INTERFACES FOR SPEECH RECOGNITION SYSTEMS: THE IMPACT OF VOCABULARY CONSTRAINTS AND SYNTAX ON PERFORMANCE

Kate S. Hone and David Golightly

ICL Institute of Information Technology,
School of Computer Science and Information Technology,
University of Nottingham, University Park, Nottingham, NG7 2RD, UK.

ABSTRACT

An experiment was conducted to investigate the effects of vocabulary constraints and syntax on human interactions with a speech interactive system. Three dialogue styles for a telephone banking application, all using constrained vocabularies, were compared: yes/no, menu and query prompts. These styles differ both in the degree of vocabulary constraint, and in how that constraint is communicated to the user. It was found that although it involved more dialogue steps the yes/no interaction style was the most effective in terms of both task completion rates and performance time. The query strategy was least preferred by users.

1. INTRODUCTION

Speech recognition technology has the potential to greatly increase the range and accessibility of services which companies can offer their clients via the telephone. Speech recognition is a cheaper solution than using human operators. It is also more 'user friendly' than the alternative of using touch-tone input from the telephone keypad, and it is not dependent on access to touch-tone technology. Already there are a range of services available over the telephone which rely on speech recognition. For example in the UK British Telecom provides a speech interactive, network based answering machine: CallMinder™. Other application areas include telephone banking and travel information. However, while there is enormous potential for such speech driven systems, their success depends on finding implementations which 'exploit the strengths of the technology and hide its weaknesses' [1]. Good interface design is key to achieving this objective.

Much contemporary research is aimed at producing truly conversational applications, which use speech recognition results, combined with grammatical rules and information about the application domain, to interpret users' requests to the system. However, such systems have considerable potential for recognition errors, especially if used for large, heterogeneous user populations, and with relatively poor quality microphones. These characteristics are typical of current telephone applications, meaning that until technology improves developers must take a different approach to interface design. Typically this means using constrained vocabulary sets, with further constraints employed a specific points in an interaction (depending on application syntax). This approach considerably reduces the likelihood of recognition errors and allows fairly robust telephone applications to be developed.

There are various ways to implement small vocabulary interactive voice response systems. The key choices center on the extent to which vocabulary size is reduced at each point in an interaction, and how the user is informed of the vocabulary restrictions. Waterworth [2] presents a hierarchy of voice entry modes which is useful as it addresses the relationship between these two variables. Four of the levels identified are shown in table 1.

Mode	Example Prompt
Query	Which service do you require?
Menu	Which service; balance, cash transfer or other?
Yes/No	Would you like to hear your balance?
Grunt	Make a sound now to hear your balance.

Table 1: Voice data entry modes (adapted from [2]).

Waterworth argues that these modes are ordered along a dimension ranging from most sophisticated (in terms of recognition technology needed) and least explicit (in terms of how the user is informed of the system's requirements) at the top, to least sophisticated and most explicit at the bottom. He also argues that there is a parallel hierarchy of size of acceptable user response class (with smallest at the bottom, largest at the top). The importance of these parallel hierarchies becomes clear when one examines the performance implications. In general recognition performance will improve as you move from the top to the bottom of the hierarchy (as the vocabulary size falls). This should act to increase dialogue efficiency due to the reduction in time consuming error correction moves. On the other hand the restrictions on what can be said tend to introduce extra steps into the dialogue. This is time consuming in itself and also brings more opportunities for recognition errors. Thus there are clearly trade-offs inherent in choosing a dialogue style for small vocabulary systems. The current paper considers the issue of how to make this choice.

