained a similar number of information units. In this case, the UMSR system achieved better understandability than the SR system.

Finally, Did We Achieve Higher User Satisfaction? For all conditions, users preferred the UMSR strategy over the SR strategy, and on no aspect was the SR system found to be significantly better than the UMSR system in our experiments.

We conclude that our hypotheses were largely confirmed, and that a challenge for future work lies in the development of methods that allow the system to predict and control for the cognitive load induced by system utterances. We believe that this is particularly important because dialog systems are frequently used in situations in which the user's hands and eyes are busy. For example, NASA is interested in spoken dialog technology for use in spacecraft (Rayner et al. 2003), and automobile manufacturers are integrating an ever-increasing number of voice services into their products.

In subsequent work, an end-to-end dialog system following the UMSR strategy was implemented in the context of the TownInfo system (Paksima, Georgila, and Moore 2009), which allows users to search for restaurants in the city of Edinburgh. The system was evaluated with participants interacting directly with the system (as opposed to a WOZ setting), and was compared to a system that implemented the typical sequential enumeration approach to information presentation. In the UMSR condition, both perceived task completion (90.7% vs. 85.2%) and actual task completion (74.1% vs. 62.9%) were higher than in the traditional sequential enumeration condition. Furthermore, dialogs with the UMSR system required significantly fewer turns to complete the task (9.24 for UMSR and 17.78 for the sequential system), and conciseness, accessibility, and efficiency were all rated more highly in the UMSR condition than the sequential enumeration condition. In addition, overall user satisfaction was significantly higher for the UMSR system. Finally, the UMSR system outperformed the sequential system particularly clearly in cases where a trade-off among options existed.

8. Related Work

Our experiments evaluate the UMSR model against the SR model described in Polifroni, Chung, and Seneff (2003). Recently, Polifroni and Walker (2008) developed several variants of the SR approach that differ in how they choose and structure attributes in the summaries presented to the user. That is, they differ in the way they rank attributes for presentation and the method used for clustering. There are two approaches to ranking, one based on the original refiner approach (which chooses to cluster on attributes for which a small number of clusters account for most of the options) and the other based on a user model (which chooses attributes for clustering based on the ranking in the user model). For clustering, they added an "associative mode" in which association rules among attributes are automatically determined and the summary statements produced describe the associations. Imagine that all options in a cluster with property X also

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

S: *I know of 35 restaurants in London serving Indian food. There are 3 medium-priced restaurants in Mayfair and 3 inexpensive ones in Soho. There are also 2 expensive ones in Chelsea.*

(a) SR with associated clustering.

S: *I know of 35 restaurants in London serving Indian food. There are 4 medium-priced restaurants with good food quality and 10 with medium food quality. There are also 4 that are inexpensive but have poor food quality.*

(b) SR with user-modeling and associated clustering.

Figure 16
Example summaries based on association rule clustering, without and with user-model based ranking, from Polifroni and Walker (2008).

happen to have property Y. Association then means that both properties X and Y are mentioned for the cluster (see Figure 16).

Polifroni and Walker (2008) evaluated the different versions of the refiner strategy approach in a reading experiment, similar to our first experiment. However, they evaluated only the first system turn in the dialog for the different conditions presented. The experimental conditions contrasted structuring with versus without a user model, and with versus without associative clustering. They found that users preferred the system utterances that were generated based on a user model when there were a large number of options available, and that the associative mode was beneficial when tailoring the summaries to the user. Their results hence support our approach: The version of the system that included user modeling and associative clustering led to dialogs that are the most similar to those of our UMSR approach.

There remain, however, a number of differences between their refined refiner with user modeling and associative clustering and our UMSR model. First, in UMSR the user model is not only used to structure, but also to select options (via pruning of the option tree). Second, Polifroni and Walker's (2008) approach contains two clustering steps: one for clustering attribute values, which is the same as in our UMSR approach, and a second one for clustering options following each query. This second clustering step corresponds to our tree structuring step. Whereas Polifroni and Walker re-cluster at every system turn, we only build up the option tree once (unless the user changes their query, e.g., by wanting to fly on another day, or asking to fly on another airline).

Finally, their notion of associative clustering (i.e., grouping together attributes whose values are connected, such as short flight duration and direct flights) is realized in our system as well, but in a less explicit way: In the tree structuring phase, attributes that have identical values for a set of options (this would for example apply if all flights with short travel time turned out to have only one leg) are moved up in the tree even if the attribute has a low rank (see Section 3.4). A difference is that our UMSR model treats the user's valuation as a more important feature for dialog structuring than associations.

Coping with Under- and Over-constrained User Queries. One frequent problem in spoken dialogs is caused by under- and over-constrained queries. In under-constrained queries, the database returns too many results to be presented at once, and the system must support users in further refining their query. In an over-constrained query, there are no

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

entries in the database that match the query exactly, and the system thus has to get the user to relax a constraint, or itself choose a constraint to relax.

Our UMSR system differentiates between cases that concern missing information about obligatory slots (e.g., `origin`, `destination`, and `date`) and underspecified preferences (concerning preferred airlines, etc.). To fill the obligatory slots, simple slot-filling questions such as *When do you want to leave?* are asked. For all other cases of over- or under-constrained queries, constraint relaxation and specification is inherent in the UMSR structuring approach. For underspecified preferences, the user model provides default specifications. In the case of over-constrained queries, we can think of constraint relaxation in terms of the tree structure: Constraints that are less important to the user (e.g., price for our example business user) are automatically relaxed simply because they are located further down in the option tree. Constraint values, namely, `good` arrival time being changed to `fair`, are automatically relaxed due to the ordering of the branches within the option tree.

Various strategies for coping with over- and under-constrained queries have been developed in the literature. Systems proactively suggest additional refinements for underconstrained queries, as in *You could try to look for cuisine type?* (Pon-Barry, Weng, and Varges 2006; Varges, Weng, and Pon-Barry 2006), or relaxations for over-constrained queries, as in *Would a slightly more expensive flight work for you?* (Qu and Beale 1999; Pon-Barry, Weng, and Varges 2006). Additionally, constraints can be relaxed automatically based an ontology (Varges, Weng, and Pon-Barry 2006) or by predefining relaxation sets (e.g., from a specific street for a restaurant to the broader neighborhood) or categorical ones (e.g., relaxing `Chinese` to `Asian`) (Chung 2004).

