Video Mail Retrieval by Voice: An Overview of the Cambridge/Olivetti Retrieval System.

M. G. Brown[1] J. T. Foote[2] G. J. F. Jones[2,3] K. Sparck Jones[3] S. J. Young[2]

[1]Olivetti Research Limited, 24a Trumpington St., Cambridge, CB2 1QA, UK
[2]Cambridge University Engineering Department, Cambridge, CB2 1PZ, UK
[3]Cambridge University Computer Laboratory, Cambridge, CB2 3QG, UK

Abstract

This paper describes current work on a video and audio document retrieval project at Cambridge University and Olivetti Research Limited (ORL). The project seeks to integrate state-of-the-art text retrieval methods with high-performance word spotting to yield a robust and efficient video mail retrieval system. A specific goal is the development of a practical retrieval system to work with Medusa, a high-bandwidth multimedia environment in daily use. This paper describes the project background, message corpus and experiment design, and presents experimental results showing audio retrieval performance very close to that of text.

1 Introduction

The last few years has seen an increasing use of multimedia applications, including video conferencing and video and audio mail [1]. Using these facilities can create large archives of video material, which poses a significant storage and retrieval problem. Users are unable to find stored messages because, unlike text, there are no simple ways to search for a particular reference. The Video Mail Retrieval (VMR) project is addressing this problem by developing a system to retrieve stored video messages by speech recognition. A specific goal of the project is the development of a useful application to work in the Medusa multimedia environment installed at Olivetti Research Ltd in Cambridge, UK.

In the simplest form of message retrieval, a user will specify a single search keyword and word spotting techniques will locate its occurrences in the audio soundtrack. A more robust system will use multiple search keys, both to minimize the effect of spotting errors and to refine the list of retrieved messages. Thus, the topic specification and search strategies developed for conventional text-based information retrieval (IR) must be adapted to this new environment [2]. Although later stages of the project will investigate open-keyword and open-user sets, the initial stage, and the work described here, focuses on a fixed, *a-priori* known set of search keywords and users. This paper begins with an overview of the ORL Medusa system; subsequent sections describe initial word spotting and information retrieval experiments, and a graphical user interface to the video mail retrieval system.

2 Medusa: Multimedia via Networked ATM

The Medusa Project at Olivetti Research Ltd (ORL) is an experiment in sending many streams of digital audio and video over a prototype 100 Megabit-per-second switched ATM network [3, 4]. The ATM fabric covers all rooms at the laboratory and some 200 network connections are in place. Figure 1 shows a typical Medusa installation. Optical fibre extends the network to the University Engineering Department and the University Computer Laboratory in Cambridge. An ATM network has useful characteristics for multimedia transport: the bandwidth is relatively high, the latency is low, and transit time jitter is also low.

Multimedia input and output functionality has traditionally been provided by cards plugged into a workstation bus. In contrast, ORL has adopted the more flexible approach of making microphones, speakers, cameras and storage systems into first-class network objects. A modular approach has been adopted to-

Figure 1: The Medusa multimedia system.

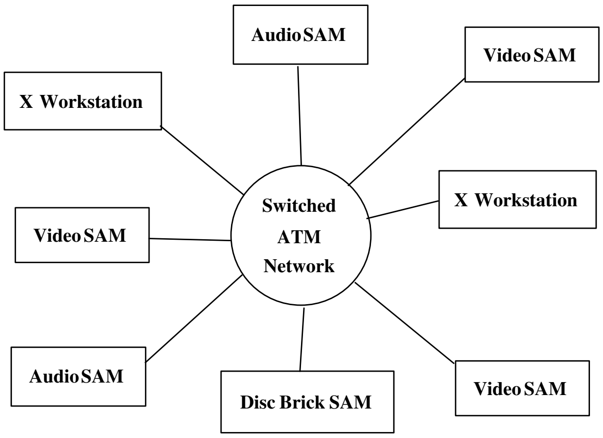

Figure 2: Connection of Medusa Direct Peripherals to an ATM network.

wards the design of the Medusa hardware and various ATM direct peripherals have been made by adding specialised option cards to an ATMos card. The ATMos card has an ATM network interface, an ARM processor and up to 32 Mbytes of memory. A range of Smart ATM Modules (SAMs) have been developed and deployed throughout the laboratory. They are lunchbox-sized desktop units which plug directly into the ATM network. Figure 2 shows connection of various Medusa ATM Direct Peripherals to the network. The 8-port ATM switches on which the network is based are also constructed from ATMos cards.

