Open-Vocabulary Speech Indexing for Voice and Video Mail Retrieval

M. G. Brown[1] *J. T. Foote*[2] *G. J. F. Jones*[2,3] *K. Spärck 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

email: mgb@cam-orl.co.uk, {jtf,gjfj,sjy}@eng.cam.ac.uk, ksj@cl.cam.ac.uk

ABSTRACT

This paper presents recent work on a multimedia retrieval project at Cambridge University and Olivetti Research Limited (ORL). We present novel techniques that allow extremely rapid audio indexing, at rates approaching several thousand times real time. Unlike other methods, these techniques do not depend on a fixed vocabulary recognition system or on keywords that must be known well in advance. Using statistical methods developed for text, these indexing techniques allow rapid and efficient retrieval and browsing of audio and video documents. This paper presents the project background, the indexing and retrieval techniques, and a video mail retrieval application incorporating content-based audio indexing, retrieval, and browsing.

KEYWORDS:

audio indexing, speech recognition, word spotting, content-based retrieval, information retrieval, browsing

INTRODUCTION

Recent years have seen a rapid increase in the availability and use of multimedia applications. These systems can generate large amounts of audio and video data which can be expensive to store and unwieldy to access. The Video Mail Retrieval (VMR) project at Cambridge University and Olivetti Research Limited (ORL), Cambridge, UK, is addressing these problems by developing systems to retrieve stored video material using the spoken audio soundtrack [3, 25]. Specifically, the project focuses on the content-based location, retrieval, and playback of potentially relevant data. The primary goal of the VMR project is to develop a video mail retrieval application for the Medusa multimedia environment developed at ORL.

Finding the information content of arbitrary spoken documents is a difficult task. This paper presents methods of rapidly and automatically locating words spoken in voice and video mail messages. Unlike other approaches, these techniques do not depend on a limited-vocabulary recognition system or on "keywords" that must be known well in advance of search time [30] [21]. Given an estimate of word locations, we show that statistical retrieval methods can be used for efficient spoken document retrieval and browsing. These exploit a "phone lattice," which represents multiple acoustic hypotheses from which possible word occurrences can be inferred. This paper is organised as follows: after an overview of audio indexing methods, the multimedia environment and corpus of spoken documents is described. Further sections present the speech recognition and information retrieval methods used, and a final section describes how they are combined in a real-time open-vocabulary video mail retrieval system.

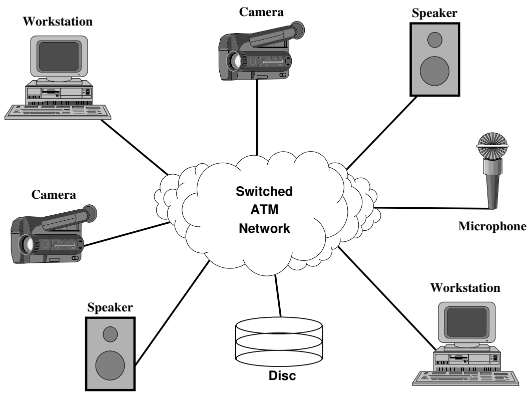

Figure 1. Connection of Medusa Network Endpoints to an ATM network

AUDIO INDEXING

The large computational cost of speech recognition is a fundamental obstacle to automatic audio indexing. Even using the most advanced decoders, recognition speed is seldom much faster than real-time, and often far slower. Though this may be acceptable for typical speech recognition applications such as dictation, it is clearly unacceptable to incur several hours of computation when searching an audio corpus of similar length.

The normal solution is to shift the computational burden ahead of need; thus for a voice-mail retrieval application, the expensive speech recognition is performed as messages are added to the archive. This allows rapid online searches as no recognition is done at search time: only the precomputed index files need be interrogated. This approach has been used in cases where a "word spotting" and/or large-vocabulary recognition system generates text "pseudo-transcriptions" from the audio documents, which may then be rapidly searched. For example, previous work on the VMR project demonstrated practical retrieval of audio messages using fixed-vocabulary word spotting for content identification [12, 5], and more recent work has explored combining word spotting and large-vocabulary recognition for audio retrieval [14].

A similar approach is taken by most other groups working on audio indexing. In the Carnegie Mellon Informedia project, a combination of verbatim text transcriptions and large-vocabulary recogniser output are used for search indexes [23]. A large vocabulary system was used for topic spotting at BBN [16], while workers at Ensigma Ltd used keyword spotting for similar ends [31]. A novel approach is taken at ETH Zürich, where subword units are used as search indexes [28].

With the possible exception of work done at ETH Zürich, the drawback of these approaches is that index terms (the desired keywords or vocabulary) must be known well in advance of search time. A large-vocabulary system has the added drawback that a statistical "language model" (defining likely word tuples) must be available, otherwise the recognition search becomes computationally infeasible. To be useful, language models must be trained on example text (typically megawords) from a (hopefully) similar domain. While this may be possible for some domains (say broadcast news), it is much less practical for others, such as the video mail domain considered here, where there is unlikely to be sufficient training data.

This paper presents an alternative approach where a phone lattice is computed before search time. Though this takes substantial computation, it is less expensive than a large-vocabulary recognition system, and has the additional advantage that it requires no language model. Once computed, the phone lattice may be rapidly scanned using a dynamic-programming algorithm to find index terms [10]. This requires a phonetic decomposition of any desired words, but these are easily found from a dictionary or by a rule-based algorithm [4]. For comparison, the lexicon of our "large" vocabulary experiments was 20,000 words, while our phonetic dictionary has more than 200,000 entries.

