SPEECHBOT: AN EXPERIMENTAL SPEECH-BASED SEARCH ENGINE FOR MULTIMEDIA CONTENT ON THE WEB

Jean-Manuel Van Thong, Pedro J. Moreno, Beth Logan, Blair Fidler, Katrina Maffey and Matthew Moores

JM. Van Thong, P.J. Moreno and B. Logan are with Compaq Computer Corporation Cambridge Research Laboratory, One Cambridge Center, Cambridge,MA 02142, USA. E-mail: Jean-Manuel.VanThong@compaq.com,Pedro.Moreno@compaq.com,Beth.Logan@compaq.com

B. Fidler, K. Maffey and M. Moores are now with AgileTV, PO Box 3827, Robina Town Centre 4230, Queensland, Australia. E-mail: bfidler@oz.agile.tv,kmaffey@oz.agile.tv,mmoores@oz.agile.tv

Abstract

As the Web transforms from a text only medium into a more multimedia rich medium the need arises to perform searches based on the multimedia content. In this paper we present an audio and video search engine to tackle this problem. The engine uses speech recognition technology to index spoken audio and video files from the World Wide Web when no transcriptions are available. If transcriptions (even imperfect ones) are available we can also take advantage of them to improve the indexing process.

Our engine indexes several thousand talk and news radio shows covering a wide range of topics and speaking styles from a selection of public Web sites with multimedia archives. Our Web site is similar in spirit to normal Web search sites; it contains an index, not the actual multimedia content. The audio from these shows suffers in acoustic quality due to bandwidth limitations, coding, compression, and poor acoustic conditions. Our word-error rate results using appropriately trained acoustic models show remarkable resilience to the high compression, though many factors combine to increase the average word-error rates over standard broadcast news benchmarks. We show that, even if the transcription is inaccurate, we can still achieve good retrieval performance for typical user queries (77.5%).

Keywords

Speech Recognition, Multimedia, Web Search.

I. Introduction

As the magnitude and use of multimedia content on the Web grows, in particular large collections of streamed audio and video files, efficient ways to automatically find the relevant segments in these multimedia streams are necessary. Unfortunately, traditional Web search engines are often limited to text and image indexing and many multimedia documents, video and audio, are thus excluded from classical retrieval systems. Even those systems that do allow searches of multimedia content, like *AltaVista* multimedia search and *Lycos MP3* search, only allow searches based on data such as the multimedia file name, nearby text on the Web page containing the file, and meta-data embedded in the file such as title and author. Clearly these systems do no perform any detailed analysis of the multimedia content.

Except for some carefully archived files, most multimedia archives on the Web are simple lists of links to long audio files, sometimes several hours in length [1]. Very often, there is no transcription available and therefore no simple means for indexing their content.

[1] see for example *http://www.broadcast.com*

Even when a transcription is available often it is not annotated or linked to the relevant points of the multimedia stream. A searchable index that provides the ability to play the segments of interest within the audio file of these shows would make these archives much more accessible for listeners interested in a particular topic.

A straightforward approach to solve this problem consists of generating the transcription automatically using a large vocabulary speech recognition system. However, speech recognition technology is currently inherently inaccurate, particularly when the audio quality is degraded due to poor recording conditions and compression schemes. Despite this, we show that we can achieve accuracy satisfactory for indexing audio from the Web if the acoustic and language models are properly trained.

SpeechBot is not the first system to offer these capabilities. In fact, there have been several studies which had similar goals [1], [2], [3], [4], [5]. We differ from these projects in several ways. First, we fetch the audio documents from the Web and build an index from that data. Second, we don't serve content, but rather keep a link to the original document, similar to traditional search engines. Third, our system is designed to scale up on demand. Finally, our search index is available and running on the Web[2].

The content currently indexed is popular talk radio, technical and financial news shows and some conference video recordings. For video streams, since no video processing is needed, we only download the associated audio track, hence reducing the bandwidth in the transcoding, and saving disk space and bandwidth. These shows are almost entirely speech, and very few of them have associated transcriptions, unlike TV shows that are often closed captioned in the U.S. Closed captions (or CC) are the textual transcriptions of the audio track of a TV program, similar to subtitles for a movie. The U.S. Federal Communications Commission (FCC) requires broadcast programs to be close captioned in order to be accessible to people with disabilities. These closed captions can easily be used for indexing purposes since the transcription is time aligned for display purposes.

Some of the radio shows we index are extremely popular: Dr. Laura is syndicated to 430 stations in the U.S. and is reported to have a weekly audience of 18 million people; Art Bell Coast to Coast/Dreamland is reported to have a weekly audience of 10 million

[2] *http://www.compaq.com/speechbot*

people. Some of these shows are very long. For example each Art Bell Coast to Coast show is four hours long, and each Dr. Laura show is two hours long. However, none of these shows have existing transcriptions, except in some cases a title for each show. This is clearly inadequate since long shows may cover a variety of different subjects and should not be indexed as a whole. Since these shows all have existing archives on the Internet, *SpeechBot* provides links to the relevant segments in these archives.

The outline of the paper is as follows. In Section II we give an overview of the architecture of the system. In Section III we present a performance analysis. In Section IV we describe our usability studies. Finally, in Section V we present our conclusions and suggestions for future work.