The advice currently available to designers on this issue is limited. Some authors have attempted to provide relevant guidelines. For instance [3] suggests that explicit prompts which constrain user utterances will be helpful for systems which have to work well for first time callers. However, while such guidelines are useful, they do not allow designers to quantify the trade-offs involved in their design decisions. For instance when do the advantages of a highly constraining prompt, like a spoken menu, outweigh the disadvantages? Published empirical research on this issue is also limited. One study, however, does address the relative merits of query, menu and yes/no styles of dialogue [4]. This research compared various versions of these prompt styles in a automated telephone operator application. It was found that a question-only style, such as "what type of call would you like to make", was least effective in constraining users to a valid keyword with minimal extraneous speech (no more than two additional words). In this condition only 61% of user responses met the criterion of keyword plus two extra words, compared to 99% when menu items were listed in the prompt. On the other hand the question-only strategy led to significantly faster transaction times than the menu strategies. Unfortunately, however, it is not clear whether this timing data would be predictive of real system performance as it was obtained with a simulated system, using a "Wizard of Oz" who accepted all valid keywords so long as they were embedded in no more than one sentence of speech. When viewed in these terms user behaviour in the question-only condition was considerably more successful than would likely be the case with a real system (approximately 99% of responses being acceptable to the Wizard). The timing results therefore do not adequately model the likely effects of recognition accuracy differences across the different strategies.

Our own previous research has attempted to predict the recognition accuracy parameters under which menu prompting will outperform query prompting strategies [5]. This research involved simulating both recogniser performance and human input using task network models. The results indicate that it is actually very hard for menu strategies to compete with query strategies in terms of transaction time, even if the use of menus is assumed to significantly improve recognition accuracy. This therefore supports the transaction time findings of [4]. However in interpreting these findings it should be noted that the current models have several simplifying assumptions. In particular it is assumed that users will enter a valid keyword following a query prompt and will keep repeating that keyword following a recognition failure and reprompt by the system.

The Wizard of Oz study reported above [4] also looked at yes/no dialogues. The research found that both implicit yes/no prompts (e.g. "...Will you accept charges?") and explicit yes/no prompts (e.g. "....Will you accept charges? Please say yes, no, or operator, now.") elicited high user compliance (approximately 97% valid responses). However, these results are not directly comparable to the menu and question-only strategies used in [4] because, while the menu and query styles were used for choosing services, the yes/no prompts were only used for confirming inputs. In fact [4] distinguish between menu 'transactions' (which can use spoken menus or question-only prompts) and yes/no 'transactions'. The implication is thus that some types of user task will naturally require a menu, others a yes/no query. We, on the other hand, suggest that yes/no prompts can be viewed as being on the same continuum as menu and query modes, and can equally well be used for choosing options. Such selection would involve going through sequential lists where each service is offered in turn ("Do you want service A?" "No" "Do you want service B?" "Yes", etc.). This approach is used successfully in some commercial telephone systems, for example British Telecom's CallMinder system. However, it is unclear under which conditions this approach will be more successful than other alternatives such as menu and query prompts.

The current work was thus carried out to investigate the relative efficiency of yes/no, menu and query prompting strategies. It differs from the previous work reported here in two key respects. First the experiment uses an actual ASR device rather than relying on a simulation of ASR capabilities. Second the three styles of interaction are used to accomplish exactly the same task to allow direct comparison between them. The home banking domain was the application chosen for the experiment. Performance was measured in terms of task completion, transaction time and user satisfaction ratings.

2. METHOD

2.1 Participants

Forty-two participants were recruited from the Nottingham area (21 males, 21 females; mean age 23 yrs 3 months). All were UK nationals and were paid for participation.

2.2 Dialogue Design

Three dialogues were used in this experiment, one using mainly yes/no prompts, one using mainly menu prompts and one using mainly query prompts. Where these styles were not used (e.g. PIN number entry) identical prompts were used across the three different dialogues. The dialogues also differed in terms of the initial announcement given to users. In the yes/no dialogue users were told to "answer yes or no to the questions unless you are given other instructions". In the menu dialogues: "every time you are asked to say something you will be given a list of words to choose from. Say the option you want after the list has finished". In the query dialogues: "the system can recognize single words and short phrases. After each question say the service you want. For example, after the question 'which service do you want?', say 'order chequebook'."