An Audio SAM supports a set of ATM microphones and speakers. It has two high-quality programmable CODECs and provides four input audio channels and four output audio channels sampling at up to 48 kHz with 16 bits of precision. The Video SAM is an ATM networked video unit to which one or two colour camera heads can be connected. The Disc Brick SAM uses the ATMos card to connect a RAID 3 array of discs to the network. An in-house micro-kernel called ATMos runs on the ARM processor of each SAM. It is capable of the rapid task-switching required to service low-latency reception and transmission of ATM cells. ORL has also developed a TURBOchannel ATM interface card so that multimedia data can be processed on workstations and video displayed at 25 frames per second in X windows.

The Medusa software applications environment is a peer to peer architecture for controlling networked multimedia devices [5]. A Medusa server process runs on each of the SAMs as well as on the ATM networked workstations. Multimedia data flows through a series of Medusa software modules on these servers. Medusa module code makes it straightforward to set up and dynamically modify media connections between software modules connected in a pipeline. There are three basic types of Medusa software module: the source module, which is specialised for audio or video output; the sink module, which has an input connection and is specialised to sink audio or video; and pipeline modules which have both inputs and outputs. Pipeline modules can buffer data or perform data transformation in real time — audio format con-

Figure 3: Example of Medusa software "module" connection.

version is done with such a pipeline module. Another type of sink module is an analyser. There are audio analyser modules to report audio loudness and perform speech recognition.

Medusa module code allows location-transparent connections to be made between these sources, sinks and modifiers of multimedia data. The choice of which ATM networked hardware platform each software module runs on is left to the application programmer and is usually made on the grounds of efficiency. For example, audio could flow between an audio source module on a Disc Brick SAM and an audio sink module on an audio SAM via a format convertor running on either machine (Figure 3), or perhaps on a powerful processor available elsewhere on the network.

Medusa application programs (including the VMR user interface described in Section 3.1) are written in Tcl/Tk [6], a scripting language with useful built-in X widgets. Tcl has been extended at ORL to include commands for initialising Medusa modules; making connections between them; and reading and writing to the controlling attributes on the modules. In the case of an audio source module the attributes include the sampling rate, the number of sample bits, and the quantisation method.

Several large Tcl applications have been written to demonstrate the multi-stream capabilities of Medusa. Each Medusa station is equipped with four camera heads, four microphones and four speakers. The Medusa phone application shows small images from the four cameras at each end of the conversation, as well as a large images from each end which can be either selected manually or automatically. Agent software uses the output from analyser modules, which are attached to the audio and video sources, to make useful video and audio stream selections automatically. The Medusa mail application (Figure 4) makes a recording of all four streams of video available from a Medusa system as well as the most appropriate microphone stream. These five media streams are passed through timestamping pipeline modules and recorded onto ATM networked Disc Bricks. During playback, all four recorded views are seen, and one of them can be selected to be shown in large format, as shown in Figure 4. At present, there exists sufficient Disc Brick