9. Conclusions and Future Work

The research presented in this article addresses the issue of how to scale information presentation strategies for spoken dialog systems to situations where many diverse options are available.

We developed an approach that combines content selection with content structuring, and showed how a user model can inform both of these steps to optimize efficiency and effectiveness of the information presentation phase in spoken dialog systems. The knowledge derived from the user model can be used to enable a dialog system to automatically present trade-offs between the relevant entities explicitly, thus improving the user's overview of the option space, and contrast options using linguistic and discourse cues. In our experiments, we found that the user model–directed approach led to shorter dialogs and that it improved task success: Users found the optimal option more often and were more satisfied with their experience. We tested this with three studies, a reading experiment, an overhearer experiment, and a Wizard-of-Oz experiment, thus providing what we believe is the first large-scale evaluation of alternative information presentation strategies that actually investigates how users interact with the dialog system to perform a task.

One significant contribution of the UMSR approach for intelligent information presentation is that it guides users through the dialog by presenting only options that are likely to be of high relevance to them, thus reducing the reliance on user speech. UMSR exploits information from the user model to provide users with the opportunity to select between the most suitable options available. It computes the trade-offs between options to most effectively narrow down the number of options. This way, considerably fewer turns are required per dialog in comparison with the SR approach, the average dialog duration is shorter, and the optimal option (i.e., flight) is booked more often.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

We conclude that combining a summarize and refine approach with user modeling is a very promising approach to improving the user experience in terms of achieving higher task success and increasing efficiency. However, there are also other parts of the presentation that could be tailored to the user (e.g., in adaptive option clustering). Although we used heuristics for how to exploit the information from the user model for content structuring, a very important next step would be to analyze the relationship between the user's valuations and how they explore the option space. It is conceivable that other ways of organizing the options could lead to even shorter refinement paths and quicker interactions. Concerning dialog turn length and complexity, we believe that better formal methods are needed in order to automatically estimate the cognitive burden placed on a user during the dialog interaction. We now briefly address these issues.

Tailored Clustering. The user model can also be used for further tailoring the system responses to the user by taking into account the user's valuations during the clustering step. Instead of using a fixed number of target clusters, it would be better to use a flexible number of natural clusters that the data fall into. This would make the labeling more difficult, but would make clusters (and therefore descriptions of option properties) more meaningful. Furthermore, the distance function used for clustering could be sensitive to the user model. For instance, the price axis used during clustering for a student who cares very much about price could be more fine-grained in the most relevant low price range.

An Adaptive User Model for Turn Length. We learned from our experiments and related work that turn length is important to control. Although we found that our system tended to produce turns that were often rather complex, in a study about choosing the optimal amount of information to convey to users, Whittaker, Walker, and Maloor (2003) found that subjects are biased towards more information. This means that providing an insufficient amount of information is perceived as worse than presenting too many details. Whittaker, Walker, and Maloor's results also indicated that conciseness should be tailored to users, because some users appreciate conciseness more than others. Our system does not currently tailor turn length to specific users, because our user model does not contain information about a user's conciseness preferences, or situational data, such as whether they are in a hurry, or can only devote a limited amount of attention on the task. Furthermore, Polifroni and Walker (2008) found in their user study that people who are not very familiar with the available options (e.g., tourists in a foreign city searching the restaurant domain) appreciate potentially longer, tailored summaries more than users who are very familiar with the data and know more precisely what they are looking for—such users preferred a simple refinement strategy or refinement with association over a user model–based recommendation.

Exploring the Relationship between the User Model and Optimal Dialog Structure (Minimize Exploration Need). We have shown that presenting users first with those options that are most relevant according to their user model improved efficiency over using an order which minimizes entropy but does not take into account the user model. However, we have not shown that this is necessarily the best possible order. Therefore, it would be interesting to run a more large-scale user experiment in which users use the system in an even more natural setting for booking flights that match their own interests, analyzing how often users explore more than one branch in the tree as a function of the tree structure. Having to investigate different branches of the tree is an indicator that the

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

tree structure is not optimal. This information could then be used to improve on the content structuring algorithm and learn more about the relationship between the user model and users' exploration strategies and interests when performing the task.

Measures of Cognitive Load. An interesting aspect in information presentation is also how much cognitive load the system-generated utterances place on the user. In this work, we try to manage cognitive load by controlling how many pieces of information are conveyed in one system turn. This is, however, a very coarse measure, as different linguistic factors have also been shown to affect cognitive load on users. For example, surprising or incoherent information structure has been observed to be more difficult to process (Vauras, Hyönä, and Niemi 1992; Van Gompel, Liversedge, and Pearson 2004), and complex syntactic structures (e.g., long-distance dependencies) have been shown to impose difficulty (Gibson 1998). Language processing difficulty is also affected by frequency: Less frequent structures and words are more difficult to process than more frequent ones (Ehrlich and Rayner 1981; Rayner 1998; Rayner et al. 1998). Furthermore, lexical access (Pollatsek and Rayner 1990) and word predictability (Rayner et al. 2001) have also been shown to play a role. Finally, deciding what is an information unit and estimating how difficult they are to remember (and recall) for a particular user is not a trivial task, and is highly dependent on individual differences in affect, aptitude, intelligence, episodic memory, and so on.

Therefore, a more fine-grained estimation of the cognitive load imposed on a user based on the number of information units to remember, the syntactic structure of the generated sentences, and their semantic coherence would allow the system to adapt to situational constraints (e.g., generate less complex but potentially less efficient interactions if the user is performing another task simultaneously), or to a specific user in order to account for age or disabilities.

In order to begin addressing some of these issues, and to gain further insight into what makes presentations based on the UMSR approach so successful, we have recently performed experiments investigating the role of linguistic devices to explicitly point out trade-offs among options (e.g., *but*, *also*, *just*, *only*), which are used in UMSR presentations to highlight relations among and trade-offs between options. We found that in both written and spoken presentations of options, those that include these linguistic devices facilitate recall for comparisons among options, when compared to a condition in which presentations do not include such linguistic cues (Winterboer 2009; Winterboer et al. 2011). In future work, we plan to consider how the syntactic complexity of the sentences in a presentation affects cognitive load, and how this interacts with both the number of options and attributes presented, and the use of linguistic devices to point out trade-offs.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. Vera Demberg was supported by Evangelisches Studienwerk e.V. Villigst. Andi Winterboer was funded by the Edinburgh-Stanford Link.

