MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

SpokenQuery: An Alternate Approach to Choosing Items with Speech

Peter Wolf Joseph Woelfel Jan van Gemert
Bhiksha Raj David Wong

TR-TR2004-121 April 2004

Abstract

A majority of spoken user interfaces deal with the task of retrieving an element from a list. Conventionally, spoken UIs deal with such tasks through hierarchies of menus or dialogs, that navigate users through a series of steps, each of which present them with a limited set of choices. In a recent paper [2] we presented an alternative approach to such UIs, termed SpokenQuery, that recasts the problem of selection from lists as one of retrieval, and demonstrated that it could result in significantly lowered cognitive load on the user. In this paper, we examine varius aspects of retrieval from spoken queries, and UIs based on such retrieval, and demonstrate that in addition to reducing the cognitive load on the user, the system is effective for searching large databases, is robust to environment noise, and is effective as a UI.

International Conference on Spoken Language Processing ICSLP 2004

SpokenQuery: An alternate approach to choosing items with speech

Peter Wolf, Joseph Woelfel, Jan van Gemert, Bhiksha Raj, David Wong
Mitsubishi Electric Research Laboratories, Cambridge, MA, USA
{wolf, woelfel, gemert, bhiksha, wong}@merl.com

ABSTRACT

A majority of spoken user interfaces deal with the task of retrieving an element from a list. Conventionally, spoken UIs deal with such tasks through hierarchies of menus or dialogs, that navigate users through a series of steps, each of which present them with a limited set of choices. In a recent paper [2] we presented an alternative approach to such UIs, termed SpokenQuery, that recasts the problem of selection from lists as one of retrieval, and demonstrated that it could result in significantly lowered cognitive load on the user. In this paper, we examine various aspects of retrieval from spoken queries, and UIs based on such retrieval, and demonstrate that in addition to reducing the cognitive load on the user, the system is effective for searching large databases, is robust to environment noise, and is effective as a UI.

1. INTRODUCTION

Many common user interface tasks deal with the problem of choosing an item or a set of items from a large list. Common examples include selection of a command from a list of all possible commands, searching for the correct help documentation from the set of all help documents, and selection of songs from play lists. Most current systems use a combination of two approaches for such tasks: menus and search. If the list can be intuitively described as a hierarchy, then menus are used; if not, then the application uses a search box. Every computer user today is completely familiar with these approaches.

2. SPEECH-BASED USER INTERFACES

Spoken user interfaces for selection tasks have largely been menu-based for practical reasons: menus greatly restrict the perplexity of the grammar to be explored by the recognizer at any time, improving its size, speed and accuracy. Unfortunately, the very restrictions that improve speech recognition in these implementations can induce other problems:

1. The “What can I say?” problem - users are often unsure of available choices, and how to say them.
2. Misrecognition: Confused users may speak options that are not in the current menu. These can be misrecognized, resulting in erroneous response.

In addition, the use of speech compounds some problems that are inherent to menus in general. E.g., even in the best designed menu hierarchies, users sometimes cannot locate a desired option. In a speech-based UI, this is exacerbated by the fact that a user cannot quickly scroll through the options visually (as would be possible with a GUI).

Consequently, speech-based UIs often perform poorly, frequently actually increasing the cognitive load imposed on the user by underlying menu interfaces. From the users' point of view it seems that speech recognition is not yet ready for prime time. While some of the shortcomings of the menu-based approach may be resolved by engaging users in a dialog with the system, users often find this approach onerous for simple selection tasks.

In [2] we have presented an alternate approach to spoken user interfaces for selection from lists. This approach, which we term “SpokenQuery”, treats such tasks as one of document retrieval using spoken input. Every item in the list is modeled as a document. The user speaks a description of the desired item or items using whatever words seem appropriate. There is no grammar to memorize, and the system is robust to out-of-vocabulary speech. The result of the query is a short list of items judged to be pertinent to the query, sorted by estimated relevance. The final choice is returned to the user. There is no repeated interaction between the system and the user - the user's choice is returned in a single pass of querying and selection. If the desired response has not been obtained, a fresh query must be spoken. Figure 1. illustrates the difference between standard spoken user interfaces and SpokenQuery. Detailed user studies reported in [2] show that the SpokenQuery interface can impose a significantly lower cognitive load on a user, as compared to a standard menu-based interface.