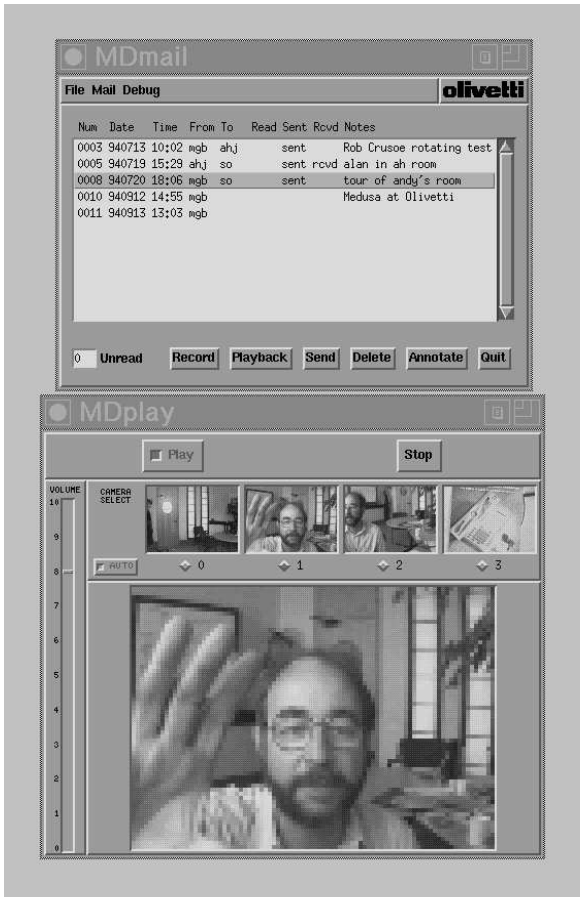

Figure 4: Medusa video mail application.

space to record 1000 typical twenty-second video mail messages.

At present, the video streams are uncompressed; the planned use of JPEG encoding will substantially increase the amount of video that it is practical to store. The potential size of this archive dramatically underlines the need for more sophisticated retrieval methods, such as those under investigation here.

3 The Video Mail Retrieval System

A prototype Video Mail Retrieval application has been developed that integrates keyword spotting, information retrieval, and video capture/playback capabilities. Figure 5 shows a block diagram of the VMR system, which operates as follows. The audio soundtrack of each message (whether from an existing archive or received as new mail) is passed to the acoustic word spotter. This computes a sequence of putative keyword hits, which is added to an index contain-

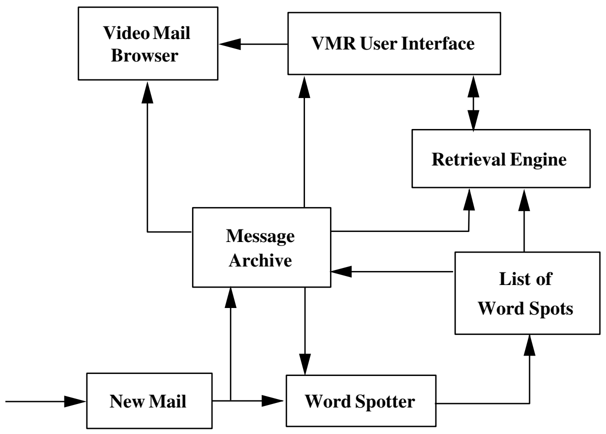

Figure 5: Block diagram of VMR system.

ing all putative hits for all messages, along with mail header information and pointers to the message data (for playback). Because the computationally-intensive word spotting phase is done off-line (as messages are added to the archive), retrieval of archived messages is nearly instantaneous.

3.1 The Video Mail Retrieval User Interface

The approach taken for the VMR user interface is the "message list filter." Upon startup, a scrollable list shows all available messages in the user's video mail archive. Various controls let the user "narrow" the list, for example, by displaying only those messages from a particular user or received after a particular time. Unsetting a constraint restores the messages hidden by that constraint; multiple constraints can be active at one time, giving the messages selected by a boolean conjunction of the constraints. Basic constrainable attributes are user name, date, and other information found in the message header. Messages are ranked by origination date, such that the most recently received document is displayed at the top of the list.

3.2 Searching for keywords

The natural extension to this scheme is to add message attributes that depend on the keywords present in the message. Messages can then be ranked by the presence/absence of a keyword or combination of keywords. In operation, the user selects one or more keywords as a search query. The resulting score for each message is computed by the retrieval engine, using the scoring methods described in section 6. The interface then displays a list of messages ranked by score,

Figure 6: Video Mail User Interface application.

with the message scores shown as bar graphs (Figure 6); messages with identical scores are ranked by date. In its simplest form, the keyword search resembles an "audio grep" that returns a list of messages containing a particular keyword.

3.3 Video Message Browser

Once the list of messages has been sufficiently narrowed, a "video browser" can be activated. This graphically represents each message as a dark horizontal bar, with keyword hits displayed as lighter regions. Time runs from left to right, and keyword hits are displayed proportionally to when they occur in a message; for example, keywords at the beginning appear on the left side of the bar. The brightness of a keyword region is proportional to the spotting score, so that more likely hits appear brighter and stand out (search keywords are shown in a red-black colour continuum while the other keywords are shown in white-black). Portions of the video message can be selected for playback by dragging over part of the bar; this lets the user selectively audition regions of interest rather than the entire message.