II. System Overview

SpeechBot is a public search site similar to AltaVista, which indexes audio from public Web sites such as *Broadcast.com, PBS.org*, and *InternetNews.com.* The index is updated daily as new shows are archived in their Web sites. The system consists of the following modules: the transcoders, the speech decoders, the librarian database, and the indexer. Figure 1 presents the system architecture at a conceptual level. Figure 2 presents the overall architecture of the system and gives an idea of the hardware requirements of *SpeechBot.* We describe each component below.

A. Transcoder

The transcoders fetch and decode video and audio files from the Internet. For each item, they extract the meta-data, download the media documents to a temporary local repository and convert the audio into an uncompressed file format (typically Microsoft PCM wav). This format preserves the original sampling rate of the stream and the number of audio channels. The meta-data (if available) contains information about the file downloaded such as the sample rate, copyright, the story title, and possibly a short description. This information is used by the Librarian database to identify and track the document while being processed by the system, and to display the result of the query to the user.

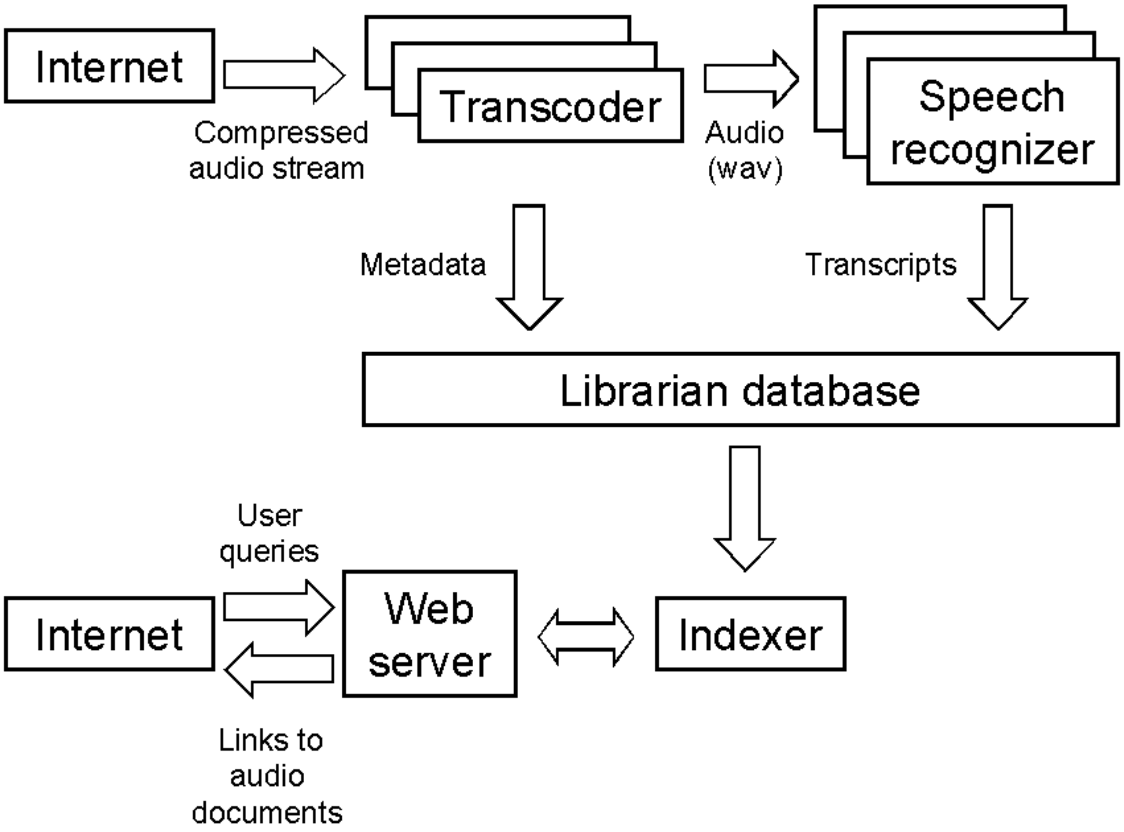


Fig. 1. Overall architecture of the system

B. Speech Recognizer

SpeechBot uses Calista, our in-house speech recognizer. Calista is derived from the Sphinx-3 system [6] and is a large vocabulary continuous speech recognition package which uses state-of-the-art Gaussian mixture, triphone based, Hidden Markov Model (HMM) technology. The best hypothesis produced by this one-pass recognizer is used to produce a textual transcription, or annotation stream, from the downloaded audio files. The annotation stream consists of the start time and end time of each word in the transcript, and the word itself. The audio files are segmented in such a way that the speech recognition can be performed in parallel on different portions of the document.

A farm of workstations tied together in a queuing system recognizes each portion of the document. Thus, even if the speech decoder is not real-time, we can still achieve sub real-time throughput. When all the portions are recognized, the results are assembled to create

