

Reducing structure in spoken dialog applications

Kurt E. Dusterhoff

Novauris Laboratories UK, Ltd.

**Millbank
Stoke Road
Bishops Cleeve
Cheltenham
GL52 8RW**

www.novauris.com

kurt.dusterhoff@novauris.com

Abstract

Most spoken dialog applications generate a series of questions that force the user to break down his or her request into segments the system can easily handle. This division of a single request into multiple pieces slows down the interaction and makes it less natural for the user, with the result that such systems constitute only a small improvement on the touch-tone, menu-driven systems they are meant to replace. A system that allows the user to elicit the appropriate action using a single utterance can decrease the time spent on the interaction and increase user satisfaction. For example, the standard directory enquiry dialog flow asks for the city and state, processes that utterance, and then asks for the name. A typical client behavior, when speaking to an operator, is to ask for the name in a location. Systems that handle single utterances for multiple pieces of related information better emulate this natural interaction. Such an approach should reduce confirmation and repair dialog and allows the user to impart more information quickly, lowering the interaction time per user. A dialog plan that allows for longer utterances will normally need to allow for permutations of the informative text within a request. The combination of flexibility and reduced interaction time results in a faster, more accessible system and an improved client experience that should replace the menu-driven and structured dialog.

Introduction

Most spoken dialog systems attempt to match information from users with data accessible by the system. Typically, such systems rely on multiple requests to gradually narrow the search for data that corresponds to what a user has said. However, in many cases, the number of requests for information is excessive, wasting valuable time in the interaction. The two scenarios below show how a dialog that requests an address can be collapsed into a single interaction.

Scenario A

What city, please?

New York

What street, please?

Fifth Avenue

Please say the number as single digits. For example, say twelve as one two.

Four five five

Scenario B

Please say the full street address and city name of your destination.

Four fifty-five Fifth Avenue,
New York

Often data that belongs together is broken up so that the system can cope with, and sometimes confirm, each small piece. In these cases, one often feels that the system is at best as irritating as the DTMF applications they are meant to supersede. This paper discusses how and when multiple requests can be efficiently combined, allowing for a faster, more effective interaction.

The relationship between pieces of information being requested helps define the types of applications that can benefit from structure reduction. When asking for more than one piece of data, the information is generally either disjointed or cohesive. What we are calling “disjointed information” is inherently unrelated, like a vet’s name and the cat they treated yesterday. Cohesive data is inherently related, like the name of a book and its author. Disjointed information typically includes a piece of data related to the caller and a piece of data related to the reason for their call. For example, a company may want its callers to give their account number and the department they need to speak to. While both pieces of information are necessary for the transaction to take place, they are not necessarily related to each other. Cohesive information is generally all about the caller or their transaction. For example, an application may ask for the caller’s name and a part of their address. Where the information gathered from the caller is cohesive, there is scope for requesting all of the information with a single question.

Joining Queries for Cohesive Information

Before delving into how and why one might join queries for cohesive information, it is important to understand why many systems do not. Frequently, when a system designer has low confidence in parts of the application, like the speech recognizer, he may wish to minimize risk and use a small-vocabulary, word recognition framework. Similarly, where there is low confidence in the callers’ ability to follow directions or provide the information, system designers will try to direct the flow. However, when requesting cohesive information, the potential to exploit redundancy in the response, enhance user satisfaction, and cut down call times can outweigh the reasons for a heavily structured dialog.

Exploiting Redundancy

When information is cohesive, and accessible to the application, there is scope for exploiting redundancy within that information. Many current systems use grammars built on-the-fly to sift through the unique parts of each datum. An alternative approach uses the redundant pieces of information to cross-check hypotheses. For example, a call routing application may ask separately for the department and name for which a number is required. Most current systems would check the department, and use that information to determine which names can be requested. A system that uses redundancy in the information may also use the name requested to cross-check that the department was recognized correctly. Rather than make a decision based on part of the information, this method allows the application to make a decision from all of the information with a single request.

Enhanced Experience

Users prefer negotiating a transaction with a party that gives them the response they need from the information the user has to hand at the time. For simple requests, such as balance queries and the local weather forecast, a DTMF system may provide the most efficient solution. As the interaction becomes more complicated, the user's desire to speak to a human increases. One reason for the rapid disenchantment with technology is that the user must somehow negotiate a series of transactions, where really all they want to do is, for example, find out how to get to 455 Fifth Avenue, New York. The two scenarios above examine likely transactions. Given the same degree of repair and confirmation (none in both examples), the second interaction involves the user giving cohesive information as a response to one prompt. The first scenario, however, requires the user to break the information into pieces that the system can handle. The second scenario allows the user to interact more swiftly and directly than the first, and is probably more satisfactory.