4 The Message Corpus

In a real-world system, the message archive of Figure 5 would typically contain video mail messages in-

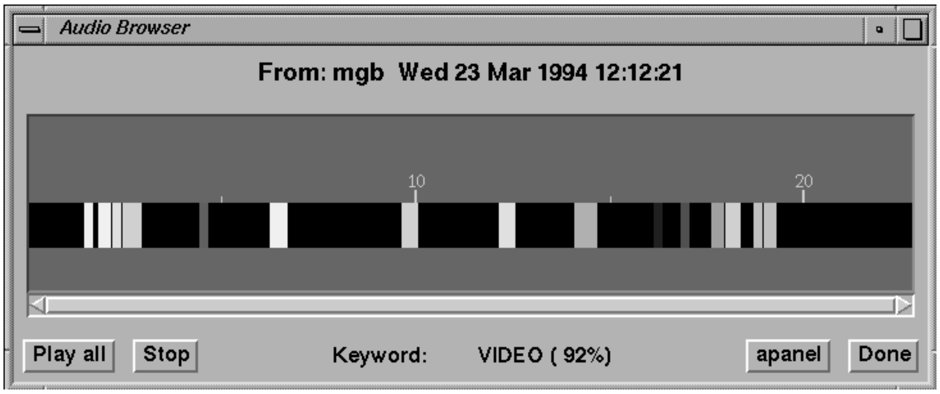

Figure 7: Video Mail Browser application.

tended for a user or group. For the initial development of the VMR system, it was necessary to create a test archive of messages with known audio and information characteristics to evaluate word spotting and message retrieval performance. Unfortunately, existing corpora intended for word spotting research were not appropriate for this purpose for two reasons. The VMR system is intended to work in a multimedia system with high fidelity audio, while most existing corpora are only telephone-quality. In addition, the information content of existing corpora is not always appropriate for information retrieval experiments. A new corpus of audio mail messages, constituting a more natural document set, was recorded at CUED.

The VMR message corpus is a structured collection of audio training data and information-bearing audio messages. Ten "categories" were chosen to reflect the anticipated messages of actual users, including, for example, "management" and "equipment." A fixed set of 35 keywords were chosen to cover the ten categories; thus the "management" category includes the keywords "staff," "time," and "meeting." Keywords may be associated with more than one category; and the keyword set includes 11 difficult monosyllabic words (e.g "date" and "mail"), as well as overlapping words (e.g. "word" and "keyword"), and word variants (e.g. "locate" and "location"). For the message data, talkers were asked to record a spontaneous response to a prompt, (with five prompts per category), for a total of 50 unique prompts.

There were fifteen talkers, of which 11 were male and 4 female. Data was sampled at 16 kHz from both a Sennheiser HMD 414 close-talking microphone and the cardioid far-field desk microphone used in the Medusa system, in an acoustically isolated room. Each talker provided about 45 minutes of speech for a total of nearly ten hours of recorded data. Half of this data was read speech necessary for acoustic model training; the remaining 5 hours of speech was 300 spontaneous messages used to evaluate both word spotting and information retrieval performance. There were 6 spontaneous messages (from 6 different users) in response to each of the 50 prompts.

All files were verified and transcribed at the word level; non-speech events and disfluencies such as partially spoken words, pauses, and hesitations were transcribed as well. These transcriptions are especially important so that the performance of the speech-based retrieval system can be compared with that of text. Phonetic transcriptions were automatically generated from a machine-readable version of the Oxford Learners Dictionary. The standard reduced TIMIT[1] phone set was augmented with additional vowels specific to British English pronunciation. A full description of the VMR corpus may be found in [8].

5 Keyword Spotting

Automatically detecting words or phrases in unconstrained speech is termed "keyword spotting;" this technology is the basis of the VMR system [9]. The best-performing keyword spotters are based on the same hidden Markov model (HMM) methods used in successful continuous-speech recognition [10]. A hidden Markov model is a statistical representation of a speech event, typically a word or phoneme; there exist fast algorithms for both training HMM parameters and finding the most likely model sequence given unknown speech. The HTK tool set developed at Cambridge University ([11]) is a powerful and flexible set of

[1] TIMIT is a widely-used corpus of spoken and transcribed English sentences [7].

software tools for developing HMM applications such as the keyword spotting system presented here.