This paper reports quantitative experiments demonstrating that Information Retrieval (IR) methods developed for searching text archives can accurately retrieve audio and video data using index terms generated on-the-fly from phone lattices. In addition, the same techniques can be used to rapidly locate interesting areas within an individual mail message. The paper concludes with the description of an on-line video mail retrieval application, including approaches to content-based audio browsing.

MEDUSA: MULTIMEDIA ON AN ATM NETWORK

The Medusa Project at ORL is a novel and extensive experiment in sending multiple simultaneous streams of digital audio and video over a prototype 100 Megabit-per-second switched ATM (Asynchronous Transfer Mode) network [7, 15]. A number of ATM Network Endpoints have been developed enabling the direct connection to the network of microphones, speakers, cameras, disk systems and LCD displays. This concept of exploding the workstation across the network has provided a very adaptable and easily extensible environment for the work presented here. Some 200 of these Network Endpoints cover all laboratory rooms and optical fibre extends the ATM network to the University Engineering Department and the University Computer Laboratory. An ATM network's high bandwidth, low latency, and low transit time jitter make it an ideal transport medium for multimedia applications.

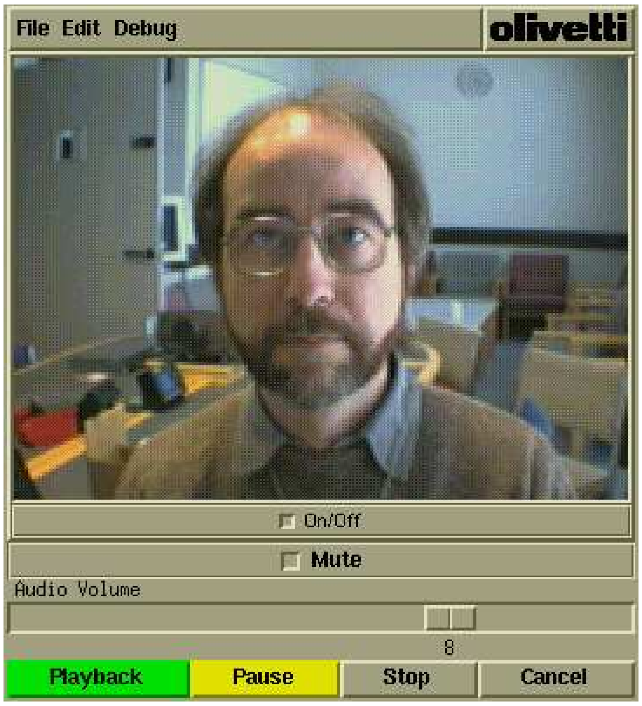

Figure 2. MDMail: Medusa video mail application

The Medusa software developed at ORL handles multimedia in a highly distributed environment. Medusa servers are run on each Network Endpoint and networked workstation in a peer-to-peer architecture. Software objects called Modules created within these servers provide media sources, sinks and pipeline processing components which can be connected together across the network in arbitrary ways. The software modules provide direct digital access to the audio and video data. This architecture enables expensive tasks like video processing and speech recognition to be sited on appropriate hardware, which can then offer its services to any source object connected to the network.

The Multimedia Repository

The Disc Endpoint, which is the size and shape of a small vertical format PC case, uses the ORL standard ATM network interface card plus a SCSI interface to make a RAID-3 array of discs available as a multimedia file server. The initial prototypes use five 2 Gbyte drives giving a storage capacity of 8 Gbytes per unit. Four Disc Bricks are currently deployed on ORL's ATM network. Disk capacities nearly double each year; it is now possible to construct 32 Gbyte devices and we anticipate 64 Gbyte ones within a year.

Medusa ATM camera Endpoints capture frames at a resolution of 176 x 128 pixels at a rate of 25 frames per second. With 5 bits per colour component packed into a 16 bit short word this equates to a raw data rate of 8.8 Megabits per second, though this may be lowered by reducing frame rate or size. Together with the 0.5 Megabits per second required for uncompressed 16-bit audio sampled at 32 KHz, a typical

30 second video mail message amounts to about 35 Mbytes of data. By pragmatically reducing the picture resolution it is possible to store over 1000 video mail messages on an 8 Gbyte device.
Recent developments have made available ATM-networked combined audio and video sources which deliver MPEG compressed data. This reduces these storage requirements dramatically, improving quality at the same time. It is now practical to build an archive containing hundreds of hours of audio/video material. As an example, a 64 Gbyte disc system could store over 90 hours of VHS quality material, equivalent to 11,000 typical video mail messages. If this were MPEG audio only some 88,000 audio mail messages could be stored. As archives with a capacity of this magnitude become more common, better ways of locating and retrieving information become essential.

THE VMR MESSAGE CORPUS

For research into the underlying speech recognition and information retrieval technologies, it was necessary to collect a corpus of mail messages and additional spoken data. (While other spoken data collections exist, they do not have the necessary information content for meaningful IR experimentation.)
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." For the message data, speakers were asked to record a natural response to a prompt (with five prompts per category), for a total of 50 unique prompts. The messages are fully spontaneous, and contain a large number of disfluencies such as "um" and "ah," partially uttered words and false starts, laughter, sentence fragments, and informalities and slang ("'fraid" and "whizzo"). There were 6 messages (from 6 different users) for each of the 50 prompts. A more complete description of the VMR corpus may be found in [11].
There were fifteen speakers, of which 11 were male and 4 female. Data was recorded 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 speaker provided the following speech data:

- **Test:** 20 natural speech messages ("p" data): the response to 20 unique prompts from 4 categories.
- **Train:** Various additional training data for speech recognition models:
  - 77 read sentences ("r" data): sentences containing keywords, constructed such that each keyword occurred a minimum of five times.
  - 170 keywords ("i" data) spoken in isolation.
  - 150 read sentences ("z" data): phonetically-rich sentences from the TIMIT corpus.