Interaction Time

Viewed solely in terms of time, Scenario A will take longer than Scenario B. The caller's speaking time is essentially equivalent in both cases. Scenario A, however, spends time delivering extra prompts. Each prompt delivery takes time that could be better used or eliminated altogether. For example, using the prompt "*What street please?*" takes just over a second to play the prompt, and between 0.5 and 1.5 seconds to ensure that the caller has finished their response. Given these figures, Scenario A will take four to five seconds longer than Scenario B. If we look at these transactions as part of a larger interaction, involving multiple transactions, we can see that reducing each transaction by a few seconds could amount to significant savings for the service provider. Alternatively, if some of that time is required for additional computing, the application can be playing company information or advertising instead of playing prompts to the caller.

Successful Applications

Two successful applications of minimally structured dialog are AT&T's "How may I help you?" system (Gorin *et al*, 2002) and Novauris Laboratories' Name and Address Retrieval system. The two applications approach the information gathering exercise from different angles, but both attempt to minimize the number of questions required to complete transactions with the caller. The AT&T system uses topic- and word-spotting technology to find important words from a medium-sized vocabulary within the caller's utterance. The Novauris system requires the caller to adhere to the conventional structure of the data being requested, but allows an extremely large active vocabulary. Together, these two systems cover a substantial range of tasks for which spoken dialogue systems are employed. One is geared to selecting from a fairly small number of alternative next steps, which may include simple messages, DTMF interactions, human agents or further ASR. The other is geared to selecting entries from very large lists.

On the surface, the AT&T approach looks as though it can handle what we are classing as disjointed information. Certainly, one might expect a system that takes 'natural language' input to cope with a wide variety of information. However, this application

interfaces with essentially the same data as a nested menu application. The number of topics that can be distinguished based on the caller's utterance is quite limited. Instead of responding to a question about the department the caller wants to reach and another question about what they want that department to do, the caller simply says the two things at once. In most cases, this single utterance allows the application to compare the various topics it has found and respond according to the most likely combination of extracted words. For example, if the caller says "I'd like to speak to someone in Accounts about a refund for double-billed calls," the system might spot topics like 'accounts,' 'billing,' 'refund,' and 'I'd like to speak to.' Given this fictional topic list, the system is likely to respond related to the 'accounts' and 'refund' topics, because they fit the acoustics and are associated with each other and the list of response behaviors. Using this sort of technology is ideal for replacing two- and three-question transactions that use a fairly small vocabulary.

The Novauris approach, on the other hand, relies on cohesive information. The system asks callers to say the entry key whose data they want to retrieve. For example, the caller accesses data related to "James Smith, 1978 Main Street, Springfield, Nebraska, 00000" by saying just this. In an extension to the application, one could receive directions to 455 5th Street, New York by saying "455 5th Street, New York." Similar to Scenario B above, the information needs to be imparted by the caller with as much adherence to a common structure as possible. However, by exploiting the inter-relationships of the parts of the address, the application can explore many possibilities covering an enormous vocabulary. The Novauris application has an active vocabulary of hundreds of thousands of words. Depending on the type of data being requested, the application accesses between a few thousand and hundreds of millions of records. The technology underlying this application is ideal for extending the use of spoken dialog systems to large data access areas, like in-car navigation, directory enquiries, and customer accounts management.

Both of these applications reduce the structure of individual transactions. Obviously, where multiple transactions need to take place, the dialog will require more than a single prompt and response pair. In such cases, the potential time savings per transaction is additive. The improvement from the users' viewpoint is much more difficult to predict.

Obstacles

While reducing the structure in dialog systems can increase application uptake and enhance user satisfaction, substantial obstacles remain. The greatest obstacle to any successful dialog system is the ability of callers to provide the information requested. Applications that minimize structure must allow users to provide information with some flexibility. Extraneous speech has to be handled or discounted, and where structured input is required, prompts must clearly direct the caller to the appropriate behavior. The AT&T application outlined here gives flexibility by limiting the number of topics that it will spot. The Novauris approach attempts to direct the caller to perform a particular behavior, and focusses on applications where there are well-known conventions for structuring the information. A further enhancement to this application has been designed to allow increased flexibility to the input utterance, but the system is inherently a retrieval

application, and therefore requires callers to sufficiently identify the record to be retrieved.

Conclusions

Reducing structure from the dialog where the application is requesting cohesive information that uses known vocabulary will improve application uptake and user satisfaction. Redundancy in cohesive information can be exploited to improve response time and accuracy, resulting in an enhanced experience for users. By lowering the number of prompts played out, each transaction will save vital seconds on the traditional nested query approach. Overall, call time will be lower, successful throughput higher, and the user experience much improved.

References

Gorin, A.L., A. Abella, T. Alonso, G. Riccardi and J.H. Wright. "Automated Natural Spoken Dialog." IEEE Computer Magazine, volume 35(4), pp. 51-56. 2002