References

Bos, Johan, Ewan Klein, Oliver Lemon, and Tetsushi Oka. 2003. Dipper: Description and formalisation of an information-state update dialogue system architecture. In *4th SIGdial Workshop on Discourse and Dialogue*, pages 115–124, Sapporo.

Carenini, Giuseppe and Johanna D. Moore. 2001. An empirical study of the influence of user tailoring on evaluative argument effectiveness. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 1307–1314, Seattle, WA.

Carletta, Jean. 2007. Unleashing the killer corpus: Experiences in creating the multi-everything ami meeting corpus.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Language Resources and Evaluation, 41(2):181–190.

Chung, Grace. 2004. Developing a flexible spoken dialog system using simulation. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 63–70, Barcelona.

Dahlbäck, N., A. Jönsson, and L. Ahrenberg. 1993. Wizard of Oz studies—why and how. *Knowledge-Based Systems*, 6(4):258–266.

Demberg, V. and Johanna D. Moore. 2006. Information presentation in spoken dialogue systems. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 65–72, Trento.

Ehrlich, S. F. and K. Rayner. 1981. Contextual effects on word recognition and eye movements during reading. *Journal of Verbal Learning and Verbal Behaviour*, 20:641–655.

Georgila, Kallirroi, Oliver Lemon, James Henderson, and Johanna D. Moore. 2009. Automatic annotation of context and speech acts for dialogue corpora. *Natural Language Engineering*, 15(3):315–353.

Gibson, Edward. 1998. Linguistic complexity—locality of syntactic dependencies. *Cognition*, 68:1–76.

Hieronymus, James L. and John Dowding. 2007. Clarissa spoken dialogue system for procedure reading and navigation. Technical Report IAC-04-T.3.07, NASA Ames Research Center, Mountain View, CA.

Hu, J., A. Winterboer, C. I. Nass, J. D. Moore, and R. Illowsky. 2007. Context and usability testing: User-modeled information presentation in easy and difficult driving conditions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1343–1346, San Jose, CA.

Komatani, Kazunori, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi G. Okuno. 2003. Flexible guidance generation using user model in spoken dialogue systems. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, pages 256–263, Sapporo.

Linden, G., S. Hanks, and N. Lesh. 1997. Interactive assessment of user preference models: The automated travel assistant. *Courses and Lectures-International Centre for Mechanical Sciences*, pages 67–78.

Martin, David L., Adam J. Cheyer, and Douglas B. Moran. 1999. The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, 91–128.

Moore, Johanna D. 2006. Natural language generation for information presentation. Paper presented at the Spoken Language Technology Workshop SLT '06, December, Aruba.

Moore, Johanna D., Mary Ellen Foster, Oliver Lemon, and Michael White. 2004. Generating tailored, comparative descriptions in spoken dialogue. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Sociey Conference, AAAI Press*, pages 917–922, Miami Beach, FL.

Nass, C. and S. Brave. 2005. *Wired for Speech: How Voice Activates and Advances the Human–Computer Relationship*. The MIT Press, Cambridge, MA.

Paksima, T., K. Georgila, and J. D. Moore. 2009. Evaluating the effectiveness of information presentation in a full end-to-end dialogue system. In *Proceedings of the SIGDIAL 2009 Conference*, pages 1–10, London.

Polifroni, Joseph, Grace Chung, and Stephanie Seneff. 2003. Towards automatic generation of mixed-initiative dialogue systems from web content. In *Proceedings of Eurospeech '03*, pages 193–196, Geneva.

Polifroni, Joseph and Marilyn Walker. 2008. Intensional summaries as cooperative responses in dialogue: Automation and evaluation. In *Proceedings of ACL-08: HLT*, pages 479–487, Columbus, OH.

Pollatsek, A. and K. Rayner. 1990. Eye movements and lexical access in reading. In D. A. Balota, G. B. Flores d'Arcais, and K. Rayner, editors, *Comprehension Processes in Reading*. Erlbaum, Hillsdale, NJ, pages 143–163.

Pon-Barry, Heather, Fuliang Weng, and Sebastian Varges. 2006. Evaluation of content presentation strategies for an in-car spoken dialogue system. In *Interspeech 2006 – ICSLP*, pages 1930–1933, Pittsburgh, PA.

Qu, Y. and S. Beale. 1999. A constraint-based model for cooperative response generation in information dialogues. In *AAAI/IAAI 1999*, pages 148–155, Orlando, FL.

Rayner, K. 1998. Eye movements in reading and information processing. *Psychological Bulletin*, 124:327–422.

Rayner, K., K. S. Binder, J. Ashby, and A. Pollatsek. 2001. Eye movement control in reading: Word predictability has little influence on initial landing positions in words. *Vision Research*, 41:943–954.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Rayner, K., S. C. Sereno, R. K. Morris, A. R. Schmauder, and C. Clifton. 1998. Eye movements and on-line language comprehension processes. *Language and Comprehension Processes*, 4:21–49.

Rayner, Manny, Beth Ann Hockey, Jim Hieronymus, John Dowding, Greg Aist, and Susana Early. 2003. An intelligent procedure assistant built using regulus 2 and alterf. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 193–196, Sapporo.

Rich, Elaine. 1979. User modeling via stereotypes. *Cognitive Science*, 3(4):329–354.

Seneff, Stephanie. 2002. Response planning and generation in the mercury flight reservation system. *Computer Speech & Language*, 16(3–4):283–312.

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

Thompson, C., M. Goeker, and P. Langley. 2004. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research (JAIR)*, 21:393–428.

Van Gompel, R. P. G., S. P. Liversedge, and J. Pearson. 2004. Antecedent typicality effects in the processing of noun phrase anaphors. In M. Carreiras and C. E. Clifton, editors, *The On-line Study of Sentence Comprehension: Eyetracking, ERP and Beyond*. Psychology Press, New York, pages 119–137.

Varges, Sebastian, Fuliang Weng, and Heather Pon-Barry. 2006. Interactive question answering and constraint relaxation in spoken dialogue systems. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 28–35, Sydney.

Vauras, M., J. Hyönä, and P. Niemi. 1992. Comprehending coherent and incoherent texts: Evidence from eye movement patterns and recall performance. *Journal of Research in Reading*, 15(1):39–54.