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 in accordance with the Wall Street Journal data collection procedures. Phonetic transcriptions were automatically generated from a machine-readable version of the Oxford Learners Dictionary. The standard reduced TIMIT phone set was augmented with additional vowels specific to British English pronunciation. The resulting 300 messages (5 hours of spoken data), along with their text transcriptions, serve as a test corpus for the speech recognition and IR experiments.
For a practical system, it cannot be assumed that speakers will be known, thus it is necessary to have speaker-independent acoustic models. To build such speaker-independent acoustic models, additional training data was obtained from the WSJCAM0 British English corpus, which consists of spoken sentences read from the Wall Street Journal. Data was collected for 100 British English speakers. The corpus contains a total of around 12 hours of spoken data. WSJCAM0 was collected at Cambridge University Engineering Department and further details may be found in [19].

ACOUSTIC INDEXING VIA PHONE LATTICES

Automatically detecting words or phrases in unconstrained speech is usually termed "word spotting;" this technology is the foundation of the work presented here. Conventional keyword spotters based on the same hidden Markov model (HMM) methods used in successful continuous-speech recognition [18]. A hidden Markov model is a statistical representation of a speech event like a word; model parameters are typically trained on a large corpus of labelled speech data. Given a trained set of HMMs, there exists an efficient algorithm for finding the most likely model sequence (the recognised words), given unknown speech data.
The work presented here takes a different approach, based on the work of James [9]. An off-line HMM system is used to generate a number of likely phone sequences, which may then be rapidly searched to find phone strings comprising a desired word. For HMM training and recognition, the acoustic data was parameterised into a spectral representation (mel-cepstral coefficients), and difference and acceleration coefficients were appended. The HTK tool set was used to construct both speaker-dependent (SD) and speaker-independent (SI) monophone models as well as speaker-independent biphone models [34]. All phone models have 3 emitting states, each with 8 Gaussian mixture diagonal-covariance output distributions.

Model Training

For every training utterance, a phone sequence was generated from the text transcription and a dictionary. These sequences were used to estimate HMM parameters as follows. Speaker-dependent "monophone" models were trained on the read messages ("r" data) and sentences from the TIMIT database ("z" data). Once single-mixture monophone models had been initialised, the number of mixture components was increased, and the parameters re-estimated. Re-estimation was halted at 8 mixture components, as additional components did not improve performance. Speaker-independent models were trained in a similar manner on the WSJCAM0 corpus of read speech, which contains more than one hundred speakers.
Though we rarely notice, phone pronunciation changes drastically depending on context (contrast the "T" sound in "attack" and "stain.") Automatic speech recognition improves substantially when phone models can be made context-dependent, thus the two "T" sounds above would have separate models because they occur in different contexts. Speaker-independent "biphone" models were constructed by "cloning" the 1-mixture speaker-dependent monophones such that each possible biphone was represented, then clustering similar states using a decision tree [32]. State parameters are tied across a cluster, then re-estimated in the usual way, once again up to 8 mixtures. An advantage of

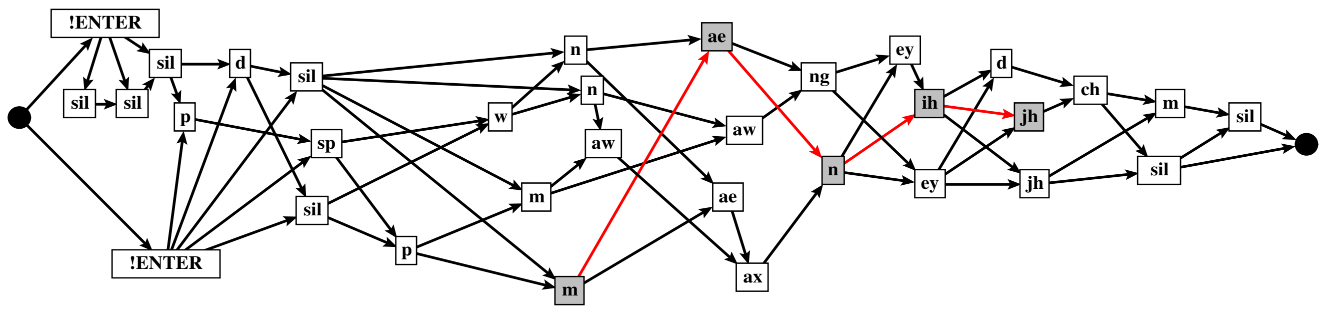

Figure 3. Phone lattice for word “manage” (`m ae n ih jh`)

this training method is that all possible biphones are modelled, yet because most states are tied, the full model set is relatively compact. There was insufficient training data to construct speaker-dependent biphones.

Lattice Generation

With a set of phone HMMs and a network of possible phone interconnections, it is possible to find the most likely sequence of phones given unknown acoustic data using an efficient search based on the Viterbi algorithm. An enhanced version of this algorithm can generate multiple hypotheses, representing the n most likely paths through the models. Such multiple hypotheses can be stored as a phone lattice: a directed acyclic graph where nodes represent phone start/end times and arcs represent hypothesised phone occurences between them. Arcs are labelled with an acoustic score indicating the likelihood that the acoustic data over that interval corresponds with the particular phone model. To simplify searching, the lattices used here have the additional constraint that all paths must start at a particular start node and end at another special node.

The “depth” of a phone lattice is the number of phone hypotheses active at a given time. This parameter is critical to the performance of a lattice-based spotting system. If the lattice is too shallow, performance will be poor due to unavoidable phone recognition errors. Because the best phone recognition systems are little more than 70% accurate [20], the chance that a given phone string will be correctly identified in a 1-deep lattice is poor. On the other hand, if the lattice is too deep, too many phone sequences become possible, most of which will be incorrect. Another drawback is that the storage requirements and search time increase substantially with lattice depth. Failing to identify an uttered search word is termed a *miss* while hypothesising a word where none is present is a *false alarm*.

