Spoken Query for Web Search and Navigation

Huixiang Gu, Jianming Li
Shanghai Jiaotong University
ghxiang@sina.com
Jianming119@263.net

Ben Walter
M. I. T.
bwalter@alum.mit.edu

Eric Chang
Microsoft Research China
49 Zhichun Road, 5F
Beijing, China 100080
echang@microsoft.com

ABSTRACT
Mobile devices will become an important platform for Internet access. Due to size constraints, in many circumstances speech is the most desirable mode of input. We have developed a system for spoken query based web navigation and searching. The spoken query is recorded by a lightweight client and transmitted to a server where the computationally intensive continuous speech recognition and query execution are performed. Lightweight clients have been developed that can either be a small downloadable Active X component running within a browser or a small application running on a handheld PocketPC. Through these interfaces, users can search for contents in an Encarta encyclopedia, content of a particular website, or navigate to popular websites.

Keywords
Mobile devices; Multimodal User Interface; Information Retrieval; Speech Recognition.

1. INTRODUCTION
The next generation of Internet devices will be mobile. The number of mobile handsets already exceeds the number of desktop computers. As existing technologies such as WAP and NTT DoCoMo's i-Mode proliferate, the mobile web will become increasingly more ubiquitous.

A major barrier to usability of these mobile platforms is their user interface. Because of their small form factor, tiny keypads or small styluses are typically used today, and text input is inconvenient, especially when searching for information [1]. Given the constraints of existing input modalities, speech provides a compelling solution. We envision that the next generation of mobile devices will include a multi-modal user interface, with keypads, styluses and touch screens used for selection type actions, and speech used for data entry [2].

We have developed a distributed client-server search system that uses a combination of spoken queries with a traditional UI for navigation of the search results. Previous work on using voice for web browsing concentrated on using voice to access items on one's favorites list and selecting hotlinks on the current page. While the voice driven browsers are definitely useful in cases when the keyboard is not accessible, when a keyboard is available, the gain from using voice to drive the browser is more limited. In this paper, we focus on using voice to browse and search for information on the Internet. The distinction is that the information to be browsed or searched reside on a server rather on the local machine used for browsing. Our client uses an ActiveX component to digitize speech, which is sent to a server application that uses Microsoft's SAPI SR engine to process the sampled speech [3]. Our system uses automatically constructed index and can operate in a Chinese or English mode. We have also developed a client that runs on a PocketPC which communicates to the server through a wired or wireless network connection.

Past work on voice-controlled web browsers has focused on using voice control as a replacement for a point and click user interface [4]. For example, a user may say "Back" and "Bookmark this page" or activate hyperlinks and hot list entries by reading their text. None of the systems we examined support dictation to complete form fields, which is precisely the most difficult task to accomplish on mobile devices.

From the Information Retrieval community, the most relevant area of research is Spoken Document Retrieval (SDR). SDR is concerned with the recall of spoken documents, such as broadcast news or voicemail. SDR is essentially the inverse problem to our system, which is termed Spoken Query Retrieval (SQR)

2. SYSTEM DESCRIPTION
Figure 1 illustrates the current system. The client can be connected to the server through a wired or wireless network. The server returns result to spoken queries through matching the result of a speech recognition system with indices automatically generated by a separate crawler and indexer.

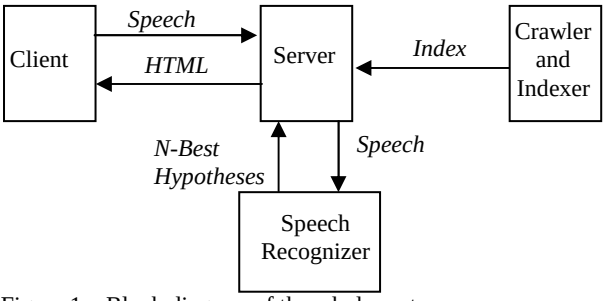

Figure 1. Block diagram of the whole system.

2.1 Crawling for Content
The primary index used by our system is a mapping from URLs to sets of optionally weighed keywords, with the index stored as a tab and space delimited flat text file. This choice of representation allows us to easily create indices during early phases of development. A website is crawled by a crawler, the html pages are sent to the indexer for indexing.

2.2 Building an Index
Once the crawler has retrieved the HTML content to the file system, keywords are extracted from the text of the HTML.

First, the HTML is parsed, and blocks of text are extracted. The blocks of text are then segmented into tokens. During the parse of HTML, the title of the page, if present, is extracted and recorded.

For English pages, the blocks of text are segmented into tokens based on white space and punctuation. Any purely or partially numerical (such as "y2k") terms are discarded, and words are converted to lower case. Stopword removal is performed using a list of 575 terms from the SMART system [5]. For Chinese pages, the blocks of text are segmented using a segmenter developed internally. During segmentation non-Chinese terms are

discarded, and after segmentation any numerical terms are removed. Stopword removal is also performed using a list of 570 commonly occurring Chinese words [6].