Walker, M. A., S. J. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy. 2004. Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, 28(5):811–840.

Walker, Marilyn and Rebecca Passonneau. 2001. Date: A dialogue act tagging scheme for evaluation of spoken dialogue systems. In *HLT '01: Proceedings of the First International Conference on Human Language Technology Research*, pages 1–8, Morristown, NJ.

Walker, Marilyn A., Rebecca Passonneau, and Julie E. Boland. 2001. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 515–522, Toulouse.

White, Michael, Robert A. J. Clark, and Johanna D. Moore. 2010. Generating tailored, comparative descriptions with contextually appropriate intonation. *Computational Linguistics*, 36(2):159–201.

Whittaker, Stephen, Marilyn Walker, and Preetam Maloor. 2003. Should I tell all?: An experiment on conciseness in spoken dialogue. In *EUROSPEECH: European Conference on Speech Processing*, pages 1685–1689, Geneva.

Winterboer, A., J. Hu, Johanna D. Moore, and Clifford I. Nass. 2007. The influence of user tailoring and cognitive load on user performance in spoken dialogue systems. In *Proceedings of the Interspeech 2007 Conference*, pages 2717–2720, Antwerp.

Winterboer, Andi. 2009. *Evaluating Information Presentation Strategies for Spoken Dialogue Systems*. Ph.D. thesis, School of Informatics, University of Edinburgh.

Winterboer, Andi and Johanna D. Moore. 2007. Evaluating information presentation strategies for spoken recommendations. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, pages 157–160, Minneapolis, MN.

Winterboer, Andi, Martin I. Tietze, Maria Wolters, and Johanna D. Moore. 2011. The user model-based summarize and refine approach improves information presentation in spoken dialog systems. *Computer Speech and Language*, 25:175–191.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Appendix 1: Dialogue Scripts for Experiments 1 and 2

Training Scenario (Steve)

Steve's user profile:
You're going to read the transcripts of dialogues between a customer, Steve, and two different automated travel agents. When asked to fill out a form indicating his preferences for air travel, here's what Steve said:

"I'm a student, so the most important thing to me is to get a cheap flight. Otherwise I'm pretty flexible. I have no preferred airline and I don't care about particular layover airports."

A: Dialogue between Steve and UMSR system:

Steve: Hello. I need a cheap flight from Edinburgh to San Francisco on Thursday, 22nd September. I'd like to arrive by 6 pm.
System: I found three flights. They all have a short travel time but require a connection. The cheapest flight is on Air France. It arrives at 3:45 pm and costs just 304 pounds.
If you prefer to arrive closer to 6 pm, there are two British Airways Flights to choose from, but they both cost 586 pounds. All other flights are more expensive or arrive too late.
Steve: Tell me more about the cheap flight, please.
System: The Air France flight leaves Edinburgh at 9:05 am and requires a connection in Paris. The total travel-time is 14 hours and 40 minutes.

B: Dialogue between Steve and SR system:

Steve: Hello. I need a cheap flight from Edinburgh to San Francisco on Thursday, 22nd September. I'd like to arrive by 6 pm.
System: There are three cheap flights from Edinburgh to San Francisco that arrive between 3 and 6 pm. One of them is on Air France and costs 304 pounds. It arrives at 3:45 pm. The second one is a KLM flight that arrives at 4:20 and costs 349 pounds. The third flight is on Continental Airlines. It arrives at 4:35 pm and costs 389 pounds.
Steve: Can you give me more detailed information about the flight for 304 pounds, please?
System: It leaves Edinburgh at 9:05 am and requires a connection in Paris. The total travel time is 14 hours 40 minutes.

Scenario 1 (Carol)

Carol's user profile:
You're going to read the transcripts of dialogues between a customer, Carol, and two different automated travel agents. When asked to fill out a form indicating her preferences for air travel, here's what Carol said: "I'm a frequent flier and collect frequent flier points with KLM, so the most important thing when booking a flight is the airline. Second most important are the arrival time and the number of legs. Third most important is the travel time. I'm not too concerned about the price or the layover airport, although I prefer not to connect in London Heathrow."

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

A: Dialogue between Carol and UMSR system:

Carol: Hello. I would like to book a flight from Edinburgh to San Francisco for Thursday, 27th July, preferably leaving Edinburgh after 11 am. If at all possible, I'd like to fly on KLM.
System: Do you have any preferences for airports?
Carol: I'd rather not connect in London Heathrow.
System: I found two flights that both require a connection. There' s a KLM flight for only 348 pounds with a connection in Amsterdam, but it takes 18 hours and 20 minutes and leaves at 7:30 am.
To leave later and travel faster, you can take the Flybe flight that leaves only at 11:45 am and has a total travel time of only 15 hours 40 minutes. But you'd need to connect in London and the price is 580 pounds.

B: Dialogue between Carol and SR system:

Carol: Hello. I would like to book a flight from Edinburgh to San Francisco for Thursday, 27th July, preferably leaving Edinburgh after 11 am. If at all possible, I'd like to fly on KLM.
System: There are no flights that leave Edinburgh after 11 am and that are on KLM. However, there is a KLM flight leaving at 7:30 am.
Carol: Are there any flights on other airlines that leave Edinburgh after 11 am?
System: There are 7 flights that leave Edinburgh after 11 am. All of them have availability in business class.
Carol: Do you have information about the travel time of these flights?
System: Most of the flights take between 18 hours and 20 hours, one has a travel time of less than 16 hours.
Carol: Please tell me about that flight with under 16 hours travel time.
System: The Flybe flight leaves Edinburgh at 11:45 am and arrives San Francisco at 11 pm. It has a total of 15 hours 40 minutes and costs 580 pounds. You'd need to connect in London Heathrow.

Scenario 2 (Walter)

Walter's user profile:
You're going to read the transcripts of dialogues between a customer, Walter, and two different automated travel agents. When asked to fill out a form indicating his preferences for air travel, here's what Walter said: "I'm a business traveller, so the most important thing for me is to fly business class. I need to get to my meetings in time and don't want to spend too much time in airports, so second most important to me are arrival time, travel time, and number of legs. Third most important is to fly on KLM, because I collect bonus miles. I'm not concerned about travelling cheaply."

