Evaluation of Content Presentation Strategies for an In-car Spoken Dialogue System

Heather Pon-Barry[1], *Fuliang Weng*[1] *and Sebastian Varges*[2]

[1]Research and Technology Center, Robert Bosch Corporation
[2]Center for the Study of Language and Information, Stanford University
heather.pb@stanfordalumni.org, fuliang.weng@rtc.bosch.com, varges@stanford.edu

10.21437/Interspeech.2006-530

ABSTRACT

In this paper we present a framework for managing information presentation in spoken dialogue systems. We describe a *content optimization* module that makes use of ontological relationships in information-seeking dialogues in order to organize knowledge base items and perform adjustments such as relaxing or tightening user constraints. We present the results of an experimental evaluation comparing two response strategies: (a) one that uses the content optimization module to offer suggestions and (b) one that gives no suggestions. The results indicate that giving such suggestions is preferred when a user query matches either no items or many items in the knowledge base, and may also lead to more efficient dialogues.

Index Terms: spoken dialogue systems, content management

1. INTRODUCTION

In recent years, conversational dialogue systems have become more sophisticated and more prevalent in our everyday lives. We see them being used for a variety of over-the-phone tasks (e.g., booking flights, finding hotels, ordering pizza, etc), and recently, being deployed in state-of-the-art luxury vehicles [1].

One current limitation in many of these systems is that users are required to recite specific phrasings or listen to lengthy prompts. For cognitively demanding situations such as driving in heavy traffic, it is of crucial importance to allow users to speak naturally and to allow flexibility in dialogue flow. Because drivers are focusing on the road and not on a graphical display, it is essential for the system content to be delivered in a way that does not overwhelm them or distract them from the task of driving.

At the same time, dialogue systems that help users access large databases need to be sufficiently informative. When speech is the main mode of communication, it is easy to overload or confuse users by giving too much or too little information. The CHAT dialogue system (Conversational Helper for Automotive Tasks) currently supports restaurant selection and mp3 player applications—both of which involve helping users access information from large databases. In cases where user queries return no matches or many matches, CHAT makes use of ontological relationships in order to provide users with suggestions about how to proceed.

This paper describes a content optimization and organization module that we have developed in an effort to address these problematic issues. We present the results of an experimental evaluation comparing two response strategies—one that gives suggestions based on ontological relationships and one that gives no suggestions. Our results indicate that whether or not suggestions are preferred depends on the number of items in the database matching the user's query. In addition, the results indicate that giving suggestions leads to fewer user turns per dialogue.

2. RELATED WORK

The issue of how to best present information in spoken dialogue systems has been the focus of much previous research. The authors in [2] present a decision-theoretic framework for generating comparisons and recommendations tailored to individual user preferences in a restaurant selection domain. They also showed that users rated the tailored responses more highly than the non-tailored responses [3]. In [4], the author presents a restaurant information system that calculates frequency statistics for the items in the result set in order to better summarize results. The framework in [4] also provides functionality for relaxing over-constrained queries.

Related empirical work has shown that 'literal' and 'cooperative' response strategies in a dialogue system accessing train schedules had complementary strengths and weaknesses depending on both the contents of the result set and the difficulty of the task [5].

Our work focuses specifically on information presentation in cognitively demanding situations such as driving a car. We aim to understand how to balance the conflicting goals of being as informative as possible and minimizing cognitive load. The system described in this paper builds on previous work described in [6].

3. THE CHAT DIALOGUE SYSTEM

This section gives an overview of the primary components and functionality of the CHAT dialogue system. CHAT provides end-to-end spoken language processing for interaction with multiple devices, using a combination of off-the-shelf components, components used in previous language applications, and components specifically developed as part of this project. Unlike the hub architecture employed by many dialogue systems (e.g., [7], [8]), we use an event-based, message-oriented middleware. Event-based architectures are the current paradigm for distributed systems, especially those allowing dynamic registration of new components.

The core components of the system are the *Natural Language Understanding* module, the *Dialogue Manager*, and the *Knowledge Manager*, each of which will be described in the upcoming paragraphs. In addition, we use Nuance (http://www.nuance.com), with dynamic grammars and class-based n-grams, for speech recognition, and Nuance Vocalizer for text-to-speech synthesis. Figure 1 below depicts the dialogue system architecture. The *Content Optimization* module, which sits between the Knowledge Manager and the Dialogue Manager, is depicted with a dashed border.