Lattice depth may be adjusted through several mechanisms. During generation, the number of active tokens at one time represents the n-best paths through the model lattice [33]. The more tokens used, the deeper the lattice. To speed lattice generation, a pruning threshold is used: this ensures that low-likelihood phone paths are not considered, saving substantial computation. The lower the pruning threshold, the more possible paths considered, and hence the deeper the resultant lattice. Figure 3 shows a lattice generated for the single utterance “manage.” For clarity, acoustic scores and start/end times are not shown, though nodes are arranged in roughly chronological order. Four tokens were used in the decoding: the resultant lattice depth was 5 as 35 arcs were generated for the 7 phones actually uttered (`m ae n ih jh` plus beginning and ending silences).

Model type:	SI monophones	SD monophones	SI biphones
Phone Accuracy	41.1%	55.4%	51.7%
Figure of Merit	48.0%	73.6%	60.4%

Table 1. Speech recognition results for VMR corpus messages

Lattice Scanning

Once the lattices have been computed, it is relatively straightforward to scan them for a phone sequence constituting a given word. Once a string of the correct phones has been found, an acoustic score for the putative term is computed as the sum of the phone arc scores normalised by the best-path score. Deep lattices will result in many hypotheses for a given word (because of different paths starting or ending within a few milliseconds) so overlapping word hypotheses are eliminated by keeping only the best scoring one. For example, Figure 3 shows two possible paths for the phone sequence `m ae n ih jh`, because of the two instances of the phone `jh` following `ih`. This scanning procedure can be made extremely time-efficient, producing hypotheses in the order of a thousand times faster than the source audio.

Speech Recognition Results

Table presents the results of lattice scanning experiments using 6 tokens, on the VMR corpus messages. The phone accuracy is defined as the ratio of correctly recognised phones, minus deletions and insertions, to the actual number of phones, for the best path through the lattice. (Experiments on the WSJCAM0 read-speech corpus using the monophone models resulted in phone accuracies nearer 60%, indicating that the natural-speech VMR corpus is more difficult than read speech to recognise.)

Putative hits generated by a word spotting system generally have an associated acoustic score. Because low-scoring words are more likely to be false alarms, the operating point of the recognition system may be adjusted by ignoring terms with a score below a given threshold. Words with scores above the threshold are considered true hits, while those with scores below are considered false alarms and ignored. Choosing the appropriate threshold is a tradeoff between the number of Type I (miss) and Type II (false alarm) errors, with the usual problem that reducing one increases the other. The accuracy of a word spotter is thus dependent on the threshold and cannot be expressed as a single number if false alarms are taken into account. An accepted figure-of-merit (FOM) for word spotting is defined as the average percentage of correctly detected words as the threshold is varied from one to

Weight Scheme		Full Vocab. *uw*	*cfw*	*cw*
Precision	5 docs	0.392	0.375	0.371
	10 docs	0.313	0.308	0.344
	15 docs	0.279	0.292	0.308
	20 docs	0.250	0.271	0.290
Average Precision		0.327	0.352	0.368

Table 2. Retrieval precision values using full text transcriptions (VMR Collection 1b)

ten false alarms per word per hour. These types of recognition error effect not only speech recognition but also information retrieval performance [12]. A drawback of the lattice scan approach is that it is not robust for short words – using lattices of the depth necessary to detect longer words results in a large number of false alarms for shorter ones. Though more sophisticated scoring mechanisms might improve this, the solution used here was to ignore very short words (3 or fewer phones), which are typically not information-rich anyway.

For comparison, the best FOMs obtainable using speaker-independent keyword models was 69.9%, but this additional accuracy was at the cost of having to explicitly search for the 35 particular keywords in the recognition phase, which is several orders of magnitude slower than the lattice-scan approach [5].

INFORMATION RETRIEVAL VIA ACOUSTIC INDEXES

Once words can be located in a speech corpus (using the techniques just described) they may be used for content-based message retrieval, by applying Information Retrieval (IR) techniques originally developed for text. IR techniques attempt to satisfy a user's information need by retrieving potentially relevant messages from a document archive.

In practice, the user composes a search "request" as a sentence or list of words from which a set of actual search "terms" is derived. A score can be computed for a document from the number or weights of matching "query" terms. Searching an archive of documents will deliver an output with documents ranked by matching score. The user can then browse high-ranking messages in this output to find desired information.

Prior to retrieval, conventional IR systems compute an "inverted file" structure where documents are indexed by term. This allows extremely rapid retrieval because documents containing a given term can be quickly located. In the VMR system described here, the actual word-level contents of message are unknown until search time and hence it is not possible to build the inverted file structure in advance. When a request is entered, the lattices are scanned for each search term, as described in the previous section; the putative hits are then used to construct an inverted file for retrieval. Ongoing retrieval efficiency is improved by preserving all lattice scan results in the inverted file, so that effort is not duplicated scanning for a term more than once.

Requests and Relevance Assessments

Evaluating retrieval performance is central to IR research. Evaluating an IR system requires a set of message requests, together with assessments of the *relevance* of each message to each of these requests. Though some previous experiments [3, 12] used a simulated request and assessment set, a more realistic set has since been collected from the user community that supplied the database messages, forming VMR Collection 1b.

A total of 50 text requests were collected from 10 users, each of whom generated 5 requests and corresponding relevance assessments. A request prompt for each category was formed from the 5 message prompts associated with the category. Users were asked to compose a natural language request after being shown the request prompt.