All three dialogues were for a home banking application. The available services were checking an account balance, ordering a statement, paying a bill and ordering a chequebook. Example sections from each dialogue are given in table 2.

When the system did not recognize an utterance the user was given the message "the system did not recognize an input" and the previous prompt was then repeated. After three consecutive fails the user was given the message "the system cannot complete this service" and users were returned to a prompt where they could choose to continue or quit the system.

yes/no dialogue Prompt: do you want to hear your balance? User: no Prompt: do you want to request a statement? User: yes Prompt: do you require a statement for your current account? User: no Prompt: do you require a statement for your savings account balance? User: yes
menu dialogue Prompt: Say one of the following services: balance, statement others? User: statement Prompt: say which statement you require: current, savings, both? User: savings
query dialogue Prompt: Which service do you require? User: savings account statement/my savings account statement OR User: statement/account statement/order statement/send statement Prompt: Which account do you require a statement for? User: savings/savings account/my savings account

Table 2: Example sections of dialogue showing selection of a savings account statement.

2.3 Experimental Procedure

Participants were given four tasks to perform: finding out a savings account balance, paying off a credit card bill by transferring money from the savings account and getting a statement for both savings and current accounts. They were instructed to speak clearly and naturally, but were given no further instructions about how to speak or what to say.

Participants were free to quit the interaction at any time by saying "quit". After completing their interaction with the system all participants were asked to fill-out a questionnaire.

2.4 Apparatus

The dialogues were programmed in Visual Basic 5.0 running on a Pentium II PC. Speech recognition was achieved using Dragon Dictate 3.0 (British Edition) and was integrated into the Visual Basic project using Dragon X-Tools. Speech output was provided through human speech recorded as .WAV files. A headmounted VXI corp Parrot 10.3 microphone/headphone was used throughout.

2.5 Experimental Design

The experiment used a between subjects design with 14 participants using each dialogue style. Each group was balanced for gender.

Data was recorded on success (whether task was completed and proportion of task completed) and, for those who completed the tasks successfully, on efficiency (time to complete interaction, number of dialogue steps). Subjective responses were also collected.

3. RESULTS

3.1 Success of Interactions

Nine out of fourteen participants (64%) in the yes/no condition, six out of fourteen participants (43%) in the menu condition and three out of fourteen (21%) in the query condition completed all four tasks successfully. Chi Square analysis did not allow rejection of the null hypothesis that this distribution of results was due to chance.

The second measure of success was the number of sub-tasks completed within the dialogue (min 0, max 4). With the yes/no dialogue the mean number of tasks completed was 3 (s.d. 1.47), with the menu 2.6 (s.d. 1.45) and with the query 1.1 (s.d. 1.69). ANOVA shows significant differences between these means, F (2, 39) = 6.23, $p<0.05$. Post hoc Scheffe tests show significant differences between the query dialogue and the yes/no dialogue, and between the query dialogue and the menu dialogue.

3.2 Time to Complete Task

The mean time to complete the entire interaction using the yes/no dialogue was 233 seconds (s.d. 43.07; N=10), using the menu dialogue it was 273 seconds (s.d. 25.73; N=6) and using the query dialogue it was 215 seconds (s.d. 40.96; N=3).

A non-parametric Man-Whitney U test was used to compare the time on task for the menu and yes/no dialogues. (The query dialogue results were not included in this analysis as the sample which completed the entire interaction was so small (N=3)). The results indicated that the time to complete the task using the menu dialogue was significantly longer than the time to complete the task using the yes/no dialogue (U=9, $p<0.05$).

3.3. Subjective Opinion

Subjective opinion was assessed using a prototype questionnaire using 53 statements, each with a seven point rating scale from agree strongly to disagree strongly. Space does not permit full reporting of this data but several significant differences are worth noting.

ANOVA revealed significant differences between the three dialogues for the statement "I would use this system" , F(2, 39) = 3.68, $p<0.05$. Post hoc Scheffe analysis showed that the significant difference lay between the rating given to the yes/no dialogue (mean = 2.9; agree) and the query dialogue (mean = 4.91; disagree); the rating for the menu dialogue was mean = 4.25, (disagree).