A: Dialogue between Walter and UMSR system:

Walter: Hello. I'd like to book a flight from Edinburgh to Brussels for tomorrow. I need to get there at around 3 pm and I'd prefer to fly business class.
System: Do you have any airline preferences?
Walter: Yes, KLM. And I'd rather not connect in London, if at all possible.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

System: 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 later than 4 pm.
Walter: Can you tell me more about the business class flights that arrive earlier?
System: 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 3:30 pm. There's a cheaper flight on Lufthansa. It arrives at 3:10 pm and costs just 270 pounds, but it takes 5 hours and 35 minutes. All other business class flights arrive much too early or very late.

B: Dialogue between Walter and SR System:

Walter: I'd like to book a flight from Edinburgh to Brussels for tomorrow.
System: There are 28 flights from Edinburgh to Brussels. Most of them require a connection and cost between 100 and 230 pounds.
Walter: Are there any direct flights with availability in business class that arrive around 3 pm?
System: There are no direct business class flights that arrive around 3 pm. However, there is one direct business class flight that arrives at 5:30. It costs 111 pounds.
Walter: No, that's too late . . . what about connecting business class flights that arrive around 3 pm?
System: There are three business class flights that require a connection and arrive around 3 pm. Two of them are on British Airways and arrive at 1:20 and 3:30. The third is on Lufthansa and arrives at 3:10.
Walter: How long do these flights take?
System: The British Airways flights take 3 hours and 20 minutes and the Lufthansa flight takes 4 hours and 35 minutes.
Walter: Are there any direct economy flights?
System: Yes, there's a direct flight on BMI. It arrives at 1:10 pm and has a travel time of 1 hour 35 minutes.

Scenario 3 (Carol)

Carol's user profile:

It's Carol again. This time she wants information about flights to Brussels. Remember what she said in her user profile: "I'm a frequent flier, so the first most important thing for me is the airline, because I collect frequent flier points with KLM. Second most important to me is arrival time and number of legs. Third most important is travel time. I'm not too concerned about the price or the layover airport, although I prefer not to connect in London Heathrow."

A: Dialogue between Carol and UMSR system:

Carol: Hello. I would like to book a flight from Edinburgh to Brussels for Thursday, 27th July, arriving at about noon. I prefer KLM flights.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

System: I found three flights with a good price. Two of them are on KLM. They both cost 111 pounds, but require a connection in Amsterdam. If you want to fly direct, there's a BMI flight arriving Brussels at 9:40 am that costs 107 pounds. All other flights arrive after noon.
Carol: Please tell me more about the KLM flights.
System: One of them arrives at 11:20 am but has a travel time of 4 hours 15 minutes. The other takes just 3 hours 10 minutes but arrives only at 4:40 pm.

B: Dialogue between Carol and SR system:

Carol: Hello. I would like to book a flight from Edinburgh to Brussels for Thursday, 27th July, arriving at about noon. I prefer KLM flights.
System: I found one KLM flight that arrives Brussels at 11:20 am. It takes 4 hours 15 minutes.
Carol: Please tell me more about that flight.
System: It leaves Edinburgh at 6:05 am and costs 111 pounds. You need to make a connection in Amsterdam.

Scenario 4 (Steve)

Steve's user profile:
It's Steve again. This time he wants to go to San Francisco. Remember Steve's user profile: "I'm a student, so the most important thing to me is to get a cheap flight. Otherwise I'm pretty flexible. I have no preferred airline and I don't care about particular layover airports."

A: Dialogue between Steve and UMSR system:

Steve: Hello. I would like to book a flight from Edinburgh to Brussels for Thursday, 27th July. Please find me a cheap flight.
System: What is your desired arrival time?
Steve: Ahm . . . in the early afternoon.
System: Ryan Air offers the cheapest flights to Brussels. They cost just 49 pounds, but you would have to connect in Dublin. There are two flights to choose from. There's an early flight, leaving Edinburgh at 8:05 am and arriving at 12:45 pm, with a travel time of only 3 hours 30 minutes. If you prefer to arrive at 5:30 pm, your total travel time would be 5 hours 30 minutes, leaving Edinburgh at 1 pm. If you want to fly direct, there's a flight on BMI that takes just 1 hour and 35 minutes. It arrives at 1:35 pm and costs 112 pounds. All other direct flights are more expensive.

B: Dialogue between Steve and SR System:

Steve: Hello. I would like to book a flight from Edinburgh to Brussels for Thursday, 27th July. Please find me a cheap flight.
System: There are three flights for only 49 pounds on Ryan Air. They all require a connection in Dublin.
Steve: Is there also a cheap *direct* flight?

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

System: The cheapest direct flight is on BMI and costs 112 pounds.
Steve: Okay… what are the departure and arrival times of the three RyanAir flights?
System: The first Ryan Air leaves Edinburgh at 8:05 am and arrives at 12:45 pm. The next one leaves at 1 pm and arrives at 5:30 pm. The last one leaves Edinburgh at 4:25 pm and arrives at 9:25 pm.

Scenario 5 (Walter)

Walter's User Profile
It's Walter again. This time he has another request. Remember his preferences: "I'm a business traveller, so the most important thing for me is to fly business class. I need to get to my meetings in time and don't want to spend too much time in airports, so second most important to me are arrival time, travel time, and number of legs. Third most important is to fly on KLM, because I collect bonus miles. I'm not concerned about travelling cheaply."

A: Dialogue between Walter and UMSR system:

Walter: Hello. I would like to book a flight from Edinburgh to San Francisco for tomorrow, arriving around 4 pm. It's important for me to fly on business class. A KLM flight would be nice.
System: There are two flights that arrive in San Francisco at 3:45 and 4:35 but neither of them is direct. They both have availability in business class and take under 15 hours. One of them is on Continental Airlines and the other is on Air France. There's also a KLM flight, but its travel time is 18 hours 20 minutes. All other flights arrive later than you requested.
Walter: Please tell me more about the Air France flight.
System: The Air France flight leaves Edinburgh at 9:05 am and arrives San Francisco at 3:45 pm. You need to connect in Paris. The flight costs 1,611 pounds on business class.