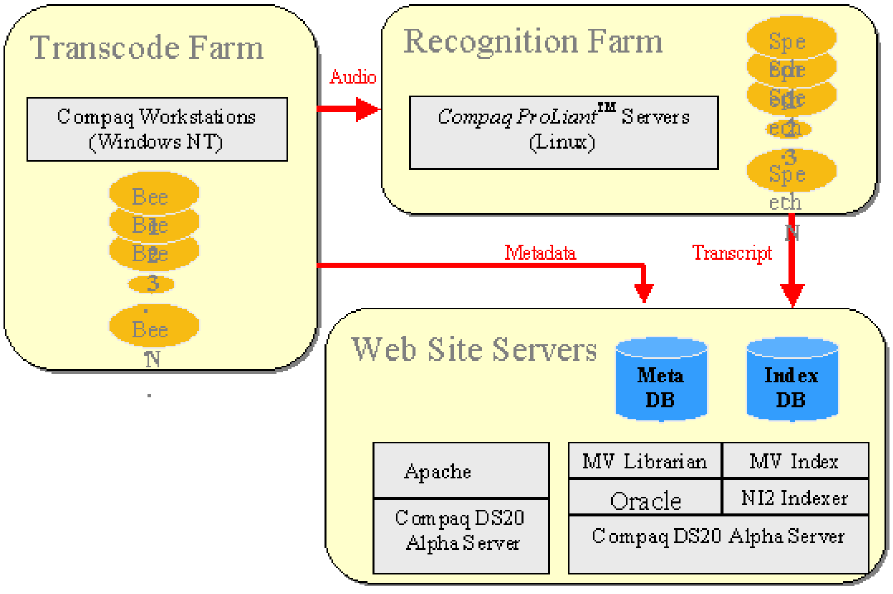


Fig. 2. Overall hardware view of the system

a fully annotated document. Calista yields an error rate of about 20% on a single pass search on the 1998 ARPA HUB4 evaluation corpora [7] with an average computational load of 6 times real time on a Compaq DS20 EV6 Alpha processor running at 500 MHz. The production system is composed of a farm of 30 Compaq AP500 and Proliant 1850 dual Pentium II/III, 450 MHz and 600 MHz, 256 Mb RAM, running Linux 6.0.

When the transcription is available we replace the speech recognition module with an aligner module. Its role is to provide time marks for each word of the input text. We have developed a method for aligning transcripts to very long audio segments. This method is robust to occasionally inaccurate transcripts, music insertions, or noisy speech. The precision measured is over 99% of words aligned within a 2 seconds misalignment [8].

In other cases the transcription is only approximate. This may be detected when the aligner module fails. Some words in the transcript might be missing or wrong words might be inserted. Sometimes even complete sentences might be missed. Following the ideas of [9] we build a language model using all the available text in the transcript and then recognize the audio using this model. This training provides a language model that is much more specific than general broadcast news models while allowing some freedom in the search. Because this model is very restrictive and close to the truth it limits dramatically the search space for the recognizer and yields lower word error rates. However, since we are performing recognition, not alignment, errors are still possible.

C. Librarian

The librarian has two main roles. It manages the workflow of tasks carried out by the individual modules, and it stores meta-data and other information required by the user interface.

The component modules often run on remote machines and do not communicate directly. Each process registers itself with the librarian and once registered can make a request for work to perform. This includes such tasks as speech recognition, text to audio alignment or insertion of text into the index. The output of one task is usually the input for another, and the librarian tracks the location of these inputs and outputs. In this sense the librarian is the brain of our system.

In addition to storing the meta-data collected by the Transcoder module, the librarian stores 10 second 'clips' of formatted text for each document. It maintains a mapping between word locations from the index, corresponding text clips, and the time the clip occurs in the multimedia document. The UI uses this information to construct the query response pages displayed to the user.

The librarian is built on an Oracle relational database running on Tru64 Unix 4.0. Clients communicate with the librarian via a in-house, clear text protocol over socket connections. The use of a central repository for shared information allows a robust distributed architecture which can scale on demand.

D. Indexer

The indexer provides an efficient catalogue of audio and video documents based on the transcription produced by the speech decoder. As well as supplying the user interface with a list of documents that match a user's query, the indexer also retrieves the location of these matches within the documents. It does this using a modified version of the AltaVista query engine [10].

The indexer sorts the matches according to relevance, as described in Section III-C. We define relevance using the term frequency inverse document frequency (tf.idf) metric [11], adjusted for the proximity of the terms within the document. This sorting is performed on both the list of documents returned and the list of match locations. In addition to the transcription of the audio, we also index the meta-data if any is available. Both the librarian and the indexer are based on the system described in [12].

E. User Interface

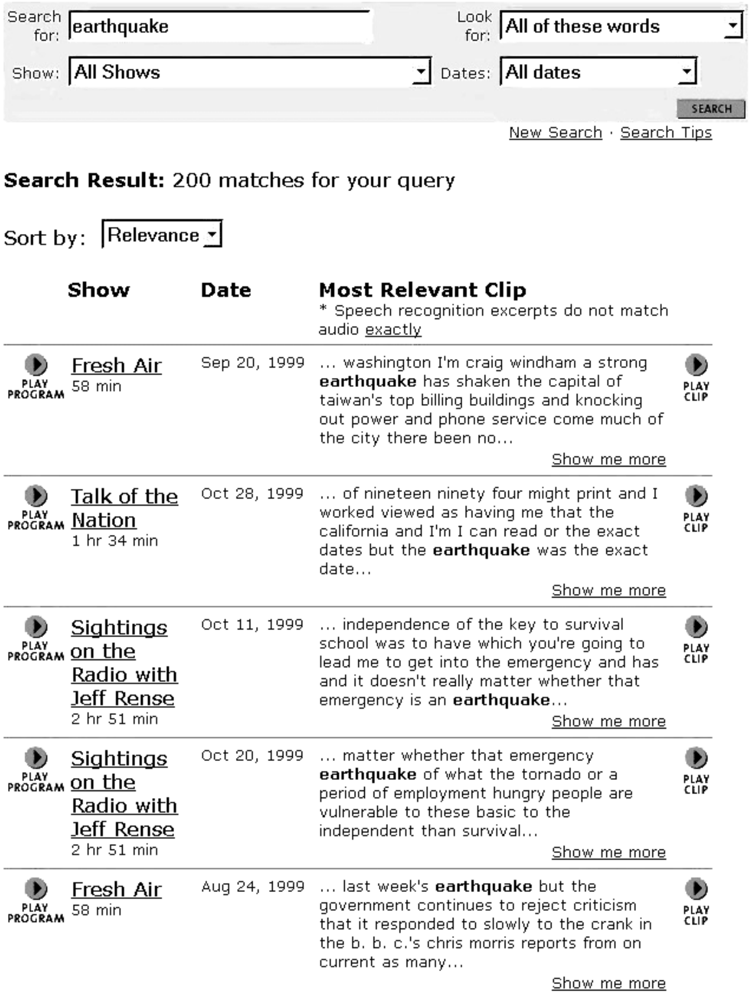
The Web server passes the user queries to the indexer, reads back the list of matches, retrieves the associated meta-data from the librarian, and formats and displays the results. As part of this process, it also performs the advanced functions described in Section IV, such as highlighting the matches within the transcript and expanding and normalizing acronyms, abbreviations and numbers.