ANOVAs also showed significant differences between the dialogues for questions which assessed users' confidence in knowing what to say to the system. For example for the statement "it is clear how to speak to the system"; F (2, 39) = 12.0, $p<0.001$. Post hoc Scheffe analysis showed the significant differences were between the yes/no dialogue (mean = 2.5, agree) and the query dialogue (mean = 4.9, disagree), and between the menu dialogue (mean = 2.1, agree) and the query dialogue.

4. DISCUSSION

Significant differences were found between the dialogues in terms of success, transaction time and subjective responses. Participants were most successful in the yes/no condition and least successful in the query condition. Timing data from completed interactions indicated that the yes/no dialogue was significantly quicker than the menu dialogue in terms of overall transaction time. In questionnaire responses, participants generally rated the query dialogue as significantly worse than the other dialogue styles, but no significant differences were found between the ratings for the menu and yes/no dialogues.

The results therefore suggest that where a small vocabulary system is used for naive users, constraining prompts which make it clear to users what they have say (i.e. yes/no or menu) are more effective than a style where users are not told the vocabulary in advance (i.e. query). Some of the recognition problems with the query dialogue were due to users trying longer phrases than the system would accept (as would be expected from [3]). However, in many instances users had in fact used a valid input, but mis-interpreted a rejection of this item as meaning that they had said something the machine could not accept, and on their next attempt they tried a new formulation of their request (sometimes acceptable, sometimes not). Thus in many cases performance was actually more disrupted by users' attempts at recovery from failed recognition than if they had persisted with their original input wording. This behaviour explains why the query strategy did not out-perform the menu strategy, a result which would have been predicted from both [3] and [4]. The subjective responses from participants suggested that users did not like the query style of dialogue because of both the poor performance and the uncertainty they felt about what to say.

Although the results showed no statistically significant differences between the menu and the yes/no dialogue styles in terms of success, more participants were able to complete the entire task with the yes/no style, and their interactions were significantly faster. Initially it seems surprising that the yes/no dialogue was faster than the menu, given that it involved several more steps (even with error recovery steps). However, the effect is explained by three key facts:

- individual yes/no prompts were shorter than menu prompts.
- yes/no recognition was more accurate than menu item recognition.
- only 3/4 options were offered per menu prompt (because of restrictions in user working memory).

While there were no significant differences in the subjective ratings for the yes/no and the menu styles, participants did on average agree that they would use the yes/no style and would not use the menu style.

Given these results it is suggested that designers of small vocabulary systems for untrained users might consider the yes/no style of interaction as a viable alternative to the menu style. The high user compliance with both of these strategies means that their behaviour can be easily modelled using the approach used in [5], so it is theoretically possible to predict the conditions under which the yes/no strategy will outperform the menu strategy for any application. Dialogue designers could use this approach to make informed choices about dialogue style for their applications.

5. ACKNOWLEDGMENTS

This research was supported by the UK Engineering and Physical Sciences Research Council, grant no. GL/L94710.

6. REFERENCES

1. Baber, C. "Developing interactive speech technology". In C. Baber and J.Noyes (Eds.) *Interactive Speech Technology*. Taylor and Francis, London, 1993.
2. Waterworth, J.A. "Interaction with machines by voice: human factors issues". *British Telecom Technology Journal, 2(4)*, 1984
3. Yankelovich, N. "How do users know what to say?" *ACM Interactions, 3(6)*, 1996.
4. Brems, D. J., Rabin, M. D. and Waggett, J. L. "Using natural language conventions in the user interface design of automatic speech recognition systems". *Human Factors, 37(2)*, 265-282, 1995.
5. Hone, K.S. and Baber, C. "Modelling the effect of constraint upon speech-based human-computer interaction". *International Journal of Human-Computer Studies*. (In Press).