B: Dialogue between Walter and SR system:

Walter: I'd like to book a flight from Edinburgh to San Francisco for tomorrow, arriving around 4 pm. It's important for me to fly on business class. A KLM flight would be nice.
System: I found one KLM flight with availability in business class. It arrives at 3:45. You'd need to connect in Amsterdam.
Walter: Are there any direct flights that arrive around 4 pm?
System: There are no direct flights from Edinburgh to San Francisco. However, there are 7 flights that arrive San Francisco between 3 and 5 pm. They all have availability in business class.
Walter: How long does the KLM flight take?
System: The KLM flight has a travel time of 18 hours 20.
Walter: 18 h is very long. Are any of those other 7 flights faster?
System: There are three flights that take under 15 hours. The British Airways flight arrives at 4:10 pm, the Continental Airways flight arrives at 4:35 pm and the Air France flight arrives at 3:45 pm.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Walter: Please tell me more about the Air France Flight.
System: The Air France flight leaves Edinburgh at 9:05 am and arrives San Francisco at 3:45 pm. You need to connect in Paris. The flight costs 1,611 pounds on business class.

Appendix 2: Dialogue Samples for Experiment 3

Task descriptions for participants:

Task A: You need to book a flight from Edinburgh to San Francisco departing on May 12th. You'd like to fly business class and arrive in San Francisco around 4 pm, so that you have time to check into your hotel and shower before your 7 o'clock dinner meeting.
Task B: You need to book a flight from Edinburgh to Brussels departing on May 20th. You'd like to arrive by 6 pm so that you can make it to dinner with your possible new employer.
Task C: You're going from New York to Frankfurt departing on June 1st. You'd like to arrive in the late morning so that you can make it to a meeting that begins at 2 pm.
Task D: You need to book a flight from San Francisco to Prague departing on June 14th. You have to attend a meeting in the late afternoon, so you need to arrive before 3 pm.

Flight booking with SR system–Edinburgh to San Francisco

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: I'd like to book a flight from Edinburgh to San Francisco.
System: Please, tell me the date on which you want to travel.
User: January 3rd.
System: I found 20 flights from Edinburgh to San Francisco. All these flights require connection. There are flights available in economy, business, and first class. I also have information about arrival times.
User: I'd like to fly business class.
System: I found 5 business class flights from Edinburgh to San Francisco. All these flights require a connection. There are flights on American Airlines, Lufthansa, and United Airlines. I also have information about arrival times.
User: What is their arrival times?
System: Tell me your preferred arrival time?
User: 4:00 pm.
System: There are two business class flights from Edinburgh to San Francisco arriving between 2:00 and 4:30 pm. The first flight departs Edinburgh on Lufthansa at 5:40 am and arrives in San Francisco at 3:38 pm and requires a connection in Frankfurt. It costs 2,708 pounds. The second flight departs Edinburgh on United Airlines at 8:30 am and arrives in San Francisco at 3:18 pm and requires a connection at London Heathrow. It costs 4,452 pounds. Would you like to book one of these flights?
User: Yes.
System: Which one do you want me to book for you?
User: The first one.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No.
System: OK, goodbye.

Flight booking with SR system–Edinburgh to San Francisco

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: Yeah, I'd like to book another flight on January 3rd from Edinburgh to San Francisco.
System: I found 20 flights from Edinburgh to San Francisco. All these flights require a connection. There are flights available in economy, business, and first class. I also have information about arrival times.
User: I'd like to fly business class and, uh, I'd like to arrive around 4:00 pm.
System: There are two business class flights from Edinburgh to San Francisco arriving between 2:00 and 4:30 pm. The first flight departs Edinburgh on Lufthansa at 5:40 am and arrives in San Francisco at 3:38 pm and requires a connection at Frankfurt. It costs 2,708 pounds. The second flight departs Edinburgh on United at 8:30 am and arrives in San Francisco at 3:18 pm and requires a connection in London. It costs 4,452 pounds. Would you like to book one of these flights?
User: Do you have any flights on KLM that get in around 4:00 pm that are also business?
System: I found no business class flights from Edinburgh to San Francisco flying on KLM arriving between 2:00 and 4:30 pm.
User: Ok, can you repeat the options for business class flights that arrive around 4:00 pm?
System: There are 2 business class flights from Edinburgh to San Francisco arriving between 2:00 and 4:30 pm. The first flight departs Edinburgh on Lufthansa at 5:40 am and arrives in San Francisco at 3:38 pm and requires a connection in Frankfurt. It costs 2,708 pounds. The second flight departs Edinburgh on United at 8:30 am and arrives in San Francisco at 3:18 pm and requires a connection at London. It costs 4,452 pounds. Would you like to book one of these flights?
User: 4:00 pm.
System: There are two business class flights from Edinburgh to San Francisco arriving between 2:00 and 4:30 pm. The first flight departs Edinburgh on Lufthansa at 5:40 am and arrives in San Francisco at 3:38 pm and requires a connection in Frankfurt. It costs 2,708 pounds. The second flight departs Edinburgh on United Airlines at 8:30 am and arrives in San Francisco at 3:18 pm and requires a connection at London Heathrow. It costs 4,452 pounds. Would you like to book one of these flights?
User: Yeah, I would like to book the first one on Lufthansa.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No. That's it. Thank you.
System: OK, goodbye.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Flight booking with UMSR system–Edinburgh to San Francisco

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: Could I please go from Edinburgh to San Francisco on May 12th in business class?
System: I found three flights from Edinburgh to San Francisco with availability in business class. They are on Lufthansa and United.
User: Excellent, I would like to arrive in San Francisco at 3 or 4 pm, is there anything like that?
System: There are two flights with availability in business class. One is on United and arrives at 3:18 pm with a total travel time of only 14 hours and 48 minutes. The other flight is on Lufthansa and arrives at 3:45 pm, but it has a longer total travel time of 17 hours and 58 minutes. Would you like to book one of these flights?
User: Yes, I'd like the one with the shorter travel time, please.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No, thanks.
System: OK, goodbye.