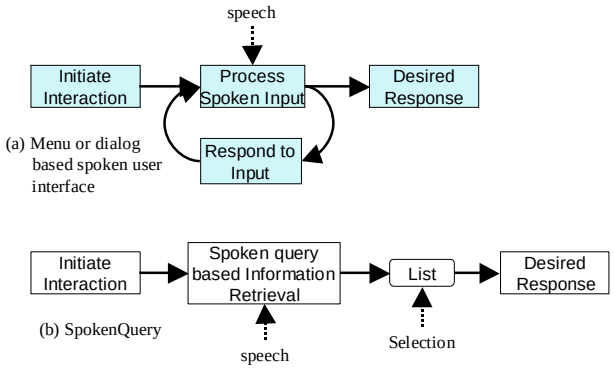

Figure 1. Difference between menu or dialog-based spoken UIs and SpokenQuery, an IR-based spoken UI. The former require repeated interactions between the user and the system, whereas the latter does not.

In this paper we extend these studies and conduct a fuller analysis of the system. We demonstrate that in addition to reducing cognitive load, the SpokenQuery UI can effectively respond correctly to a user under a variety of conditions, as measured by appropriately specified performance metrics.

Since SpokenQuery is a spoken user interface based on IR principles, any analysis of the system must evaluate both the IR and UI aspects. At the heart of SpokenQuery is a spoken input based information retrieval engine [4]. Conventionally, spoken input based IR has been implemented by converting spoken input to unambiguous text with a speech recognizer, to query a text-based IR system. However, speech recognition can be highly error prone, especially when the input language has few restrictions, as is the case for a SpokenQuery UI. Simply applying standard text-based IR to the text transcription output by a recognizer may not be effective [1]. To account for the uncertainty in the recognizer output, SpokenQuery utilizes the entire search space of the recognizer, to construct queries for IR.

In addition, since SpokenQuery is meant to be a UI, the presentation of the output of the IR is constrained by the limitations of the particular UI for which it is intended. Also, the goals of IR are more stringent - it is not sufficient for the returned responses to be pertinent to the topic of the query; rather, they must enable the specific interaction desired by the user.

The experiments presented in this paper attempt to analyze both the IR and UI aspects of SpokenQuery in an integrated manner. Various aspects of the IR system are evaluated in the context of a UI to an example business-address-finder application. The results obtained, when considered in conjunction with the user studies reported in [2], indicate that SpokenQuery is an effective and viable alternative to menu-based interfaces.

The rest of this paper is arranged as follows: in Section 3 we briefly outline the implementation of SpokenQuery. In Section 4 we describe our performance metrics, in Section 5 we present experimental analyses of the system, and in Section 6 we present our conclusions.

3. IMPLEMENTATION OF SPOKENQUERY

SpokenQuery is implemented as a combination of speech recognition and vector based information retrieval. The items to be retrieved are modelled as documents, that can be indexed by meta data or text descriptions. The spoken input is converted to a query vector that is matched against the index to produce a relevance ranking of the documents. Both documents and queries are represented as bags of words. The ordering of words is not currently used in the document ranking process.

In SpokenQuery, the spoken input is decoded by a recognition system to generate a word-level lattice. The word frequencies in the lattice are weighted by their *a posteriori* probabilities to form a query vector. The *a posteriori* probability of a word is the ratio of the total likelihood of all paths through the node representing that word to the total likelihood of all paths through the lattice. The dot product of the query and document vectors can produce a score, which can then be used to rank the documents.

The top ranked documents are then presented to the user. Documents may be presented either through a display, or by speaking them out to the user via a text-to-speech convertor. The number of documents to be presented can be set either by design choice, or on the basis of a relevance score threshold. If the document/response desired by the user is in the presented list, the user can select it by one of various modalities, such as pressing a button. If not, the interaction must be repeated.