5.1 Word Modelling

For hidden Markov model (HMM) training and recognition, the acoustic speech data is parameterized into a spectral representation. Whole-word talker-dependent keyword models and monophone filler models were constructed for each of the 15 talkers, using the HTK tool set. Keyword models were trained on approximately 10 instances of each keyword, of which half were spoken in isolation and half in the context of read messages. Filler models, to represent non-keyword speech, were trained on sentences from the TIMIT corpus and the non-keyword parts of the read messages. For initialization, the phone sequences were aligned to the training data using TIMIT monophones, which proved to be satisfactory even for British English talkers.

Separate models were trained for both the head-microphone and desk-microphone data. When comparing the speech signal and background (environmental) noise, the head-microphone data had an average SNR of 35-45 dB. Though the SNR of the desk-microphone data was substantially less (about 20-25 dB), the recordings are subjectively crisp, with little of the audible reverberation characteristic of omnidirectional microphones.

5.2 Keyword Recognition

Keyword spotting is done with a two-pass recognition procedure. First, Viterbi decoding is performed on a network of just the filler models, yielding a time-aligned sequence of the maximum-likelihood filler monophones and their associated log-likelihood scores. Secondly, another Viterbi decoding is done using a network of the keywords, silence, and filler models in parallel. In a manner similar to Rose & Paul [12], keywords are rescored by normalizing each hypothesis score by the the average filler model score over the keyword interval. Unlike [12], however, the average log likelihoods are divided rather than subtracted, which results in better performance [13]. Because of the limited training data, it was necessary to tune the filler models so that they did not match an undue number of keywords. A satisfactory solution was to introduce filler models of common 3-phone sequences by concatenating three monophone models, and adjusting the word transition penalty to penalize the filler sequences (which must be traversed in groups of three).

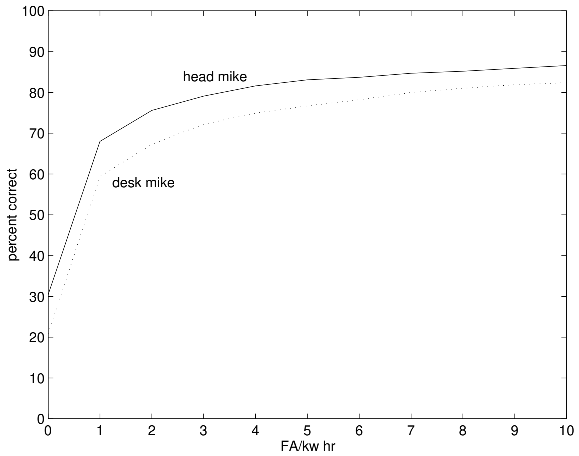

Figure 9: VMR Task Keyword Spotting ROC.

Data set:	head	desk
Talker Avg.	81.2%	76.4%

Table 1: Average Figures of Merit.

Figure 8 shows the network topology for the keyword recognition Viterbi pass.

5.3 Recognition Results

An accepted figure-of-merit (FOM) for word spotting is defined as the average percentage of correctly detected keywords as the threshold is varied from one to ten false alarms per keyword per hour. Keywords spotting results were scored against aligned text transcriptions containing the keywords. The FOM for the two sets are shown in Table 1, averaged across both the 15 talkers and the 35 keywords. The receiver operating characteristic (ROC) curve for both the close-talking and far-field microphone data is shown in Figure 9. Keyword spotting results for both the head and desk microphone data were used for the retrieval experiments of Section 6.

6 Information Retrieval

6.1 Methodology

Information retrieval experiments require message *queries* and assessments of message *relevance* to queries. Since real user queries and assessments were not available for the message corpus, they were simulated for our first tests as follows. Queries were constructed from the message prompts. To reduce vari-