Flight booking with UMSR system–Edinburgh to San Francisco

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: I would like to book a flight from Edinburgh to San Francisco.
System: Please, tell me the date on which you want to travel.
User: January 3rd.
System: There are no direct flights from Edinburgh to San Francisco but I found 3 flights with availability in business class. However, none of these 3 is on KLM. To travel on KLM you'll have to fly economy.
User: Um, ok, I'll fly business class with another airline.
System: Please wait while I look up the database .... I found 3 flights from Edinburgh to San Francisco with availability in business class. They are on Lufthansa and United. Do you have preferred departure or arrival time?
User: I'd like to arrive around 4:00 pm.
System: There are 2 flights with availability in business class. One is on United and arrives at 3:18 pm with a total travel time of only 14 hours and 48 minutes. It costs 4,452 pounds. The other flight is on Lufthansa and arrives at 3:38 pm but it has a longer total travel time of 17 hours and 58 minutes. It costs 2,708 pounds. Would you like to book one of these flights?
User: 4:00 pm.
System: There are two business class flights from Edinburgh to San Francisco arriving between 2:00 and 4:30 pm. The first flight departs Edinburgh on Lufthansa at 5:40 am and arrives in San Francisco at 3:38 pm and requires a connection in Frankfurt. It costs 2,708 pounds. The second flight departs Edinburgh on United Airlines at 8:30 am and arrives in San Francisco at 3:18 pm and requires a connection at London Heathrow. It costs 4,452 pounds. Would you like to book one of these flights?

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

User: I'll book the United flight. Get nice food.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No. Thanks.
System: OK, goodbye.

Flight booking with SR system–Edinburgh to Brussels

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: Yes, please.
System: What would you like to book?
User: I would like to book a flight from Edinburg to Brussels.
System: Please, tell me the date on which you want to travel.
User: On December 18th.
System: I found 21 flights from Edinburgh to Brussels. There are flights available in economy and business class. There are direct flights as well as flights that require a connection.
User: I'd like a direct flight in business class.
System: There are two direct business class flights from Edinburgh to Brussels. The first flight departs Edinburgh on BMI at 12 pm and arrives in Brussels at 2:40 pm. It costs 452 pounds. The second flight departs Edinburgh at 3:55 pm and arrives in Brussels at 6:15 pm. It costs 446 pounds. Would you like to book one of these flights?
User: Yes.
System: Which one do you want me to book for you?
User: The first one.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No.
System: OK, goodbye.

Flight booking with SR system–Edinburgh to Brussels

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: Yes, from Edinburgh to Brussels on December 18th.
System: I found 21 flights from Edinburgh to Brussels. There are flights available in economy and business class. There are direct flights as well as flights that require connection.
User: I would like to fly business class on KLM.
System: There is one business class flight from Edinburgh to Brussels flying on KLM. The flight departs Edinburgh on KLM at 12:40 pm and arrives in Brussels at 4:50 pm and requires a connection at Amsterdam. It costs 187 pounds. Would you like to book this flight?
User: Yes, please.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No.
System: OK, goodbye.

Flight booking with UMSR system–Edinburgh to Brussels

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: Yes.
System: Tell me what do you want to book?
User: I would like to go from Edinburgh to Brussels.
System: Please, tell me the date on which you want to travel?
User: December 18th.
System: There are two direct flights from Edinburgh to Brussels with availability in business class. All other direct flights only offer availability in economy class. If you are willing to make a connection there is a KLM flight with business class availability arriving at 4:50 pm.
User: Yes, I would like that KLM flight.
System: Wait . . . I found one flight from Edinburgh to Brussels with availability in business class on KLM and it requires a connection in Amsterdam. It arrives at 4:50 pm with a total travel time of 3 hours and 10 minutes and costs 187 pounds. Would you like to book the KLM flight?
User: Yes . . . yes.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No.
System: OK, goodbye.

Flight booking with UMSR system–Edinburgh to Brussels

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: Yes.
System: Tell me what do you want to book?
User: I would like to book a flight on KLM.
System: From where to where?
User: From Edinburgh to Brussels.
System: Please, tell me the date on which you want to travel.
User: December 18th.
System: There are two direct flights from Edinburgh to Brussels with availability in business class. All other direct flights only offer availability in economy class. If you are willing to make a connection, there is a KLM flight with business class availability arriving at 4:50 pm.
User: What time do the direct business flights arrive?

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

System: Please wait while I look up the database. I found two direct flights from Edinburgh to Brussels with availability in business class. Both flights are on BMI. The first flight arrives at 6:30 pm with a total travel time of 1 hour and 35 minutes. The second flight arrives at 2:40 pm with a total travel time of 1 hour and 40 minutes. Would you like to book one of the BMI flights?
User: No, neither of those flights will do, I need to fly on KLM. What flights do you have on KLM?
System: I found one flight from Edinburgh to Brussels with availability in business class on KLM and it requires a connection at Amsterdam. It arrives at 4:50 pm with a total travel time of 3 hours and 10 minutes. Would you like to book the KLM flight?
User: I'd like to book the flight.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No.
System: OK, goodbye.

Flight booking with SR system–San Francisco to Prague

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: Yes.
System: Tell me what do you want to book?
User: Uhh, I want a flight from San Francisco to Prague.
System: Please, tell me the date on which you want to travel.
User: On January 30th.
System: I found 21 flights from San Francisco to Prague. All these flights require a connection. There are flights available in economy, business, and first class. I also have information about price range.
User: I'd like business class.
System: I found 11 business class flights from San Francisco to Prague. All these flights require connection. There are flights in the expensive and moderate price range. I also have information about arrival times.
User: Arrival around 3:00 pm.
System: I found six business class flights from San Francisco to Prague, all arriving between 1:00 and 3:30 pm. All these flights require a connection. I have information about departure times. I also have information about price ranges.
User: A flight on KLM?
System: There is one business class flight from San Francisco to Prague flying on KLM, arriving between 1:00 and 2:30 pm. The flight departs San Francisco on KLM at 3:20 pm and arrives in Prague at 2:20 pm and requires a connection in Amsterdam. It costs 4,574 pounds. Would you like to book this flight?
User: Yes.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No.
System: OK, goodbye.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Flight booking with SR system–San Francisco to Prague

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?

User: Yes, from San Francisco to Prague on January 30th.

System: I found 21 flights from San Francisco to Prague. All these flights require connection. There are flights available in economy, business, and first class. I also have information about price range.

User: I would like to fly business class on KLM.

System: There are two business class flights from San Francisco to Prague flying on KLM. The first flight departs San Francisco on KLM at 3:20 pm and arrives in Prague at 2:20 pm and requires a connection at Amsterdam. It costs 4,574 pounds. The second flight departs San Francisco on KLM at 11:20 am and arrives in Prague at 2:45 pm and requires a connection at Amsterdam. It costs 4,574 pounds. Would you like to book one of these flights?

User: What time does the first flight arrive in Prague again?