4. PERFORMANCE METRICS

SpokenQuery differs from traditional IR interfaces in an important manner: it is intended to function as a user interface to applications that involve selection from lists. As a result, unlike conventional IR systems where the “correct result” for a query is usually subjective, the correct result for SpokenQuery can either be a specific entry in the list, or a set of entries that the user wishes to peruse.

A second factor is that devices that benefit most from spoken UIs are usually small, with tiny screens that can display no more than a few lines. In the extreme case, such as in devices mounted in cars where the user is not at liberty to divert their attention to screens, the system must convey information to the user via a speech synthesizer. Such output modalities are most effective when the number of choices presented to the user is small.

The metrics chosen to quantify the performance of SpokenQuery consider both above factors. In all experiments, except where stated otherwise, queries have been formulated from the perspective of a user who requires only a specific response from the system (as would be the case for most UI interactions). It has been assumed that system responses must be shown in a display of limited size. The closer the *correct* response is to the top of the returned list, the more correct the system has been. The performance of the system has been measured in terms of the “accuracy of the response”. The reported accuracies measure the fraction of interactions in which the desired item is present in the returned list. For instance, in a display of size 5, the response is deemed 100% accurate if the correct response is ranked five or higher. Any interaction between the system and the user where the desired item did not appear within the display was deemed 0% accurate. This measure makes sense from the perspective of a real user using a real system. The reported accuracies are average accuracy measured over several interactions. The experiments we perform measure the following:

- Accuracy in display: The accuracy of the information presented to the user in response to a query, in a standard automotive application.
- Robustness of the IR system to environmental noise: this attempts to measure the effect of background noise on the accuracy of the spoken UI. This is an important test for any speech-based UI that is likely to be deployed in any real-life scenario.
- Robustness of IR to query formation: This measures the relation between the extent of information presented by the user, and the returned responses.
- Precision vs. Recall: Relating the fraction of all relevant responses retrieved, to the retrieval accuracy of the retrieved responses. This standard metric is relevant to the case where a user desires a *set* of responses, rather than a specific response.

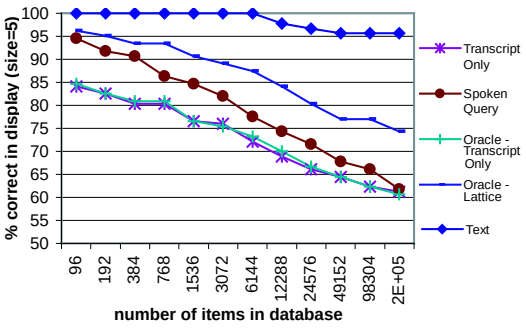

Figure 2. Accuracy in display

5. EXPERIMENTAL RESULTS

All experiments reported in this section were carried out on a spoken UI for a simulated business-address finder. The aim was to obtain the addresses of specific directory entries pertaining to given queries. A directory of up to 200,000 businesses and addresses was used for the experiment. The test set was a corpus of 500 spoken queries collected in a quiet laboratory and in a moving automobile. A far-field microphone was used to record the queries. The speech recognition system used for the UI was the CMU Sphinx-3.3 recognizer, trained on 65 hours of broadcast news data.

5.1. Accuracy in display

In this test, we measure the overall accuracy of the SpokenQuery system. The test set used in this experiment consisted of 200 queries recorded in a moving automobile. The display size was assumed to be 5 lines. The size of the document data base ranged from 96 to 196608 entries.

Figure 2 shows the results of this experiment. In the figure, the curve labeled “text” represents the response achieved with retrieval based on the true transcription of the queries. Thus, this curve represents the performance that can be achieved using an IR-based UI for the task, when the speech recognizer is perfect, i.e. it makes no mistakes and includes no spurious words in its search space. Therefore, “text” queries define the best possible performance for an IR-based UI for the task.

The “Transcript Only” curve represents the naïve method of performing Information Retrieval from a spoken query - simply take the normal transcript output from the speech recognizer and use it as a text query.

In the curve labeled “Oracle, Transcript only”, queries have been formulated from the single best transcript output by the recognizer. However, only words in the transcript that were actually spoken in the query are counted. This represents an ideal confidence scoring mechanism that can somehow selectively assign zero confidence to all non-query words in the transcript. The curve therefore represents an estimate of the upper bound of the performance achievable from the recognizer's output transcript alone.