The Web server returns up to 20 pages of up to 10 documents each sorted by relevance. External links to the original audio and video files are displayed, as well as a link for each document to a page with more details. This is shown in Figure 3 where we present the look and feel of a search page with the query *earthquake*.

The user has access to a within document search by clicking the *show me more* button under each hit. Figure 4 shows an example of such a page for the query *Bill Clinton*. These details include a navigable timeline of the 20 most relevant matches within the document, and also an option to display between 30 seconds and 2 minutes of the transcript text surrounding the match. Clips can be selected either directly from the timeline or from a list of clips sorted by relevance and displayed as a pull-down menu at the bottom of the page.

III. Performance Analysis

In this section we obtain objective measurements of the main components of the *Speech-Bot* system. We describe the performance of the transcoder, the speech recognition engine, the information retrieval engine (a combination of the speech recognition output and the indexer) and the Web server.



Search for: earthquake

Look for: All of these words

Show: All Shows

Dates: All dates

Search Result: 200 matches for your query

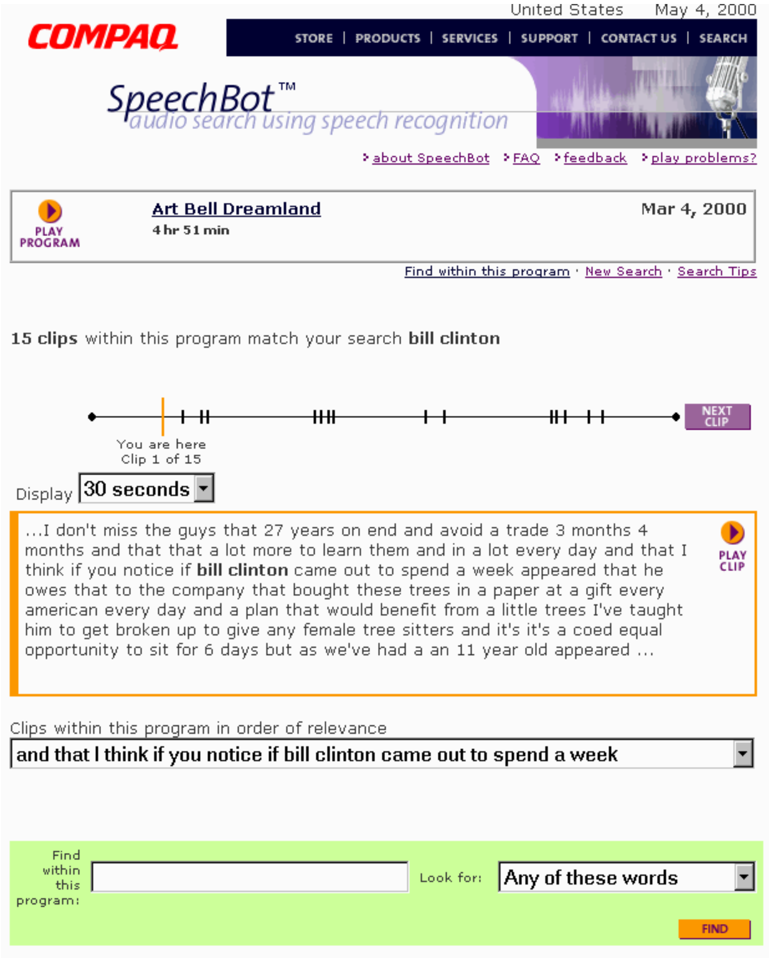
Sort by: Relevance

Show	Date	Most Relevant Clip * Speech recognition excerpts do not match audio exactly
Fresh Air 58 min	Sep 20, 1999	… washington I'm craig windham a strong **earthquake** has shaken the capital of taiwan's top billing buildings and knocking out power and phone service come much of the city there been no… Show me more
Talk of the Nation 1 hr 34 min	Oct 28, 1999	… of nineteen ninety four might print and I worked viewed as having me that the california and I'm I can read or the exact dates but the **earthquake** was the exact date… Show me more
Sightings on the Radio with Jeff Rense 2 hr 51 min	Oct 11, 1999	… independence of the key to survival school was to have which you're going to lead me to get into the emergency and has and it doesn't really matter whether that emergency is an **earthquake**… Show me more
Sightings on the Radio with Jeff Rense 2 hr 51 min	Oct 20, 1999	… matter whether that emergency **earthquake** of what the tornado or a period of employment hungry people are vulnerable to these basic to the independent than survival… Show me more
Fresh Air 58 min	Aug 24, 1999	… last week's **earthquake** but the government continues to reject criticism that it responded to slowly to the crank in the b. b. c.'s chris morris reports from on current as many… Show me more