Ideally, the relevance of all archived messages should be assessed; however this is not practical even for our 300 message archive (which is considered a very small archive by the standards of text IR). A practical alternative is to assess only a subset chosen to contain (hopefully) all the relevant messages. A suitable assessment subset was formed by combining the 30 messages in the category to which the original message prompt belonged, plus 5 messages from outside the category having the highest query-message scores (using *cfw* weights, as in Section). Subjects were presented with the transcription of each message and asked to mark it as "relevant", "partially relevant", or "not relevant" to the request they had just composed. Messages were presented in random order to avoid possible sequencing effects during assessment. The following sections report results only for the set of messages assessed as highly relevant.

Text Preprocessing

In standard text retrieval systems, documents and requests are typically preprocessed to improve retrieval performance and storage efficiency. The first stage is usually to remove function words (such as "the," "which") which do not help retrieval. The remaining words are then stemmed to reduce word form variations that inhibit term matching between documents and requests.

Retrieval performance for spoken documents can be expected to suffer degradation due to recognition errors (either misses or false alarms on the search terms). In the VMR project, IR benchmarks are established using the full text transcriptions of the 300 messages. The relative degradation can then be computed by comparing retrieval performance with that for the text transcriptions.

The text transcriptions as well as the written requests were therefore preprocessed before search. Function words were removed using a standard "stop list" [27]. The remaining words were reduced to stems using a standard algorithm due to Porter [17]. For example, given the request

```
In what ways can the windows interface of a
workstation be personalised?
```

the following query is obtained:

```
wai window interfac workstat personalis
```

As the lattice is scanned only for the terms in the query, stop words are ignored, which is fortunate because they are generally short, and thus may not be reliably located by the lattice scanning method. The issue of suffix stripping is slightly more complex. There are several options here: search only for the term as it appears in the request, search for the suffix stripped term, search for all terms which reduce to the same stem as the request term, or search for the shortest dictionary entry which reduces to this stem. The option taken here is to search for the term as it appears in the original request, although the other options are under investigation. Also the phonetic dictionary was expanded to include all search terms and hence rule-based phonetic prediction was not needed. Finally, search terms having 3 or

Weight Scheme		Average Precision (Relative %)		
		uw	*cfw*	*cw*
Text	Full Vocab	0.327 (100%)	0.352 (100%)	0.368 (100%)
Spoken Documents	SD monophones	0.262 (80.1%)	0.285 (81.0%)	0.315 (85.6%)
	SI monophones	0.174 (53.2%)	0.199 (56.5%)	0.222 (60.3%)
	SI biphones	0.224 (68.5%)	0.262 (74.4%)	0.277 (75.3%)

Table 3. Absolute and relative Average Precision for different lattice acoustic models (VMR Collection 1b)

fewer phones were excluded since they generate too many false alarms with the lattice depths used. For example, the word "date" was not searched for, as its typical phone decomposition (`d ey t`) was only three phones long. Though it is clearly not optimal to discard search terms, less than 10% of query terms not in the stop list were too short, and their absence did not harm retrieval performance unduly.

Message Scoring

Given a query, an estimate of each message's relevance depends on the number of terms in common. This estimate gives the query-document matching score, and allows all messages in the archive to be ranked in terms of potential relevance [27]. Considering search term presence/absence only, the simplest scoring method is just to count the number of terms in common, often called the *unweighted* (*uw*) score. Better retrieval can be achieved by weighting terms, for instance by the *collection frequency weight* (*cfw*),

$$cfw(i) = \log \frac{N}{n[i]}$$

where N is the total number of documents and $n[i]$ is the number of documents that contain search term i. This scheme favours rarer (and hence more selective) terms. The query-document matching score is then the sum of the matching query term weights. A more sophisticated weighting scheme takes into account the number of times each term occurs in each document, normalised by the document length. This latter factor is important since a document's relevance does not depend on its length, hence neither should its score. The well tested *combined weight (cw)*, described further in [26], is

$$cw(i,j) = \frac{cfw(i) \times tf(i,j) \times (K+1)}{K \times ndl(j) + tf(i,j)}$$

where $cw(i,j)$ represents the *cw* weight of term i in document j, the term frequency $tf(i,j)$ is the frequency of i in j, and $ndl(j)$ is the normalised document length. $ndl(j)$ is calculated as

$$ndl(j) = \frac{dl(j)}{\text{Average } dl \text{ for all documents}},$$

where $dl(j)$ is the length of j. The combined weight constant K must be determined empirically; after testing we set $K = 1$.

For full-text documents, the document length $dl(j)$ is simply the number of terms in document j. However, when deriving search terms from the phone lattice the situation is less clear. The simplest estimate of $dl(j)$ is the number of search terms actually located in the document. This is unsatisfactory for several reasons: for example, a short document with a large number of false alarms may appear comparatively long. This motivates other estimates of document length, such as its length in time. Our experiments indicate that a better representation of $dl(j)$ is the number of phones in the the most likely phone sequence. This is easily computed during speech recognition and is intuitively a reasonable measure since the number of phones should be independent of speaking rate and hence is a better measure of the number of words actually spoken.

Measuring IR Performance

Given a query, a matching score may be computed for every document in the archive using the methods just described. Documents can then be ranked by score so that the highest-scoring documents (potentially the most relevant) occur at the top of the list. Retrieval performance is often measured by *precision*, the proportion of retrieved messages that are relevant to a particular query at a certain position in the ranked list. One accepted single-number performance figure is the *average precision*. For each query, the precision values are averaged for each relevant document in the ranked list. The results are then averaged across the query set, resulting in the average precision. Other less reductive retrieval evaluation metrics are available and generally preferable, but this single-number performance measure is a useful basic performance indicator.