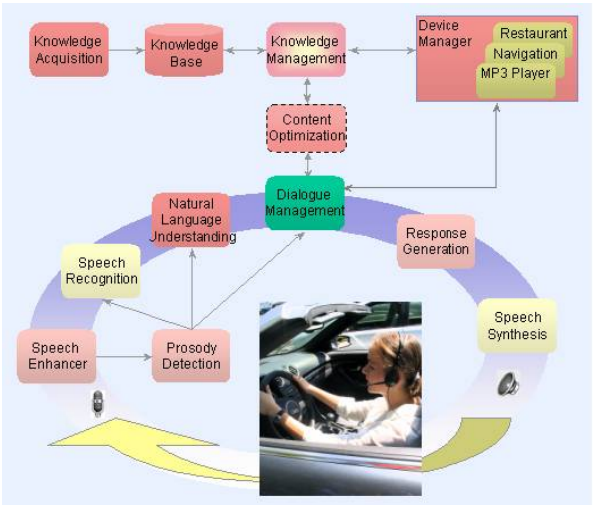

Fig 1. Dialogue System Architecture

The **Natural Language Understanding** (NLU) module has been designed to provide maximum flexibility of spoken language while still achieving robustness. The Bosch NLU module consists of two separated paths: a deep structural analyzer, including a HMM-based tagger [9], a statistical dependency parser, and a semantic mapper [6]; and a shallow semantic parser. The latter, similar to [10], produces dialog acts, predicate, and attribute value pairs in a flat representation. The output from both paths and their confidence scores are sent to the Dialogue Manager for further processing [6].

The **Dialogue Manager** (DM). Interaction is mediated and managed by the CSLI Dialogue Manager. The DM uses the information-state-update approach to maintain dialogue context, which is then used to interpret incoming utterances (including fragments and revisions), resolve noun phrases, construct salient responses, track issues, etc. The DM takes the input from the SR and NLU modules as well as the current context, and uses a global optimization algorithm to select the best candidate interpretation. Detailed descriptions of the Dialogue Manager can be found in [11] and [12].

The **Knowledge Manager** (KM) controls access to knowledge base sources (such as domain knowledge and device information) and their updates. Domain knowledge is structured according to domain-dependent ontologies. The current KM makes use of OWL, a W3C standard, to represent the ontological relationships between domain entities. Protégé (http://protege.stanford.edu), a domain-independent ontology tool, is used to maintain the ontologies offline. In a typical interaction, the Dialogue Manager converts a user's query into a semantic frame (i.e. a set of semantic constraints) and sends this to the KM via the content optimizer (to be detailed later).

The **Natural Language Generator** provides flexible feedback about what the system understood and what actions it performed. It goes beyond standard template generation and uses a set of recursive rules encoded as productions written in an expert system shell (see [13] for more details). These rules produce candidate system responses that are ranked based on alignment and variation scores, combined with ngram-based filters. There is evidence that the use of overgeneration and ranking is well-suited to model alignment in dialogue.

Applications currently supported are a Restaurant Selection System and a MP3 Music Player. The experiment described in this paper was conducted using the Restaurant Selection System. An example restaurant selection dialogue is shown below.

USR: i want to find a German restaurant on main street
SYS: There are no German restaurants on Main Street. However, I found 32 restaurants on Main Street that serve European food. Can you try to refine your search?
USR: a formal one
SYS: I found 9 European restaurants on Main Street that have a formal dress code. I am going to list the first few:
SYS: PASTA PRONTO
SYS: VALENTINO'S
SYS: DI MILLE'S ITALIAN RESTAURANT
SYS: Should I continue?
USR: tell me about pasta pronto
SYS: PASTA PRONTO is a moderate Italian restaurant. Dress code is formal, service is excellent and rating is four star. It's on MAIN STREET, in GRANT, 3.65 miles away.

Figure 2. Example dialogue with restaurant system

4. CONTENT MANAGEMENT

The **Content Optimization** (CO) module acts as an intermediary between the DM and the KM to regulate the amount of information to be presented to the user.

When the Content Optimizer obtains a semantic frame from the Dialog Manager, it performs the following operations:

1. Resolve ambiguous property names
2. Merge with previous frame (if this is a query revision rather than a new query)
3. Send query to KM
4. Select top-level strategy (e.g., processEmpty, processMedium, processLarge, none) based on number of items returned
5. If top-level strategy=none, return query result.
6. Else: select constraint to modify, select specific modification strategy, perform modification, and return to step (3).

The Content Optimizer contains generic strategies for performing modifications such as relaxing, tightening, adding, and removing constraints. The selection of which constraint to modify and which modification to perform (if any) is determined by a combination of factors including the number of items in the result set, the system ontology, prosodic information, information from the user model, and current road conditions.

