Information Presentation in Spoken Dialogue Systems

1. Purpose of Project

The goal of spoken dialogue systems is to offer an efficient and natural way of accessing information services of various types that range from product recommendation over access to emails to personal banking. The use of systems with speech interaction is particularly attractive for applications where the user communicates via telephone or can currently not use her hands and eyes to operate the system. A challenging issue in spoken dialogue systems, which this proposal addresses, is the restriction that (possibly complex) information has to be presented linearly.

In traditional systems, all different options and their respective attributes are simply enumerated. But if a lot of complex information is to be conveyed, this form of presentation makes it difficult for the user to remember the various aspects of multiple options and to compare them in memory, especially if not all of the user's concentration can be focused on the task (for example in dialogue systems that are used while driving a car).

So a great challenge in spoken dialogue system design is to present many options in an easily memorable and understandable way. This can be achieved by structuring the information in a way that takes into account the user's interest and valuation, leading to a structure that presents those criteria first that are most important for the user's decision process. What is important to a user is retrieved from a user model, which is acquired beforehand by either explicitly asking the user for the required information or by observing her choices.

Dialogue duration is another very important issue. As the recent evaluation of 9 dialogue systems shows, shorter task duration correlates with higher user satisfaction. Possible ways to shorten dialogue duration are to identify a small subset of relevant options, so that only these are presented in detail. This has been done in recent work by either asking questions to the user in order to narrow down the set of remaining options or by applying the knowledge provided in a user model to identify the options that are particularly promising.

A problem can arise when options are omitted and the information space is not fully accessible to the user. User confidence in the system might be decreased, because the user feels that the system is hiding information from her and she has no possibility to access it and override her user model.

Another crucial aspect in system development is its transferability to new domains. If evaluation criteria are hard coded, extending the database or applying the system to another domain might cause adaptation to be very laborious. Therefore, an algorithm that calculates these criteria dynamically from the database improves system transferability.

Improving system performance on these aspects will lead to greater user satisfaction and increase acceptance of spoken dialogue systems.

To address these issues, I will build on recent work in the field of spoken dialogue generation. The principal goal of my project is to develop domain independent techniques for presenting options that meet user constraints in spoken dialogue.

My approach to the problem will be implemented within FLIGHTS, an end-to-end spoken dialogue system for flight booking, recently built at Edinburgh.
2. Background

The FLIGHTS system [4] was an important contribution to the field of spoken dialogue generation because it showed how user preferences could be modelled and used to improve the generation of system responses. The user model affects many aspects of generation, such as content selection and discourse structuring, the choice of referring expressions and the control of prosody. The goal of the project was to generate spoken descriptions that are both easy to understand and memorable. Information from the user model was exploited to cut down dialogue duration and enhance understandability by choosing appropriate referring expressions and by incorporating into realization prosodic features that support the information structure and the rhetorical structure of the content that is to be conveyed to the user.

To shorten dialogue duration, the most relevant subset of options was determined and only the most relevant attributes for choosing among them were mentioned. Evaluation and ranking of an option were based on z-score, a commonly used measure. The relevance measures used in FLIGHTS were the 'compellingness' and 's-compellingness' measures introduced by [2]. 'Compellingness' measures the strength of supporting or opposing evidence for an argument (i.e. attributes having a value that is significantly better than the value of the top ranked attribute are compelling attributes). Options with compelling attributes are thus considered to be worth mentioning, because they offer trade-offs. The s-compellingness of an attribute represents the contribution of that attribute to the evaluation of an option (i.e. it mentions why an option is a possibly interesting trade-off).

To improve structure and understandability of a description, referring expressions that captured the main argument for an option were chosen. Discourse structure was determined by grouping and ordering attributes according to their valuation by the user and discourse connectives were then inserted to contrast positive and negative arguments. This informational structure (i.e. contrast between alternative attributes of an option and their relation to previously presented options) is supported by intonation. The use of intonation allows one to highlight important points and express contrast intelligibly. In FLIGHTS, theme (the topic of the interaction, on which the attention of the participants is presumably focused) and rheme (new information that is communicated) were assigned to each sentence to realize pitch accents correctly.

In my project, I will make use of the already implemented features of the FLIGHTS system and combine them with work by Polifroni, Chung and Seneff [5], who integrated into the restaurant recommendation system MATCH [7] a component that dynamically organizes numeric and symbolic data according to the current dialogue context. At each turn, their system calculates a set of options that meet user's constraints and differ along one dimension. Then a summary of the different attribute values of the current option set is generated (i.e. “I have found 19 seafood restaurants. They are predominantly in Back Bay, the North End, South Boston and the South End.”, where the user constraint is “seafood” and the dimension along which restaurants differ is their location). The presentation of this summary then allows the user to refine her constraints on that dimension and thereby to further narrow down the set of relevant options.

This approach is implemented using a clustering algorithm that allows one to dynamically adjust categories to the data by clustering the attribute values (i.e. calculating clusters of concrete prices which can then be described as “cheap”, “low”, “medium”, etc.). Summaries, that are generated on the basis of these clusters, make use of expressions like “all of them (are cheap)”, “none of them (serves seafood)”. To achieve maximally effective summarization, the attribute that splits the data into the smallest number of groups is chosen for clustering.