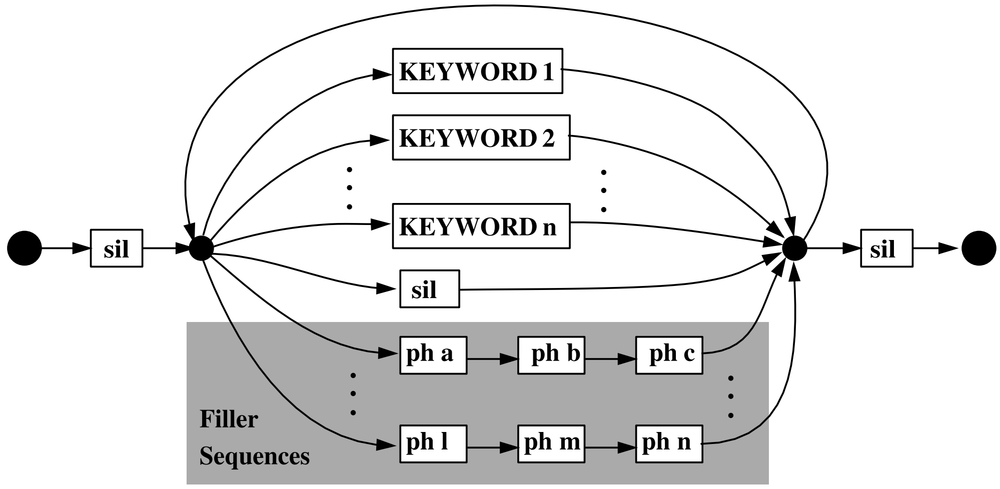

Figure 8: Keyword recognition network.

ations in word form that inhibit retrieval matching, query words were suffix-stripped to stems using a standard algorithm [14]. Queries were formed from the prompts by selecting those stems also found in the keyword stem list. For example, given the prompt

```
Your current project is lagging behind
schedule.  Send a message pointing this
out to the other project management staff.
Suggest some days and times over the next
week when you would be willing to hold a
meeting to discuss the situation.
```

the following query was obtained:

```
project messag project manag staff time meet
```

The 6 recorded messages generated in response to each prompt were assumed relevant to the query constructed from that prompt. Note that the 24 other messages in the same category, which are quite likely to contain similar keywords, are assumed not relevant; retrieval of one of these messages is construed as an error. Thus the retrieval task is comparatively difficult.

In retrieval, a score is computed for each query-document pair; the search output may then be ranked by matching score [15]. With keyword presence/absence only, the score is the number of keywords in common, often called the coordination level (cl) score. Keywords may also be usefully weighted, for instance by the *inverse document frequency* (idf) weight,

$$idf_i = \log \frac{N}{n[i]}$$

where N is the total number of documents and $n[i]$ is the number of documents that contain keyword i. Thus, keywords occurring in a small number of documents are favoured. For this weighting scheme the query-document score is the sum of keyword weights for each keyword occurring in both the query and the document.

Retrieval performance is often measured by *precision*, the proportion of retrieved messages which are relevant to a particular query. One conventional single-number performance figure, average precision, is derived as follows: the precision values obtained with each new relevant document in the ranked output for an individual query are averaged, and the results are then averaged across the query set. A single-number performance measure like this is useful for comparing text and word spotting results.

6.2 Calibration via Text Retrieval

Acoustic word spotting is prone to false alarms and missed keywords, so retrieval can be expected to suffer. The extent of the degradation can be measured by comparing the retrieval performance on word spotting results to the retrieval performance on text. We used our transcribed corpus to provide us with this performance standard, applying suffix-stripping, matching and scoring as described in the previous section.

An additional problem of word spotting is that unrelated acoustic events will often resemble valid keywords. For example, the last part of “hello Kate” is acoustically quite similar to the keyword “locate.” Because even the best acoustic models cannot discriminate between homophones, the output of an ideal word

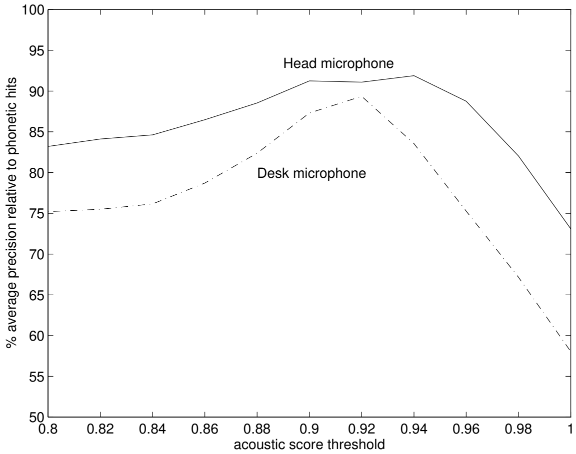

Figure 10: IR results versus threshold.