After keyword extraction has been completed, a flat text file containing a mapping from URLs to keywords and their frequencies of occurrence is created. During this procedure, the keyword frequencies are transformed to weights according to the *tfidf* metric, one of the commonly used term-weighting metrics from Information Retrieval.

The output of the keyword weighting procedure is a file where each term is assigned a decimal normalized weight, and another file that contains the inverse document frequencies for each term in the vocabulary.

The server performs the computationally intensive continuous speech recognition, and executes retrieval using a vector space model. The server accepts TCP connections on a pre-specified socket. When the client connects, it transmits the audio data to the server. The server then passes the audio data to SAPI based speech recognition server, which performs the CSR. The output of SAPI SR server is used to form the query, which is executed in a vector space model search engine. The hits are sent to client on the same TCP connection in text format, and then the server closes the connection.

Speaker-independent continuous speech recognition is performed by Microsoft's speech recognition engine, which is accessed through Microsoft Speech API (SAPI). The SR engine operates in dictation mode, with lexicon adaptation performed using the keywords extracted from the website. Because current SR Engines are monolingual, the server can currently answer either Chinese or English queries, but the languages cannot be intermixed.

We use the vector space model for information retrieval, where documents (URLs) and queries are represented a vectors in high dimensional spaces where each dimension corresponds to a keyword. To execute a query, the similarity of the documents to the queries is computed by taking the cross product of word frequency vector in the query sentences and the word weighting vector for each document. The top N-ranked documents are returned to the client in a buffer, whose format is the title, followed by a tab character, followed by the URL.

2.3 Client Processes

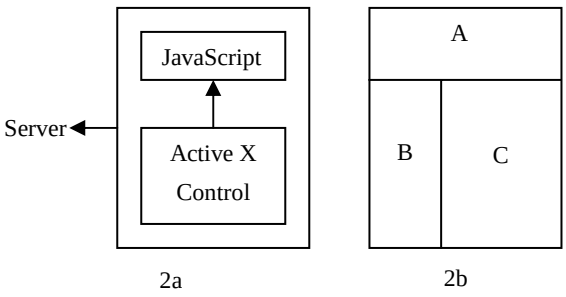

Figure 2. 2a shows the architecture of the client, 2b shows the appearance of the client to a user of the system.

The Voice Search Client is a combination of HTML, JavaScript and an ActiveX component written with VC++. The architecture and visual appearance of the client are shown in Figures 2a and 2b. A frameset with three frames constitutes the user interface. Frame "A" contains the ActiveX control and instructions on how to use the system. Frame "B" is used to display the results of a search. Frame "C" is used to navigate to the pages in the result set.

To perform a query, the user clicks the left mouse button on a button on an ActiveX control and speaks the query. As the user speaks, they can see visual feedback in the form of a waveform monitor in the ActiveX control. The speech data is transmitted to the server on a pre-specified TCP port. The ActiveX control downloads the results returned by the server and raises an event in the JavaScript.

A JavaScript fragment responds to the event raised by the ActiveX control, and retrieves the raw results from the control. The JavaScript formats the (title, url) pairs returned by the server into hyper linked HTML and shows them in frame B.

3. RESULT AND DISCUSSION

The work that we carried out thus far has demonstrated the feasibility of spoken query web search. Users can ask for information from an Encarta website such as "*Tell me something about cats"* and receive an HTML page containing links to articles related to cats. Currently, the accuracy of spoken queries is not as good as perfect text input due to misrecognitions. However, the ease of speaking can persuade the user to speak more search words, which can improve retrieval results. We also plan to study techniques for improving the accuracy of speech recognition, such as updating the language model used by the speech recognition engine in addition to updating the lexicon in a domain dependent fashion and indexing the content of the server in multiple levels of detail, ranging from phones, syllables, words, to phrases

The preliminary signal processing can be incorporated into the client, including feature vector computation [2]. If the processing is done on the client side, the bandwidth requirements are modest, and could be supported by present generation of cellular networks. This is an area we plan to pursue in future work.

Larger scale user studies need to be conducted to see how users would interact with a spoken language search engine. We may be able to identify a small collection of query formulations that could be incorporate into a hybrid CFG/LM CSR system developed for the MiPad project [2]. This will provide the system with the ability to better handle common phrases such as "What is" and "Tell me". We will also study how users interact with a mobile device in a multimodal fashion, where by allowing the user to speak the query and point to the correct result among the list provided by the server, the most efficient form of interaction becomes possible.

4. REFERENCES

[1] Buyukkokten O. et al., "*Focused Web Searching with PDAs*", The 9th International WWW Conference, Amsterdam, The Netherlands, May 15-19, 2000.

[2] Huang, X. et al., "*MiPad: A Next Generation PDA Prototype*", ICSLP2000.

[3] Microsoft Speech Application Program Interface (SAPI) Version 5.0, http://www.microsoft.com/speech.

[4] http://www.conversa.com.

[5] Buckley C., "*Implementation of the SMART information retrieval system*", Technical Report, #85-686, Cornell University, 1985.

[6] Kwok, K.L. "*Comparing Representations in Chinese Information Retrieval*", Conference on Research and Development in Information Retrieval, ACM-SIGIR, pp.34-41.