The effect of user-tailoring and varying verbosity on the quality of automatically generated descriptions was investigated in the multi-modal restaurant recommendation system MATCH by
M.A. Walker et al., see[6]. A paired t-test confirmed the hypothesis that people prefer responses generated with their own model over those generated from random models and gave significant evidence for the effectiveness of user tailoring. Evaluation also confirmed that varying conciseness has a strong impact on user satisfaction, indicating that the choice of an appropriate conciseness is crucial. Also, G. Carenini and J. Moore [2] found that the difference between non-tailored and tailored-concise arguments significantly indicates the higher effectiveness of tailored concise arguments. Finally, the evaluation result of the IDEA system by G. Carenini [1] found a significant impact of conciseness on how convincing an automatically generated evaluative argument is.

[3] also uses user tailoring and generates short description and recommendation texts that create discourse structures taking into account speaker's intentions and local coherence. Text structure is determined on the basis of the performance of different options with respect to the user model and by contrasting different options against one another. Also, the system of Chiarcos and Stede justifies every recommendation by relating each attribute of a decision to another option to which it is compared and to the user model (i.e. “as you requested”, “resembles the previous book in that”, “matches your request”, “which the previous book didn't”).

3. Methods

My master project will implement a synthesis of these different recently developed approaches, and will thus be a timely contribution to this research area of spoken dialogue system design. Clustering attribute values like in [5] will increase transferability and flexibility of the system with respect to the database. At the moment, valuations for attribute values are hard coded and thus inflexible. Building on the user model used in the current FLIGHTS system, I will cluster options along the attribute that is most important for the user. These clusters will then be summarized, creating an intuitive structure for the set of options and improving understandability. With this approach, contrarily to the current FLIGHTS system, all of the information will remain accessible to the user.

My hypothesis is that with my system, the user will be more confident and more certain to have chosen the best possible option, even if she does not actually explore more options in detail. She will hear a summarization of the irrelevant options and will therefore have the impression of a better overview over the option space than in systems that don't even mention the existence of some options or those that simply enumerate them. Also, the tree structure created by the successive clustering will provide a far more efficient and easy to memorize way of exploring different options than the original option enumerating version. I hypothesize that my version requires less concentration from the user than the original FLIGHTS system, because the amount of information per turn (before a decision is required) is smaller.

Interesting issues arising from my project are how this design affects user confidence, dialogue duration, user performance while only partly concentrating on the system, and overall user satisfaction.

I now want to describe in more detail the steps I will be taking during the 14 weeks of my project. The first milestone will be the implementation of a clustering algorithm. Concretely, for each level of the resulting option tree, this algorithm would cluster the different values for an attribute online. This will allow the mapping of concrete attribute values onto valuations, and finally to a natural language description for them. For example, depending on the prices available, a fare of 350€ will receive a low valuation and be described as “expensive” if the other flights for the same destination are much cheaper (i.e. for a flight from Edinburgh to London), whereas the same fare will be described as a “good price” in the case when all the other flights
available are more expensive (consider i.e. a flight from Edinburgh to Los Angeles). The basic motivation for this is that, at a summarization level, price difference matters not generally but only if it is significant. The output of this first step is a small number of clusters for each attribute. At each level, the choice of the attribute along which the options are clustered will depend on the relevance estimations for each attribute for the actual user. These are determined by the z-score measure (giving an overall value to each option) and the compellingness measure (whether an option is an interesting trade-off). Determining these values is part of the current FLIGHTS system, so I will tie up this information with my clustering algorithm. If the price of a flight is most important for a user, options would first be clustered into different price categories. Then a plan for a summary of each cluster will be generated, leading to subsets representing alternative trade-offs. This summary should mention the most important arguments for each cluster and make use of aggregation strategies as shown in [5] (i.e. summarizing with “all of them”, “predominantly”). For example, “There are several cheap flights but none of them is direct” or “There are two cheap flights and both of them arrive between 3 and 4 pm”. If for the current most important attribute there exist more than two categories, adjacent categories could be unified, if the rest of their attributes were homogeneous. For example, “There are also five flights in a moderate to expensive price range which are all direct.” However, the exact target output form will still have to be determined during the work on the project. Also, I will investigate which and how many attributes within a subset should optimally be mentioned. Ideally, the information will be arranged in a tree structure that minimizes paths length for the options that are most attractive to the user. A potential problem is that the clusters which result from clustering along the most important attribute may lead to very non-homogeneous clusters that are difficult to summarize.