Fig. 3. View of the search page

A. Transcoder

The current version of the transcoders can handle a variety of different formats both streaming and non streaming. Most of the documents downloaded however are RealAudio encoded. Transcoders run in parallel on a farm of 8 Compaq AP400 dual 400 MHz Pentium II workstations, 256 Mb RAM, under Windows NT. Each machine can handle 4 streams in real-time, leading to a maximum throughput of 768 downloadable hours of audio per day. Notice that the streaming formats cannot be downloaded at a faster than real time rate since by design streaming servers feed the data at real time rates, not faster nor slower.



15 clips within this program match your search **bill clinton**

Display: 30 seconds

...I don't miss the guys that 27 years on end and avoid a trade 3 months 4 months and that that a lot more to learn them and in a lot every day and that I think if you notice if **bill clinton** came out to spend a week appeared that he owes that to the company that bought these trees in a paper at a gift every american every day and a plan that would benefit from a little trees I've taught him to get broken up to give any female tree sitters and it's it's a coed equal opportunity to sit for 6 days but as we've had a an 11 year old appeared ...

Clips within this program in order of relevance: and that I think if you notice if bill clinton came out to spend a week

Find within this program:

Look for: Any of these words

Fig. 4. View of the *Show Me More* page

B. Speech Recognition

The speech recognition system has been tested on randomly selected segments from several popular radio shows: Coast to Coast with Art Bell, Dr Laura Schlessinger, Ed Tyll, Rick Emerson, and the Motley Fool Radio Show. Four 15-minute segments were selected from each of five shows for word error rate analysis. The majority of the audio streams are encoded with the 6.5 Kbps RealAudio codec. A few audio streams are encoded at higher rates. After download, the audio is stored in a wav file sampled at 8 kHz. Acoustic conditions vary as shows may have telephone conversations, commercials, several people talking simultaneously, or music in the background. The selected segments are transcribed manually, and the transcription is used to estimate the Word Error Rate

($WER = (S + I + D)/T$). Where S, I, D and T are the number of substitutions, insertions, deletions and total number of words respectively.

We break each 15-minute test segment into smaller utterances that are more easily handled by the recognizer. Each segment is broken into 30 seconds long subsegments overlapping by 5 seconds. We use a statistical 3-gram language model trained using the DARPA broadcast news Hub-4 1998 text corpora [7]. An n-gram statistical language model gives the probability of the current word given the previous n-1 words. Our training set has been trained on a corpus of about 100 millions words. It contains a vocabulary of 64,000 words which corresponds to 4 million bigrams and 15 million trigrams in the language model.

Since the audio has been encoded for streaming purposes, we have to adapt the acoustic models to the compression/decompression scheme. We investigated the impact of acoustic adaptation by exploring two acoustic modeling approaches. In our first approach we build acoustic models by training on 100 hours of the Broadcast News corpus provided by LDC (Linguistic Data Consortium) [13] at its original 16 kHz sampling rate with recording studio acoustic quality. When using these models the test data had to be up-sampled to 16 kHz. The second approach used models trained on the same training corpus but after being encoded using the 6.5 Kbps RealAudio codec, and then decoded to a sampling rate of 8 kHz. This encoding/decoding operation was performed to reduce the acoustic mismatch between the training corpora and the testing corpora.

On the test set we observed an improvement in the average word error rate (WER) from 60.5% for the first set of acoustic models to 49.6% when the 8 kHz RealAudio encoded/decoded models were used. These results are for a system with 16 Gaussian mixture components per HMM state and 6,000 shared states. Table I presents full results for the 8 kHz RealAudio system.

As a comparison, the Total Error Rate (TER) of television news and sports captioners is usually between 1.5% and 2.5% [14]. The TER is however measured slightly differently to the WER. The TER calculates the proportion of errors over the total number of words. Errors includes mis-typed words and punctuation errors which usually do not occur in transcriptions generated by a speech recognizer.

TABLE I

Speech Recognition WER. 8 kHz RealAudio models, 16 Gaussian mixture components per clustered HMM state. 6000 cluster states

Show Name	**Chunk 1**	**Chunk 2**	**Chunk 3**	**Chunk 4**
Art Bell	50.5%	46.3%	51.2%	44.5%
Dr. Laura	51.2%	47.4%	52.7%	59.2%
Ed. Tyll	62.3%	54.0%	49.9%	47.0%
R. Emerson	48.2%	51.5%	53.1%	56.4%
M. Fool	44.1%	38.8%	47.2%	43.5%

The Calista speech recognizer can be adjusted to trade off processing time for accuracy within certain limits. We observe that ranges between 6 and 30 times longer than real time on a Pentium II, 450 MHz processor under Linux give good accuracy/processing time tradeoffs for a production system. The dominant parameter is the beam of the recognition search. During recognition, hypotheses with likelihood less than the beam are discarded, reducing the number of calculations required. Table II shows the performance of the recognition system as this factor is adjusted. In production mode we use 8 Gaussian mixture components per state and a beam of 10^{-58}. The recognizer then runs at around 13 times real time (xRT). Because the speech recognizer is parallelized when running on archived data, and assuming an average of 13 times real-time, we are able to process approximately 110 hours of audio per day on a farm with 60 processors.