Retrieval Results

This section presents experimental retrieval results for VMR Collection 1b, both for text transcriptions and for indexes derived from the monophone and biphone lattices discussed in Section . All results are for the *a posteriori* best acoustic threshold, and all *cw* scores estimate the document length as the number of phones. The basic comparisons are between monophone and biphone results, and between SI and SD model results. However, it is helpful to set this spoken document retrieval performance against reference performance for text.

Thus Table 2 shows retrieval performance using full open-vocabulary text transcriptions. These results confirm that more sophisticated weighting schemes improve retrieval performance. Note that the average precision score is the average of the precision values for the relevant documents in the ranked list. Figure 3 shows retrieval performance for the various models and weighting schemes for the spoken documents. Absolute average precision is shown, and also that relative to the average precision obtainable from text (which may be considered the best possible). Clearly the *cw* scheme produces the best retrieval performance. In addition, retrieval performance is well-correlated with speech recognition accuracy. Thus the SI biphones are better than the SI monophones, but the SD monophones are better still. It is not surprising that the SD monophones resulted in the best retrieval performance as they had the best phone recognition accuracy. But SD models limit the usefulness of the VMR system and hence the extension to SI modelling is important. Though the SI biphones do not perform as well as

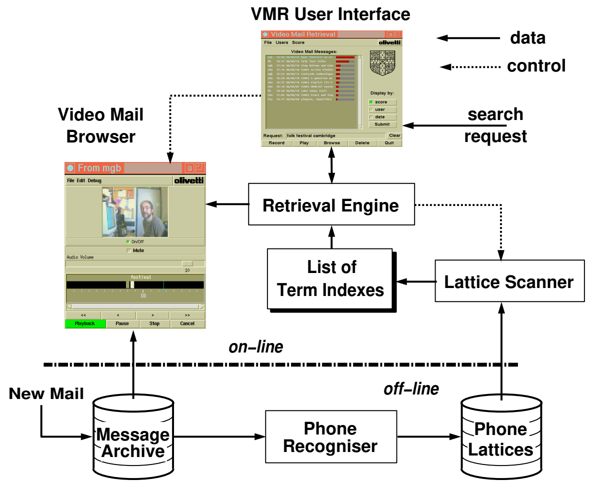

Figure 4. Block diagram of video mail retrieval system

Figure 5. Video Mail User Interface application

the SD monophones, the speaker-independent models allow messages uttered by any speaker to be retrieved. It should be noted that one of the speakers is a native American and another speaker's accent is strongly influenced by his central European background. The British English models used here are not well matched to these speakers resulting in lower recognition performance. Nevertheless, reasonable retrieval performance can be obtained. Clearly spoken document retrieval is less good than the text case, but not disastrously so. More importantly, with the best weighting scheme (cw) the speaker independent biphone models perform respectably in themselves and not much worse than the speaker-dependent case.

A REAL-TIME VIDEO MAIL RETRIEVAL APPLICATION

At some point, results from both the keyword spotting and information retrieval must be presented to the user. 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. Using information in the mail message header, 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.

A natural addition to this scheme is to add message attributes that reflect the information content of the audio portion, as determined using the retrieval methods described previously. In operation, the user types in a text search query. The resulting score for each message is computed by the retrieval engine, and the interface then displays a list of messages ranked by score. Scores are represented by bar graphs, as in Figure 5; messages with identical scores are ranked by time. In its simplest form, the keyword search resembles an "audio grep" that returns a list of messages containing a particular keyword. Because the search is phonetic rather than textual, certain "tricks" can be used to enhance search effectiveness:

- Concatenation: A "+" in the query instructs the dictionary lookup routine to concatenate two words into one long phone string. Example: to find the word "`netscape`" search for "`nets+cape`."
- Word Stems: If a given search fails on a long word, shorter variants may work better. Example: To find "`managerial`" search for "`manage`."
- Homophones: If a given word is not in the dictionary, it may still be found using a homophone (exact rhyme). Example: to find "`Basque`" search for "`bask`."
- Phonetic representation: If not in the dictionary or amenable to the previous approaches, a word may still be found by entering its phonetic composition directly. Example: "`Yeltsin`" = "`#y+#eh+#l+#t+#s+#ih+#n`." The initial "`#`" is necessary to distinguish single-letter phones from single-letter words (eg "`#b`" $\neq$ "`B`" = "`#b+#iy`"). A help menu is available that displays a list of phones and their pronunciations.

All the above approaches may be used in conjunction; for example the following query would be useful to find the term "`ATM`" : "`A+T+M eighty`"

Figure 4 shows a block diagram of the video mail retrieval application. All archived messages have corresponding lattices, generated when the mail message was added to the archive. When the user types a search request into the interface, the information retrieval engine interrogates the inverted file to determine whether any previously unseen terms are present. If so, the lattice scan engine locates term occurrences in the available lattices. These new hypotheses are then added to the inverted file and preserved for future reference. The actual search time for unseen terms, though extremely rapid, is not instantaneous. In our demonstration system, searching more than an hour of audio data for these terms will take less than five seconds for a typical request. Once all available term hypotheses have been located, a ranked list of messages is computed, and returned to the GUI.

A Video Message Browser

After the ranked list of messages is displayed, the user must still investigate the listed messages to either find the relevant one(s) or determine that the retrieval was ineffective and that a new search is required. While there are convenient methods for the graphical browsing of text, e.g. scroll bars, "page-forward" commands, and word-search functions,

Figure 6. Video mail browser showing detected keywords

existing video and audio playback interfaces almost universally adopt the "tape recorder" metaphor. To ensure that nothing important is missed, an entire message must be auditioned from start to finish which takes significant time. In contrast, the transcription of a minute-long message is typically a paragraph of text, which may be scanned by eye in a matter of seconds. Even if there is a "fast forward" button it is generally a hit-or-miss operation to find a desired section in a lengthy message. We can, however use the term indexes to provide a reliably economical way to access and review audio/video data.