The manner in which a constraint is relaxed depends on what kind of values it takes. For example, CUISINE values are related hierarchically (e.g., Chinese, Vietnamese, and Japanese are all subtypes of Asian), whereas PRICE values are linear (e.g., cheap, moderate, expensive), and CREDITCARD values are binary (e.g., accepted or not accepted). The configuration files associate each domain-specific constraint with a generic strategy in the content optimizer core.

Based on ontological structures, the Content Optimizer calculates descriptive statistics for every set of items returned by the knowledge manager in response to a user's query. When there are too many items in a result set, these figures can be used by the Dialogue Manager to give suggestions (e.g., "There are 85 songs. Do you want to list them by a genre such as Rock, Pop, Soul, or Jazz?").

The Content Optimizer was designed with emphases on both portability and easy integration with user models. The module contains a library of domain-independent strategies and makes use of external configuration files to specify under which conditions a strategy ought to be used. In this way, the work required to add a new domain is minimal—requiring only a new configuration file and a domain ontology.

5. EXPERIMENT

We ran an experiment comparing two response strategies that varied in the suggestions given to the user. One strategy made use of the content optimization module and gave suggestions about how to proceed when a user query returned no results or many results (for this experiment 'many' means 'greater than 30'). The second strategy gave responses without such suggestions. Example responses are shown below. Aside from the times a user query returned many results or no results, the two response strategies were identical.

Suggestions strategy (S)

1.a Many matches: *I found 656 cheap restaurants. You can refine your query by adding criteria such as cuisine type.*

1.b No matches: *I found no restaurants in Jackson that serve Indonesian food. However, there are 25 restaurants in Jackson that serve Southeast Asian food.*

No Suggestions strategy (NS)

2.a Many matches: *I found 656 cheap restaurants.*

2.b No matches: *I found no restaurants in Jackson that serve Indonesian food.*

5.1 Experimental Design

Participants were 16 native English speakers, ranging between 19-65 years of age. The experimental procedure included three warm-up tasks followed by six evaluation tasks. For each task, participants read a short scenario description containing three criteria, and were instructed to use their own language to talk to the system and find a restaurant. The six evaluation tasks were designed so that a query containing the three criteria would return (a) few matching restaurants, (b) no matching restaurants, or (c) many matching restaurants.

Each participant was randomly assigned to Group A or Group B. Group A received suggestions (S) for tasks 1-3 and no suggestions (NS) for tasks 4-6. To counterbalance, Group B received suggestions (S) for tasks 4-6 and no suggestions (NS) for tasks 1-3. The experiment design is summarized below in Table 1.

We predicted that users would prefer strategy S over NS in the tasks with no matches and many matching restaurants, and that there would be no difference in preferences for the tasks with few matching restaurants.

To ensure that task pairs ({1, 4}, {2, 5}, {3, 6}) would be comparable, each pair contained the same three criteria (e.g., cuisine type, location, and rating) but differed in the values.

Task	Num Matches	Response Strategy Group A	Response Strategy Group B	Prediction
1	Few (5)	S	NS	same
2	None (0)	S	NS	S > NS
3	Many (487)	S	NS	S > NS
4	Few (2)	NS	S	same
5	None (0)	NS	S	S > NS
6	Many (429)	NS	S	S > NS

Table 1. Experiment design

5.2 Results

Data was collected for a total of 96 dialogues. User satisfaction was gauged by asking users to indicate on a 5-point Likert scale their agreement or disagreement with the following statement, "The system's utterances in this task were easy to understand and provided exactly the information I was interested in when choosing a restaurant" (in line with [3]), where 1 = strongly disagree and 5 = strongly agree.

The user satisfaction results indicate that responses with suggestions are preferred over responses with no suggestions for queries with no matches and many matches, in line with our predictions. Mean user ratings are summarized below in Table 2. Paired samples T-tests were performed for each task category. The difference in means was not significant for the few matches and no match categories, but differences were significant (indicated with an '*' below) for the many matches category ($p < 0.1$). Contrary to our prediction that S and NS would be equally preferred for the few matches category, NS responses were rated slightly higher.

	No matches (N=32)	Few matches (N=32)	Many matches (N=32)
S	3.68	4.50	4.25*
NS	3.31	4.75	3.63*

Table 2. Mean user satisfaction ratings

We ran an ANOVA to determine whether any of the user satisfaction ratings differed as a function of subject group, and found that this was not the case (i.e., the order in which users saw the two response strategies did not affect their ratings).

Within the 'many matches' category, we looked at means for individual tasks. The mean user ratings are shown below in Table 3. Interestingly, S was rated significantly higher than NS for Task 3 ($p < 0.1$), but for Task 6, there was no difference.