C. Information Retrieval Performance

The effectiveness of retrieval depends on a number of factors such as the complexity of the queries, the length of the documents, the size of the collection, the accuracy of the automatic transcriptions, and the information retrieval techniques used.

The retrieval performance was evaluated on the public site with static content since the index needs to be stable during the tests which last several days. We had independent and unbiased testers evaluate a set of queries selected from the list of 100 most frequently

TABLE II

WER AND SPEED (X REAL TIME) OF SPEECH RECOGNIZER AS A FUNCTION OF BEAM AND NUMBER OF GAUSSIANS. 8 KHZ REALAUDIO TRAINED MODELS

Number of Gaussians	**Beam**	**WER**	**xRT**
16	10^{-64}	49.6%	23.2
16	10^{-58}	51.5%	18.5
8	10^{-64}	51.9%	18.0
8	10^{-60}	53.0%	14.5
8	10^{-58}	53.9%	13.0

submitted queries. Each tester performed 50 queries, and assessed the precision of the top shows returned for each. The study was limited to the top 20, 10 and 5 shows, based on the assumption that typical users tend only to look at the first couple of pages of the retrieved results [15]. The query terms were selected from the top 100 queries submitted to the public site from December 1st, 1999 to March 1st, 2000. The words were selected such that they cover a large variety of topics, varying length of words (phoneme-wise), and varying types of words such as acronyms and proper nouns. The queries were either nouns or noun phrases with a maximum length of 3 words. Example queries are *Bill Clinton*,*Internet* and *Y2K*.

Testers judged the relevance of the retrieved documents based on a *concept hit* approach, *i.e.* a document was considered relevant if and only if the concept described in the query was present in the retrieved document. To assess whether a given result was relevant, testers read the transcripts returned with the search results of the detailed page and, for each clip of the document, listened to the corresponding parts of the audio show. To evaluate the retrieval results, we used a standard average precision metric [16]. We did not run the experiment on a standard test set, like TREC-SDR [3], but rather on real data (both audio content, and queries) where the diversity of audio condition and content is very high. Unlike TREC evaluations, we do not know the ground truth of our index, so we cannot measure recall. We therefore measure precision which still meaningfully reflects how well our index serves our users' information needs.

Table III presents average retrieval precision numbers for the above described experiments for two different ranking functions.

TABLE III

AVERAGE RETRIEVAL PRECISION FOR WHOLE DOCUMENTS USING TWO DIFFERENT RANKING FUNCTIONS

Ranking function	**Docs (count)**	**Duration (hours)**	**Queries (count)**	**Average precision**
A	4695	4188	50	71.7%
B	4695	4188	50	75.5%

Ranking function A scores documents using a slightly-modified implementation of an algorithm for ranking documents in large text index. Function A is calculated using an approximate term frequency inverse document frequency (tf.idf) metric [11], combined with a score based on the proximity of query terms to each other and a score based on their location within the document. For speed of search, the term frequency in the approximate tf.idf is calculated as a step function indicating whether the term occurs once or more than once in the document. The proximity bias helps to retrieve documents with a multi-word query string, even if the user didn't choose the *this exact phrase* option. Location within the document is calculated such that terms closer to the beginning of a document receive a higher score.

Ranking function B scores documents using a true tf.idf metric, combined with a score based on the proximity of query terms to each other. The bias towards terms occurring earlier in the document is not included in function B.

The retrieval performance of the system is better than expected considering the accuracy of the speech decoder. In many studies, document retrieval has been shown to be remarkably resilient to the speech recognizer word error rate. Recent experiments show that a word error rate of 30% reduces recall by 4% relative, and a word error rate of 50% reduces it by only 10% relative [17], [4]. There are several explanations for these numbers. First the query words are often repeated several times during a show and are thus more likely to be recognized. Second, the keywords tend to be longer than stop words (e.g. *a, the, and ...*), so the speech recognition search is more constrained and tends to perform better on these words. Also, if there are many words in the query, missing one or two of them may still permit retrieval of the document.

Retrieval errors were due to several reasons. First, insertion or substitution recognition errors cause query words to appear erroneously in the transcripts. Ranking function A is more sensitive to this type of error: we observed several cases where an insertion at the very beginning caused the document to get an erroneously high score. This type of false alarm error represents roughly half of the cases of the appearance of non-relevant documents. Ranking function B helps to alleviate this problem by weighting terms equally throughout the document without using a locality bias. This makes intuitive sense when you consider that an audio document is less likely than a text document to be about a single topic.

All of the words used for evaluation were in the vocabulary of the speech recognition engine, thus the average precision measured should be viewed as an upper bound performance. However, our Out Of Vocabulary (OOV) rate is very low (about 1.2% with a static language model built in late 1999) and has a minor impact on our recognition performance. It is likely that other factors such as acoustic and language model mismatch, and pronunciation inaccuracies are the main contributors to the word recognition error rate. We have found that the OOV of user queries (12%) is considerably higher than the speech recognition OOV. This rate has clearly an implication on the performance of the system since a query with a word that is out of vocabulary do not return any valid hit [18].

The second main reason for a non-relevant document to be retrieved is that the query words are mentioned out-of-context, or are inherently ambiguous. For example, the query *aids* returned many documents which talked about *aids* meaning *helps* rather than a disease. These problems may be overcome by introducing a notion of subject popularity, and introducing bias in the ranking scheme [19].