Our browser is an attempt to represent a dynamic time-varying process (the audio/video stream) by a static image that can be taken in at a glance. A message is represented as horizontal timeline, and events are displayed graphically along it. Time runs from left to right, and events are represented proportionally to when they occur in the message; for example, events at the beginning appear on the left side of the bar and short-duration events are short. When a particular file is selected for browsing, the list of term indexes (including exact times of search term occurrences) is available to the browser from the inverted file. For a selected message, the browser contents are computed dynamically for the current query by following a linked-list through the inverted file which links hypotheses pertaining to this particular message. Potentially interesting portions of the message are thus easily identified.

A simple representation is to display putative keyword hits on the timeline, as in Figure 6. The timeline is the black bar; the scale indicates time in seconds. When pointed at with the mouse, keyword names are highlighted in white (so it may be read in the presence of many other keyword hits). Clicking on the desired time in the time bar starts message playback at that time; this lets the user selectively play regions of interest, rather than the entire message. Figure 6 shows the keyword hits "folk" and "festival" (highlighted) displayed on the time bar, starting about eight seconds into the message; a time cursor (the triangle-bar) indicates the current playback time. This approach may be enhanced by displaying different search terms in different colours, and weighting them according to term or document frequency (so less discriminating search terms appear less prominent in the display).

Due to false alarms, displaying all putative hits can sometimes give a misleading impression of the true relevance. Another approach to content-based browsing has been motivated by our work with broadcast news retrieval, where teletext transcriptions were used as indexing sources [2]. Individual keyword hits are not displayed in this approach; rather the message is considered as a series of overlapping "windows," consisting of a short, fixed interval. A query-window matching score can be computed for each interval, just as for an entire document. The window scores can then be displayed such that the higher-scoring windows appear brighter and more prominent. The resulting display is essentially a "low-pass filtered" version of the putative hits, and is thus less time-precise but less cluttered as well, and automatically incorporates the term weighting factors so that less-discriminating terms are not given undue importance. A similar approach is adopted in the "TileBars" data visualisation tool developed by Hearst [6], which displays document length and term relevance in the initial ranked list. This type of output could easily be incorporated into our ranked list display, allowing users to make a more informed choice of documents prior to browsing.

Video Cues

For the video mail application, we have focussed on the audio stream because that is where nearly all information of practical interest will be found. It is, in general, much more difficult to extract useful information from a video signal[1]. Image retrieval is a challenging task and many problems remain unsolved. Most work does not extend much beyond simple measures of colour or shape similarity [24, 1], although there is promising work based on wavelet analysis [8]. While efforts in face and gesture recognition are in progress at ORL and elsewhere [22], less sophisticated analyses can still yield information helpful for browsing. For example, a Medusa video analyser module can detect the activity in a video stream; one application uses this activity information to automatically select a preferred video stream from several available in each room. One simple strategy is to determine the "activity" of a video stream from an estimate of frame-to-frame difference. A large, impulsive value can indicate a new scene or camera, while moderate values over a period of time indicate subject or camera motion. Near-zero values mean a static (and therefore uninteresting) image. Though not yet implemented, current plans are to add this activity information to the browser, enabling automatic detection of a camera or scene change. In this case, a "thumbnail" image of the new view would be displayed on the timeline. Also, active areas could be highlighted to indicate that something of potential interest is occurring in the video stream.

FUTURE WORK AND CONCLUSIONS

The work presented here is only the latest step towards general audio and video retrieval. Previous work, by the VMR

[1] This is particularly true in the video mail environment, where the vast majority of messages are just "talking head" images from a small pool of users, against static backgrounds.

group and by others, has shown that spoken document retrieval using speech recognition is becoming practical [12, 9]. Future work in this project will be to integrate different audio index sources available from large vocabulary recognition and conventional as well as lattice-based word spotting [14, 13]. In addition, work will need to be done to make the system robust to environmental noise, microphone differences, accent variability, and telephone-bandwidth speech. Another promising area is to use other types of audio information, such as speaker or music identification, to help index multimedia streams [29]. In conclusion, this paper presents useful methods of indexing audio and video sources, and demonstrates a real-time audio retrieval application, although still on a small scale. Much more work needs to be done on scaling up, especially on handling large numbers of documents and their correspondingly large lattices. But as multimedia archives proliferate on the WWW and elsewhere, technology like that presented here will be become indispensable to locate, retrieve, and browse audio and video information.

ACKNOWLEDGEMENTS

The authors thank the following people for their invaluable assistance: David James for useful discussions, Steve Hedge for the network editor used to create Figure 3, Frank Stajano for Medusa application libraries, Julian Odell for decoder improvements, and Tony Robinson for compiling the BEEP British English pronunciation dictionary.
Olivetti Research Limited is an industrial partner of the VMR project. This project has been supported by the UK DTI Grant IED4/1/5804 and SERC (now EPSRC) Grant GR/H87629.