For the curve labeled “Oracle Lattice”, queries are formed selectively from all the query words that are present in the recognition lattice. Words in the lattice that were not actually spoken are ignored. This represents an idealized lattice processing mechanism that selectively sets the weight of all non-query words. The curve represents an estimate of the upper bound of the performance achievable by SpokenQuery.

As can be seen from Figure 2., the Oracle curves provide strong evidence that the Lattice contains more information for selecting documents than the best transcription alone, and better search accuracy is possible making optimal use of a Lattice result.

The curve labeled “SpokenQuery” shows the performance obtained with SpokenQuery. SpokenQuery performs better than “Transcript Only”, but not as well as “Oracle Lattice”. It is reassuring to see that SpokenQuery also performs better than “Oracle, Transcript Only”.

In a separate experiment, not shown here, it was observed that little accuracy is gained beyond a display size of 5. The desired response typically either appears in the top 5 entries of the returned list, or not at all. This characteristic was observed across multiple tasks. While the actual number 5 could be a feature of the tasks and the recognizer used, this does indicate that the IR based approach to spoken UIs can be supported even with minimal displays.

5.2. Noise robustness of SpokenQuery

The aim of this experiment was to measure the robustness of the SpokenQuery UI to environmental noise. For this experiment, we recorded a corpus of queries in a quiet office and then artificially added white noise to the signal. We then measured response accuracy in a display of size 5, as a function of the signal to noise ratio of the spoken queries. The results of this experiment are shown in Figure 3.

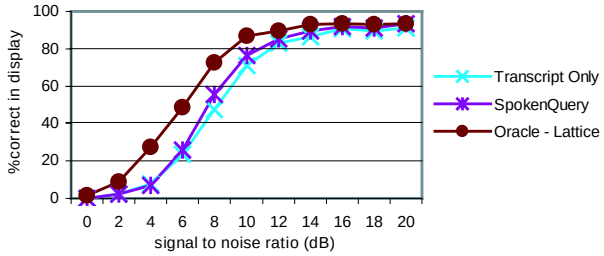

Figure 3. Accuracy as a function of noise.

Figure 3. shows that the performance of SpokenQuery is largely dominated by that of the recognizer. At high SNRs, the recognition accuracy is very high, and there is no significant difference between formulating queries with the recognizer's text output, or the recognition lattice. At very low SNRs, both the recognizer output and the lattice have few of the query words, and neither approach is effective. At intermediate SNRs, retrieval based on the lattice can be significantly superior to that obtained with the text output of the recognizer, as indicated by the Oracle curve. However, SpokenQuery itself is only slightly better than text-based retrieval, indicating that the lattice scoring mechanism used by the algorithm can be significantly improved.

5.3. The dependence of accuracy on query formulation

The aim of this experiment was to evaluate the effect of variations in query formulation on the accuracy of the sys-

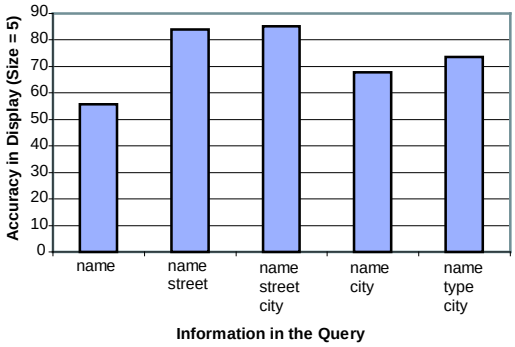

Figure 4. Accuracy as a function of query type

tem. To minimize the effects of ASR variations with speaker, the experiment was performed on queries recorded by a single male voice. A total of 87 queries were recorded in an office (with a far-field microphone) for each of the 5 tests, totaling 435 queries in all. These utterances describe the same businesses with different combinations of words. The database contained 200,000 items.

Each item in the database has several pieces of meta-data associated with it: a name, a type, a street, and a city. The experiment tested the effects of speaking different subsets of this information. E.g, one might speak just the name, or the name and the city.