Finally, documents may not be retrieved because of the ranking function itself. Although these errors probably contribute to the performance of the system, we were not able to evaluate their impact because we don't have a properly labeled test set to measure the recall.

D. Web Site Performance

To estimate overall performance in terms of speed as perceived by a user, we used automated test scripts which fire thousands of queries at the site over a certain period of time, and read back the results. The scripts measure page views per second and average latency. The queries used are real user input gathered from the site logs since *SpeechBot* went live in December 1999. They are submitted in multiple parallel sequences with random variable delays to simulate likely real-world conditions.

The results were found to be affected by the frequency of the query terms in the database, by database and index size, and by the number of CPUs available for use by the system. It is possible to split components of the *SpeechBot* system over several single or multi-processor machines in order to increase performance with a very large database. Performance profiling using DCPI [20] and our own logging tools showed that the main bottlenecks were usability features including acronym expansion and ranking of matches within documents. Performance improvements in these areas have resulted in an increase from 5.83 page views per second to 12.15 page views per second. The tests used $3,063$ hours of content with all components running on a single Compaq DS20 AlphaServer EV6 (2 CPUs) and simulated 275 users waiting an average of 15 seconds between clicks. A site with this performance would support over 1 million hits per day.

The fastest results to date, 27 page views per second with an average latency of 0.76 seconds, were obtained with no acronym expansion or within-document ranking, and a relatively small database (200 hours of content) with the librarian and the indexer running on a Compaq AlphaServer 4100 with 4 processors and 2Gbytes of memory and the UI running on a separate Compaq AlphaServer 4100 with 4 processors and 2Gbytes of memory.

We have also conducted performance tests of the database as a standalone component since it is a potential bottleneck for the system when serving the user interface. The test of the database component alone running on one AlphaServer 4100 with four Alpha processors at 466MHz measured a throughput of 29 user sessions per second, equivalent to 88 pages per second or 7.5 million pages over a 24-hour period. These performances are better than the presentation server, which generates the HTML pages at a maximum rate of about 30 pages per second [12].

IV. Usability Studies

The usability of the site was extensively evaluated and improved. A group of seven experienced Internet users aged 18-50, most (five) with experience watching video content or listening to audio content on the Web were each observed while they attempted to achieve the same set of goals using a *SpeechBot* prototype site. The goals were expressed in the form of two scenarios written in terms of what information the user was to discover using the site, with each scenario covering the two universal goals of searching: to make the widest possible search and to narrow down the results of a search. Before acting out the scenarios, each user was asked to perform a free-form search for a subject of their own choice. Users were given no instruction on the use of the site, and each session took a little over an hour. Users were encouraged to talk through their thought processes and expectations prior to each interaction with the site and then comment on the response obtained from the site. Each user also completed a brief questionnaire at the end of the session. The trends in behavior were analyzed to determine the major factors to be improved.

Although the interface is very similar to most text-based search engines users encountered several difficulties. When performing multiword searches, in general users expected to see all query words in the results, which is not necessarily the case depending on how the search has been performed. Interestingly, even regular *AltaVista* users did not understand the query syntax. For instance, when looking for *divorce in Dr. Laura's interviews*, most first tried *laura divorce*, clever ones then tried *laura+divorce*, but nobody tried *+laura +divorce*. This problem was addressed by exposing the possible relationships between search words *(all/any/phrase/Boolean)* in a drop down menu and making *all* the default. Second, the syntax for query terms is sometimes non-intuitive. For example, users may type acronyms in different forms: *USA* or *U.S.A.*, but without special processing not all written forms will match their spoken counterpart. To alleviate this problem, terms like abbreviations, acronyms, and numbers are now expanded and normalized by the system to a common form for both querying and indexing. For instance, the number 250 will be trained and recognized as *two hundred and fifty*, probably be searched as 250 and will be indexed in its normalized form as 250. Finally, the advanced search was confusing and provided no additional functionality other than Boolean search. This was therefore dropped after adding the Boolean capability as a choice on the simple search page.

Some improvements to the display of results were indicated by user testing. Highlighting the query words was essential, and gave the user strong feedback on the relevance of the document even if the speech recognition output was sometimes hard to read and understand. Users wanted guidance on what topics could be searched. To address this, we added a list of indexed shows grouped by topic to the home page.

The accuracy of the transcription appeared to be a problem too. For example, on some content such as noisy or poorly articulated speech segments, the output of the speech recognition can be very poor. A solution to this problem is the subject of ongoing research. Lastly, users were often frustrated by the reliability of the RealAudio proxies and servers. Our query servers are stable and responsive, but the audio has to been streamed in from a variety of content-hosting sites, some of which are less reliable.

A smaller in-house study on the public site was later conducted on a group of four university students to verify that the changes made prior to launch had resolved the original usability problems. This study showed that most of the problems had been fixed, but that there was a residual problem when searching with multiple keywords input. Depending on how closely together the words were spoken in the original audio document, users still occasionally found that the results didn't contain all the search words they expected. To address this, we are currently investigating a modification to make the default search option behave more like *all these words nearby.*

V. Conclusions and future work

SpeechBot is the first audio indexing and retrieval system for the Web. It incorporates speech recognition technology and so is able to operate both with and without transcriptions. The current version of the system is capable of indexing up to 800 hours of audio per week, and is configured in such a way that it can easily scale up on demand. It has given us a unique opportunity to test a spoken data retrieval system on a large scale. Our experiments show acceptable retrieval accuracy despite high recognition error rates. This suggests that indexing of audio documents on the Web is feasible given the current level of recognition and retrieval technologies.