REFERENCES

1. R. Barber, C. Faloutsos, M. Flickner, J. Hafner, W. Niblack, and D. Petkovic. Efficient and effective querying by image content. *J. Intelligent Information Sys.*, (3):1–31, 1994.
2. M. G. Brown, J. T. Foote, G. J. F. Jones, K. Spärck Jones, and S. J. Young. Automatic content-based retrieval of broadcast news. In *Proc. ACM Multimedia 95*, pages 35–43, San Francisco, November 1995. ACM.
3. M. G. Brown, J. T. Foote, G. J. F. Jones, K. Spärck Jones, and S. J. Young. Video Mail Retrieval using Voice: An overview of the Cambridge/Olivetti retrieval system. In *Proc. ACM Multimedia 94 Workshop on Multimedia Database Management Systems*, pages 47–55, San Francisco, CA, October 1994.
4. C. Coker, K. Church, and M. Liberman. Morphology and rhyming: two powerful alternatives to Letter-to-Sound rules for speech synthesis. In *ESCA Workshop on Speech Synthesis*, pages 83–86, Autrans, France, Sept 1990. ECSA.
5. J. T. Foote, G. J. F. Jones, K. Spärck Jones, and S. J. Young. Talker-independent keyword spotting for information retrieval. In *Proc. Eurospeech 95*, volume 3, pages 2145–2148, Madrid, 1995. ESCA.
6. M. A. Hearst. TileBars: Visualisation of term distribution information in full text information access. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, Denver, CO, May 1995. ACM.
7. A. Hopper. Digital video on computer workstations. In *Proceedings of Eurographics*, 1992.
8. C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *Proceedings of the SIGGRAPH 95 Conference*, pages 277–286, Los Angeles, CA, August 1995. ACM SIGGRAPH.
9. D. A. James. *The Application of Classical Information Retrieval Techniques to Spoken Documents.* PhD thesis, Cambridge University, February 1995.
10. D. A. James and S. J. Young. A fast lattice-based approach to vocabulary independent wordspotting. In *Proc. ICASSP 94*, volume I, pages 377–380, Adelaide, 1994. IEEE.
11. G. J. F. Jones, J. T. Foote, K. Spärck Jones, and S. J. Young. VMR report on keyword definition and data collection. Technical Report 335, Cambridge University Computer Laboratory, May 1994.
12. G. J. F. Jones, J. T. Foote, K. Spärck Jones, and S. J. Young. Video Mail Retrieval: the effect of word spotting accuracy on precision. In *Proc. ICASSP 95*, volume I, pages 309–312, Detroit, May 1995. IEEE.
13. G. J. F. Jones, J. T. Foote, K. Spärck Jones, and S. J. Young. Retrieving spoken documents by combining multiple index sources. In *Proc. SIGIR 96*, Zürich, August 1996. ACM.
14. G. J. F. Jones, J. T. Foote, K. Spärck Jones, and S. J. Young. Robust talker-independent audio document retrieval. In *Proc. ICASSP 96*, volume I, pages 311–314, Atlanta, GA, April 1996. IEEE.
15. I. Leslie, D. McAuley, and D. Tennenhouse. ATM Everywhere? *IEEE Network*, March 1993.
16. J. McDonough, K. Ng, P. Jeanrenaud, H. Gish, and J. R. Rohlicek. Approaches to topic identification on the switchboard corpus. In *Proc. ICASSP 94*, volume I, pages 385–388, Adelaide, 1994. IEEE.
17. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.
18. L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, February 1989.
19. T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals. WSJCAM0: A British English speech corpus for large vocabulary continuous speech recognition. In *Proc. ICASSP 95*, pages 81–84, Detroit, May 1995. IEEE.
20. T. Robinson, M. Hochberg, and S. Renals. IPA: Improved phone modelling with recurrent neural networks. In *Proc. ICASSP 94*, volume 1, pages 37–40, Adelaide, SA, April 1994.
21. R. C. Rose. Techniques for information retrieval from speech messages. *Lincoln Laboratory Journal*, 4(1):45–60, 1991.
22. F. Samaria and S. J. Young. A HMM-based architecture for face identification. *Image and Vision Computing*, 12(8):537–543, October 1994.
23. M. A. Smith and M. G. Christel. Automating the creation of a digital video library. In *Proc. ACM Multimedia 95*, pages 357–358, San Francisco, November 1995. ACM.
24. S. W. Smoliar and H. J. Zhang. Content-based video indexing and retrieval. *IEEE Multimedia*, 1(2):62–72, Summer 1994.

25. K. Spärck Jones, J. T. Foote, G. J. F. Jones, and S. J. Young. Spoken document retrieval — a multimedia tool. In *Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 1–11, Las Vegas, April 1995.

26. K. Spärck Jones, G. J. F. Jones, J. T. Foote, and S. J. Young. Experiments in spoken document retrieval. *Information Processing and Management*, 32(4):399–417, 1996.

27. C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

28. M. Wechsler and P. Schäuble. Speech retrieval based on automatic indexing. In C. J. van Rijsbergen, editor, *Proceedings of the MIRO Workshop*, University of Glasgow, September 1995.

29. L. Wilcox, F. Chen, and V. Balasubramanian. Segmentation of speech using speaker identification. In *Proc. ICASSP 94*, volume S1, pages 161–164, Adelaide, SA, April 1994.

30. L. D. Wilcox and M. A. Bush. Training and search algorithms for an interactive wordspotting system. In *Proc. ICASSP 92*, volume II, pages 97–100, San Francisco, 1992. IEEE.

31. J. H. Wright, M. J. Carey, and E. S. Parris. Topic discrimination using higher–order statistical models of spotted keywords. *Computer Speech and Language*, 9(4):381–405, Oct 1995.

32. S. J. Young, J. J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proc. ARPA Spoken Language Technology Workshop*, Plainsboro, NJ, 1994.

33. S. J. Young, N. H. Russell, and J. H. S. Thornton. Token passing: a simple conceptual model for connected speech recognition systems. Technical Report CUED/F-INFENG/TR.38, Cambridge University Engineering Department, July 1989. ftp://svr-ftp.eng.cam.ac.uk/pub/reports/young_tr38.ps.Z.

34. 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.