To incorporate the new algorithm into the system, it will be necessary to adapt several of its components. This involves firstly extending the data base to have a wide variety of flights and attributes that makes clustering interesting. Secondly, I will write O-Plan operators in which the structure of the output will be determined - i.e. which attributes are to be mentioned. Thirdly, I need to specify the order in which the different clusters are to be presented and to contrast groupings of attributes. Adding new DIPPER (dialogue management) rules will involve creating a discourse structure that enables the user to give feedback after the initial information acquisition and question answering process. The turn will be given back to the user, so that she can decide about what cluster she wants to hear more. I do not plan to adapt the speech recognition component of the system. All of the input will be typed, and depending on how much work it is to integrate a parser and what kind of user interaction my system requires, I will either consider to integrate a parser, extend the currently used key word spotter by some new rules or provide input in a simple logical form such as a graphical user interface that proposes a choice of possible inputs. The choice made by the user will then be passed on to the O-Plan component that calculates the clustering for the remaining subset of options. Extending the sentence planner (Xalan XSLT) is necessary because at the moment the sentence planner is very limited and specialized to the actual output range of the current system. My version of it is very likely to have other structures and words and so they will have to be processable for the sentence planner. One challenge will be to generate an appropriate natural language representation of the summarizations of a cluster, where I will use generalized quantifier phrases like shown in [5]. The same applies to the CCG lexicon. Depending on how much time I have left, I will decide whether to include the theme/rheme information or not. Including it would be better for evaluation, because it is included in the current system, and I want to prevent evaluation results from being biased by that factor.
4. Evaluation

An evaluation with subjects using the original FLIGHTS system, my version and a conventional option enumerating system under real conditions will, even though highly desirable, not be possible. One reason for this is that I will focus on the design and implementation of the clustering algorithm and probably not have enough time to adapt all the components for an end-to-end system; further obstacles are the robustness and time efficiency problems of the FLIGHTS system.

The hypothesis that my version of the system is easier to understand under circumstances like using it in a car, won't be testable either, because an evaluation in a car simulator would be far too costly for this project.

An alternative form of evaluation is to record humans in the role of a user and the three different systems beforehand and play them to subjects, who would then be asked to fill in a questionnaire asking e.g. which system they would decide to buy, which system they feel most confident in and ask comprehension questions.

A possible evaluation of the clustering algorithm would be to compare how often subjects try to investigate options from different tree branches in a version where the clustering algorithm was based on their specific user model for clustering vs. in a version where it was based on a user model that is sufficiently different from their own one. Having to investigate different branches of the tree is an indicator that the tree structure was not optimal. In an optimal tree, the user should at every point be able to choose the optimal branch directly. To evaluate this, my version of the system would at least have to be able to process text input and print text output. Time efficiency problems of the system would probably not be as relevant in this form of evaluation as in the end-to-end system version, because time inefficiency seems to mainly be due to the speech generation component. Robustness would still be an issue, but subjects could be instructed to only use a fixed syntax that is not problematic for processing, or input would be based on templates, as already mentioned above.

The risk that this manipulation would have a biasing effect on the result of this particular evaluation seems minor to me, because the only parameter that changes in this evaluation is the user model that underlies the tree structure. The input form should not have any impact on that.

However, user's behaviour in navigating through the option tree is probably not only dependent on the clustering algorithm. I hypothesize that it is also dependent on the conciseness of the cluster summary, which will be controlled by a parameter. The influence of conciseness on the performance of the system can be estimated by measuring user satisfaction while the only aspect changed in the system is the conciseness parameter. This kind of evaluation was successfully done in several recent systems [2],[7]. One important aspect is the linear presentation of the information. Printing a text as proposed in the previous evaluation method would bias the result because information would not be presented linearly to the user any more: the user could go back in the text and compare different options not only in memory, so a text output based evaluation of this aspect would not give relevant results. This could be solved by either integrating the text-to-speech component of the system or scrolling the text automatically so that the subjects cannot go back.

How many of these evaluation proposals I will be able to realize mainly depends on the number of components of the system that I run robustly and on the how many subjects I can get for evaluation.

5. Risks and Contingency Plan

A risk is that I might need more time for integrating my algorithm and adapting the different components of the system than I expected. The actual FLIGHTS system components are written in various programming languages and I have no experience in working with such a big system. If that turns out to be a serious problem, I will have to concentrate on the core components, which are the clustering algorithm and the content planning component.
My contingency plans for this case are as follows: If integrating the parsing component takes too much time I won't do any automated speech recognition, but will either fall back to keyword spotting or presenting menus. I also might not have time to adapt rules for theme/rheme annotation of the output sentences. In that case I would use a TTS system without prosody control or do text generation only.

6. Deliverables

The deliverables of the different stages of my project are:
1) an information presentation strategy, based on the summarization of clusters
2) several pieces of code:
   an implementation of the information presentation strategy
   a clustering algorithm for structuring options according to their compellingness
   integration into the different components of the FLIGHTS system (as far as time allows)
3) for the evaluation:
   recordings of the system answers and of simulated user input
   a questionnaire designed to evaluate the system
   [component evaluations of the summarization and the clustering algorithm using observations about subjects' navigation through the option tree]
4) a thesis presenting the work

7. Work Plan

* The feasibility estimation for adaptation of the different system components will be done so early on in the project, because Mike White, who played a decisive role in the creation of the CCG component, will leave the institute in June.

---

1 optional, depending on the status of the system and the availability of evaluation subjects
References