There is still room for many improvements. The OOV rate for user queries could be improved by performing better text normalization. We are also planning to investigate indexing word categories and/or perform subword based retrieval. Speech recognition accuracy can be increased by providing specialized audio models, vocabulary, pronunciation dictionaries, and language models. Acoustic models could be adapted to specific shows. Language models could be trained for specific domains, like sport or financial news. Finally, we plan to update the recognition vocabulary, and improve word pronunciation for proper nouns.

To improve IR accuracy, we are investigating different relevance ranking algorithms and alternative ways of indexing the output of the speech recognizer, as well as query and document expansion methods. We also plan to improve the user interface since random access to multimedia documents raises interesting new questions about the efficiency of the current design. A better segmentation of shows could allow search and retrieval of small sections of audio recordings that could be listened to as a whole, thus minimizing the navigation within a document. Finally, we plan to port the speech recognition engine to other languages, such as French and Spanish.

VI. Acknowledgments

The *SpeechBot* project has benefited from the insights and work of many people. We would like to thank first the Compaq Software Engineering Australia group, Greg McCane, Liam Davies, Ian Walters, Shazina Hashmi, Sue Salamacha, and Robyn Hancock, without whom the site would not exist. We also thank Frank Bomba and his group including Scott Blackwell, Gene Preble, and Ben Jones. They provided talented engineering and testing support. Scott has been in charge of running the site, keeping the index fresh, and performing numerous analyses on our Web site. Bhiksha Raj and Ed Whittaker helped us by giving their insight on language modeling. We also would like to acknowledge the past contributions of Ron Gentile, Dave Goddeau, R. Paul Johnson, Andrew Leonard, Anna Litvinova, Mike Swain, and Chris Weikart. Finally, we would like to thank the Hiser

Group and APL Digital, both in Australia, for their help in the usability studies and Web site graphic design respectively.

References

[1] H. D. Wactlar, A. G. Hauptmann, and M. J. Witbrock, "Informedia: News-on-demand experiments in speech recognition," in *DARPA Speech Recognition Workshop*, 1996.

[2] S. E. Johnson, P. Jourlin, G. L. Moore, K. Sparck Jones, and P. C. Woodland, "The cambridge university spoken document retrieval system," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[3] D. Abberley, G. Cook, S. Renals, and T. Robinson, "Retrieval of broadcast news documents with the thisl system," in *Proceedings of the 8th Text Retrieval Conference (TREC-8)*, 1999.

[4] J. Garfolo, E. Vorhees, C. Auzanne, V. Stanford, and B. Lund, "Spoken document retrieval track overview and results," in *Proceedings of the 7th Text Retrieval Conference (TREC-7)*, 1998.

[5] P.C. Woodland, T. Hain, S.E. Johnson, T.R. Niesler, A. Tuerk, and S.J. Young, "Experiments in broadcast news transcription," in *Proc. ICASSP 98, Vol II, pp. 909-912, Seattle, Washington*, 1998.

[6] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and E. Thayer, "The 1996 hub-4 sphinx-3 system," in *Proceedings of the 1997 DARPA Speech Recognition Workshop, Chantilly, Virginia*, 1997.

[7] D. S. Pallet, J. G. Fiscus, J. S. Garafolo, A. Martin, and M.A. Przybocki, "1998 broadcast news benchmark test results," in *DARPA Speech Recognition Workshop*, 1999.

[8] P. J. Moreno, C. Joerg, J. M. Van Thong, and O. Glickman, "A recursive algorithm for the forced alignment of very long audio segments," in *International Conference on Spoken Language Processing*, 1998.

[9] P. Placeway and J. Lafferty, "Cheating with imperfect transcripts," in *International Conference on Spoken Language Processing (ICSLP)*, 1996.

[10] M. Burrows, "Method for indexing information of a database. u.s. patent 5,745,899," 1998.

[11] G. Salton and M. J. McGill, ," in *Introduction to Modern Information Retrieval. McGraw-Hill*, 1983.

[12] B. Eberman, B. Fidler, R. A. Iannucci, C. Joerg, L. Kontothanassis, D. E. Kovalcin, P. J. Moreno, M. J. Swain, and J. M. Van Thong, "Indexing multimedia for the internet," in *In Visual Information and Information Systems. D. P. Huijsmans and Arnold W.M. Smeulders (Eds.) Springer-Verlag*, 1999.

[13] "Linguistic data consortium (ldc)," http://www.ldc.upenn.edu.

[14] G. Robson, *Inside Captioning*, CyberDawg Publishing, Castro Valley, California, 1997.

[15] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz, "Analysis of a very large altavista query log," in *SRC Technical Note 1998-014*, 1998.

[16] G. Salton, *Automatic Text Processing: The transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, Reading, Massachusetts, 1989.

[17] M. Witbrock, "Search engines and beyond," in *SearchEngine Meeting, Boston*, 1999.

[18] B. Logan, P. Moreno, JM Van Thong, and E. Whittaker, "An experimental study of an audio indexing system for the web," in *International Conference on Spoken Language Processing (ICSLP)*, 2000.

[19] A. Singhal and F. Pereira, "Document expansion for speech retrieval," in *ACM SIGIR'99*, 1999.

[20] J. Dean, C. A. Waldspurger, and W. E. Weihl, "Transparent, low-overhead profiling on modern processors," in *Workshop on Profile and Feedback-Directed Compilation, Paris, France*, 1998.