Figure 4. shows the results of the experiment. As expected, speaking more information was always better than speaking less. The more specific the information in the query, the higher the rank of the correct item in the returned list. Nevertheless, even when only a few of the descriptive fields are spoken, the correct item often appears on the display, provided the information spoken is sufficiently distinctive.

5.4. Precision vs. Recall

Often, users desire one of a class of items, instead of specific items. For instance, a user might query for "Chinese Food", or "Hospital". In these cases the user will generally be satisfied if the result contains many of the correct class of items. In this situation, simple accuracy measurements, that chiefly measure precision, are not sufficient - it is equally important for the system to have high recall. There is an inevitable trade-off - recall can be increased by increasing the size of returned lists, but this usually results in lower precision, and vice versa. In this experiment, we evaluate the precision and recall of SpokenQuery, and text based retrieval. The experiment was performed on the same corpus used in Section 5.3. Figure 5. shows the results.

We observe from figure 5. that although spoken query has a lower equal-error-rate (EER) (where precision equals recall) than retrieval with typed in text queries, the EER is nevertheless close to 80%. For most practical tasks, this implies that in an appropriately sized display, the user would find the large majority of returned responses to be useful. While Precision/Recall curves cannot predict how satisfied actual users will be a given instance of the UI, since the display sizes required for the EER to be achieved are rarely available, they are nevertheless very useful for comparing retrieval engines, tuning and optimization.

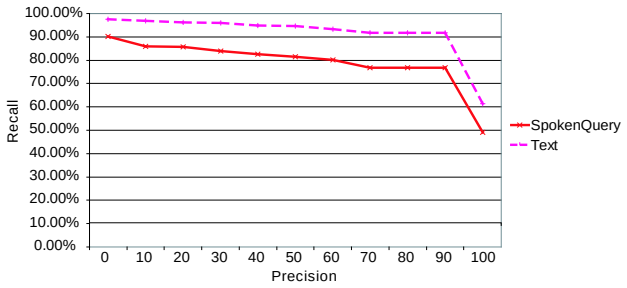

Figure 5. Precision vs. Recall

6. OBSERVATIONS AND CONCLUSIONS

The experiments reported in this paper indicate that the response desired by the user is returned over 80% of the time for realistic tasks. If the user does not suceed in retrieving the desired item the first try, they can repeat their query, possibly differently. Users can expect to obtain the desired response within three repetitions of the query 99% of the time. This compares favourably with a menu-based system where a standard interaction always requires multiple inputs by the user, even when the recognizer is 100% accurate! When considered in conjunction with the usability studies reported in [2], this shows that a SpokenQuery based UI can indeed be very effective, and may in fact be superior to a menu-based UI.

The Oracle experiments also show that the complete potential of the SpokenQuery approach remains to be tapped. Future research will attempt to close the gap between current performance and Oracle performance. The overall performance of the system is also expected to improve by moving from a bag-of-words model to one that considers word N-tuples (e.g. word pairs), since the interrelationship between words is a crucial component of response descriptions. The current recognizer is a word-based recognizer that utilizes a general N-gram language model. Superior performance can be expected by using task-specific grammar-based language representations. The problem of flexible vocabularies can be tackled by recognizing word particles [3], rather than complete words. Future research will explore all these avenues.

REFERENCES

1. Barnett, J. *et. al.*, "Experiments in Spoken Queries for Document Retrieval", Proceedings of Eurospeech '97, pp. 1323-1326, Rhodes, Greece, 1997.
2. Forlines, C. *et. al.*, "Spoken queries for in-car digital music selection", submitted to the ACM Symposium on User Interface Software and Technology,2004.
3. Van Thong, J. M., Whittaker, E.W.D.W., Moreno, P., "Vocabulary independent speech recognition using particles", Proc. Automatic speech recognition and understanding workshop (ASRU), Trento, Italy, Dec. 2001.
4. Wolf, P. and Raj, B., "The MERL SpokenQuery Information Retrieval System: A System for Retrieving Pertinent Documents from a Spoken Query", Proceedings of ICME, Vol. 2, pp. 317-320, August 2002.