System: The first flight departs San Francisco on KLM at 3:20 pm and arrives in Prague at 2:20 pm and requires a connection at Amsterdam.

User: I'd like to book that flight.

System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?

User: No.

System: OK, goodbye.

Flight booking with UMSR system–San Francisco to Prague

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?

User: I do. I need to go from San Francisco to Prague on June 14th.

System: There are no direct flights from Edinburgh to Prague, but I found 14 flights with availability in business class. Two of these are on KLM.

User: OK, I would like to go there on KLM and I have a meeting there at 4 pm. Do you have something coming in at around noon, or 1 pm?

System: I found only one KLM flight with availability in business class that will get you there on time. It arrives at 2:20 pm with a total travel time of just 14 hours but it requires a connection in Amsterdam and costs 4,574 pounds. All other flights take longer. Would you like to book the KLM flight?

User: Yes, please.

System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?

User: No. I don't.

System: OK, goodbye.

Flight booking with UMSR system–San Francisco to Prague

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?

User: Yes, I would.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

System: Tell me what do you want to book?
User: I'd like to book a flight from San Francisco to Prague.
System: Tell me the date on which you want to travel, please.
User: January 30th.
System: There are no direct flights from San Francisco to Prague but I found 11 flights with availability in business class. Two of these are on KLM.
User: I would like a business class flight on KLM, do you have anything arriving before 3:00 pm?
System: There are two business class flights from San Francisco to Prague on KLM that will get you there on time. The first flight arrives at 2:45 pm with a total travel time of 18 hours and 25 minutes. The second flight arrives at 2:20 pm with a total travel time of only 14 hours.
User: I would like the second flight please.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No.
System: OK, goodbye.

Flight booking with SR system–New York to Frankfurt

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: I'd like to depart New York on June 1st and go to Frankfurt, please.
System: I found 23 flights from New York to Frankfurt. There are direct flights as well as flights that require a connection. I also have information about fare classes.
User: I'd like a business class flight arriving before 2 pm, sorry, arriving before noon.
System: I found 13 business class flights from New York to Frankfurt arriving between 9 and 11:30 am. There are direct flights as well as flights that require a connection. I also have information about departure times.
User: I'd like a direct flight and if there is anything on KLM?
System: I found no direct business class flights from New York to Frankfurt flying on KLM arriving between 9:30 am and 12:30 pm.
User: I'll just take a direct flight then between 9:30 and 12:30 pm, please.
System: I found four direct business class flights from New York to Frankfurt. All these flights are in the moderate price range. All these flights depart between 6 and 9 pm. The flights are on Continental, Delta, and Lufthansa.
User: [Hesitation] Lufthansa, please.
System: There is one direct business class flight from New York to Frankfurt flying on Lufthansa arriving between 9:30 and 12:30 pm. The flight departs New York on Lufthansa at 7:05 pm and arrives in Frankfurt at 9:55 am. It costs 2,190 pounds. Would you like to book this flight?
User: Sure, please.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No.
System: OK, goodbye.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

Flight booking with UMSR system–New York to Frankfurt

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?

User: Yes.

System: Tell me what do you want to book?

User: I would like to book a business class flight from New York to Frankfurt on KLM.

System: Tell me the date on which you want to travel?

User: January 15th.

System: There are six direct flights from New York to Frankfurt with availability in business class. The only KLM flight with availability in business class arrives at 1:35 pm and requires a connection.

User: Tell me about the KLM flight.

System: I found one flight with availability on business class from New York to Frankfurt on KLM. It arrives at 1:35 pm with a total travel time of 12 hours and 35 minutes and costs 1,260 pounds. Would you like to book the KLM flight?

User: Yes, please.

System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?

User: No.

System: OK, goodbye.

Flight booking with SR system–New York to Frankfurt

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?

User: Yes, I would like to book a flight from New York to Frankfurt on January 15th.

System: I found 23 flights from New York to Frankfurt. There are direct flights as well as flights that require connection. I also have information about fare classes.

User: I need a business class flight.

System: I found 17 business class flights from New York to Frankfurt. I have information about arrival times. There are direct flights as well as flights that require connection.

User: It should arrive in the late morning, around 11:00.

System: I found 13 business class flights from New York to Frankfurt arriving between 9:00 and 11:30 am. There are direct flights as well as flights that require connection. I also have information about departure times.

User: How many of those flights are direct?

System: I found 4 direct business class flights from New York to Frankfurt arriving between 9:00 and 11:30 am. All these flights are in the moderate price range. All these flights depart between 6:00 and 9:00 pm. The flights are on United, Continental, and Delta.

User: Are any on KLM?

System: I found no direct business class flight from New York to Frankfurt flying on KLM arriving between 9:00 and 11:30 am.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022

User: Ok, I'll book one of the United flights.
System: There is one direct business class from New York to Frankfurt flying on United arriving between 9:00 and 11:30 am. The flight departs New York on United at 7:00 pm and arrives in Frankfurt at 9:10 am. It costs 3,990 pounds. Would you like to book this flight?
User: I'll book the flight.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No.
System: OK, goodbye.

Flight booking with UMSR system–New York to Frankfurt

System: This is the FLIGHTS information system. I am now connected to the network. Would you like to book a flight?
User: Yeah, I would like to book a flight.
System: Tell me what do you want to book?
User: Well, I want to book a flight on January 15th, uh, and that's going to be from New York to Frankfurt.
System: There are six direct flights from New York to Frankfurt with availability in business class. The only KLM flight with availability in business class arrives at 1:35 pm and requires a connection.
User: Uh, well, I don't think that's going to be possible. I do prefer to fly business class, though. What are the other business options that will get me there in the late morning?
System: I found three direct flights from New York to Frankfurt with availability in business class that will get you there on time. But none are on KLM. If you don't mind arriving earlier there is a shorter direct flight on United.
User: When is that going to be?
System: I found one direct United flight that departs at 7:00 pm and arrives at 9:10 am. It has the shortest total travel time of only 8 hours and 10 minutes and costs 3,990 pounds. Would you like to book the flight?
User: Yes, I would like to book that one, please.
System: Sounds good, let me book this flight for you. This will take a moment. Thank you for waiting; the electronic ticket and itinerary have been emailed to you. Would you like to book another flight?
User: No.
System: OK, goodbye.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00064 by guest on 20 August 2022