We also measured whether response strategies had any effect on dialogue length (number of turns). Across all 96 dialogues, the mean number of turns per dialogue was 14.37 for S and 16.69 for NS. For S dialogues, mean user turns was 5.29 and mean system turns was 9.08; for NS dialogues, mean user turns was 6.48 and mean system turns was 10.21. These results are shown below in Table 4. A paired samples T-test was run on each of the S-NS pairs. The difference in means for user turns per dialogue was significant ($p < .01$) (marked below with an '*') while the differences among the system turns and the total turns were not significant.

	Task 3 - Many matches (N=16)	**Task 6 - Many matches (N=16)**
S	4.38*	4.00
NS	3.25*	4.00

Table 3. Mean ratings for 'many matches' category

	Total Turns / Task (N=96)	**User Turns / Task (N=96)**	**System Turns / Task (N=96)**
S	14.37	5.29*	9.08
NS	16.69	6.48*	10.21

Table 4. Mean number of turns per task

5.3 Discussion

The results of the experiment indicate that users prefer responses with suggestions for the cases when they encounter empty result sets or very large result sets, and that they prefer responses with no suggestions for queries with few results. This preference is most salient for situations where the query result contains many matching items.

The fact that the S-NS difference in means for Task 3 was so pronounced and the means for Task 6 were identical indicates that responses with suggestions may only be desirable for new users. One possible explanation is that participants in Group A, who received NS responses in Task 6, were sufficiently familiar with the space of possible constraints such that the lack of suggestions made no difference.

The lower number of mean dialogue turns for the S strategy, as compared to the NS strategy, indicates that giving responses with suggestions may lead to more efficient dialogues. While it is promising that the 'suggestions' strategy led to fewer user turns, further analysis is needed to determine whether these dialogues are less cognitively demanding.

6. CONCLUSION

This paper describes a framework for managing information presentation in spoken dialogue systems by utilizing ontological relationships to make suggestions. The experimental results support our hypothesis that giving users suggestions about how to proceed, either by adding new constraints or relaxing existing constraints, is preferred in situations where the user's original query yields no matching items or many matching items. Plans for future work include further testing of response strategies and integration with a dialogue system for route navigation.

7. ACKNOWLEDGEMENTS

The work described in this paper was supported by the NIST Advanced Technologies Program and by Robert Bosch Corporation. This research was conducted when Heather Pon-Barry was at Bosch RTC. Her new affiliation is the Division of Engineering & Applied Sciences, Harvard University.

8. REFERENCES

[1] P. Heisterkamp, "Linguatronic: Product-level speech system for Mercedes-Benz cars," in *Proc. of Human Language Technology Conference (HLT)*, San Diego, California, March 2001.

[2] M. Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G Vasireddy, "Speech-plans: Generating evaluative responses in spoken dialogue," in *Proc. of INLG*, 2002.

[3] M. Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G Vasireddy, "Generation and evaluation of user tailored responses in multimodal dialogue," *Cognitive Science 28*, pp. 811-840, 2004.

[4] G. Chung, "Developing a flexible spoken dialog system using simulation," in *Proc. of ACL 2004*, pp. 63-70.

[5] D. Litman, S. Pan, and M. Walker, "Evaluating response strategies in a web-based spoken dialogue agent," In *Proc. of COLING-ACL'98*, Montreal, Canada, pp. 780-786, August, 1998.

[6] F. Weng, S. Varges, B. Raghunathan, F. Ratiu, H. Pon-Barry, B. Lathrop, et al., "CHAT: A Conversational Helper for Automotive Tasks," in *Proc. of ICSLP 2006*, Pittsburgh, PA, September 2006.

[7] B. Pellom, W. Ward, and S. Pradhan, "The CU Communicator: An architecture for dialogue systems," in *Proc. of ICSLP 2000*, Beijing, China, November 2000.

[8] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue, "GALAXY-II: A reference architecture for conversational system development," in *Proc. of ICSLP 1998*, Sydney, Australia, December 1998.

[9] E. Dermatas and G. Kokkinakis, "Automatic stochastic tagging of natural language texts," *Computational Linguistics 21*(2), pp. 137-163, 1995.

[10] Y. He and S. Young "A Data-Driven Spoken Language Understanding System," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, US Virgin Islands, 2003.

[11] O. Lemon, A. Gruenstein, and S. Peters, "Collaborative activities and multi-tasking in dialogue systems," *Traitement Automatique des Langues (TAL)*, 43(2), 2002.

[12] D. Mirkovic, and L. Cavedon, "Practical Plug-and-Play Dialogue Management," in *Proc. of PACLING-2005*, Tokyo, Japan, August 2005.

[13] S. Varges, "Chart generation using production systems," in *Proc. of 10th European Workshop on Natural Language Generation*, Aberdeen, Scotland, August 2005.