	Absolute	Text Rel.	Phon. Rel.
Text	0.332	100%	—
Phonetic	0.317	95.5%	100%
Head	0.291	87.6%	91.9%
Desk	0.283	85.2%	89.4%

Table 2: Idf acoustic retrieval performance, relative to text and phonetic.

spotter that reports all keyword phone sequences provides a more legitimate standard of comparison than text. This was simulated by scanning the message phonetic transcriptions for sequences that match keyword phone sequences.

6.3 Acoustic Message Retrieval Performance

The word spotter outputs a list of putative keyword hits and associated acoustic scores. Since the message retrieval scheme uses only the presence/absence of a keyword in a message, the acoustic scores are thresholded such that only hits with a score above the threshold are counted. Figure 10 shows plots of average idf retrieval precision against score threshold for both the head and desk microphones, with precision shown relative to the phonetic standard. The retrieval results appear somewhat sensitive to the absolute threshold value, this factor appears to be more significant for the desk microphone. The best threshold value represents an optimal trade off between the numbers of true hits and false alarms. Table 2 compares acoustic retrieval performance to the ideal text and phonetic standards, at the *a-posteriori* best thresholds. Performance with data from the desk microphone is, as expected, slightly worse overall.

7 Conclusions and Further Work

Our first retrieval system is admittedly limited in scale, and works mainly on an artificial retrieval task. Current plans are to enhance the retrieval system to accommodate open keyword and user sets, allowing a search for arbitrary words spoken by anyone [16]. Using the system in a real-world office environment will undoubtedly raise other issues which must be addressed, such as noise robustness and the actual information content of video mail messages. These should not be insurmountable, however, and the first steps presented here suggest that state-of-the-art information retrieval and word spotting techniques can be successfully combined to provide a useful retrieval environment.

8 Acknowledgements

Olivetti Research Limited is an industrial partner of the VMR project and we are indebted to them for the use of the Medusa system. This project is supported by the UK DTI Grant IED4/1/5804 and SERC Grant GR/H87629.

References

[1] A. Hopper. Pandora – an experimental system for multimedia applications. In *ACM Operating Systems Review*, April 1990.

[2] U. Glavitsch and P. Schäuble. A system for retrieving speech documents. In *Proceedings SIGIR '92*, pages 168–176. ACM, 1992.

[3] A. Hopper. Digital video on computer workstations. In *Proceedings of Eurographics*, 1992.

[4] I. Leslie, D. McAuley, and D. Tennenhouse. ATM Everywhere? *IEEE Network*, March 1993.

[5] S. Wray, T. Glauert, and A. Hopper. The Medusa applications environment. In *Proceedings IEEE International Conference on Multimedia Computing and Systems*, pages 265–273, Boston, May 1994. IEEE.

[6] J. K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, 1994.

[7] L. F. Lamel, H. K. Kassel, and S. Seneff. Speech database development: Design and analysis of the acoustic-phonetic corpus. In *Proceedings of the DARPA Speech Recognition Workshop*, pages 26–32, 1986.

[8] G. J. F. Jones, J. T. Foote, K. Sparck Jones, and S. J. Young. Video Mail Retrieval Using Voice: Report on keyword definition and data collection. Technical Report 335, Cambridge University Computer Laboratory, May 1994.

[9] J. T. Foote, G. J. F. Jones, and S. J. Young. Video Mail Retrieval Using Voice: Report on whole word based keyword spotting. Technical report, Cambridge University Engineering Department, September 1994.

[10] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[11] S. J. Young, P. C. Woodland, and W. J. Byrne. *HTK: Hidden Markov Model Toolkit V1.5*. Entropic Research Laboratories, Inc., 600 Pennsylvania Ave. SE, Suite 202, Washington, DC 20003 USA, 1993.

[12] R. C. Rose. Techniques for information retrieval from speech messages. *Lincoln Laboratory Journal*, 4(1):45–60, 1991.

[13] K. M. Knill and S. J. Young. Speaker dependent keyword spotting for hand-held devices. Technical report, Cambridge University Engineering Department, July 1994.

[14] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[15] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

[16] D. A. James and S. J. Young. A fast lattice-based approach to vocabulary independent wordspotting. In *Proceedings of ICASSP 94*, pages 377–380, Adelaide, 1994. IEEE.