ACOUSTIC INDEXING FOR MULTIMEDIA RETRIEVAL AND BROWSING

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

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

ABSTRACT

This paper reviews *the Video Mail Retrieval (VMR)* project at Cambridge University and ORL. The VMR project began in September 1993 with the aim of developing methods for retrieving video documents by scanning the audio soundtrack for keywords. The project has shown, both experimentally and through the construction of a working prototype, that speech recognition can be combined with information retrieval methods to locate multimedia documents by content. The final version of the VMR system uses pre-computed phone lattices to allow extremely rapid word spotting and audio indexing, and statistical information retrieval (IR) methods to mitigate the effects of spotting errors. The net result is a retrieval system that is open-vocabulary and speaker-independent, and which can search audio orders of magnitude faster than real time.

1. INTRODUCTION

Recent years have seen a rapid increase in the availability 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 ORL (formerly Olivetti Research Limited and now Olivetti and Oracle Research Laboratory) has addressed these problems by developing systems to retrieve stored video material using the spoken audio soundtrack. The project has specifically focused on the content-based retrieval and browsing of video mail messages.

Earlier versions of the VMR system used a fixed keyword vocabulary and conventional word spotting [1, 2]. The final VMR system is shown in Fig 1 and it uses phone-lattices to allow the use of unrestricted keywords[3]. To retrieve a desired message, a user enters a text search request and the precomputed phone lattice attached to each message is examined for occurrences of the search words. These occurrences are then combined using an information retrieval (IR) metric to give a score for each document. Finally, the retrieved documents are presented in score order to the user who can then browse through them using a graphical user interface.

The remainder of this paper describes the phone-lattice based acoustic indexing, the information retrieval mechanism and the user interface. Also included are experimental results comparing the relative performance of phone-lattice based and conventional keyword spotting.

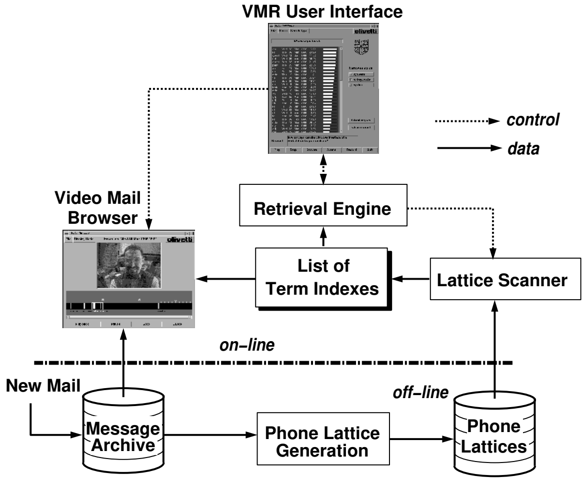

Figure 1. Block diagram of video mail retrieval system

2. ACOUSTIC INDEXING VIA PHONE LATTICES

In contrast to many applications of speech recognition, document retrieval requires large quantities of data to be scanned many times faster than real-time. For practical systems, this implies that most of the data needed for retrieval operations must be precomputed. If word recognition were used then conventional inverted indices could be constructed to allow very rapid retrieval. However, current large vocabulary recognisers typically yield poor accuracy on unrestricted spoken material and furthermore, many of the most important keywords (e.g. names, places) will not be in the LVR system's vocabulary.

To avoid these problems, the VMR system uses precomputed phone lattices as the basis of its acoustic indexing. When each video document is added to the database, a phone lattice representing multiple phone hypotheses is computed for the soundtrack. 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 other than a phone bigram. Also, as noted above, it does not suffer from the problem of out-of-vocabulary words.

Once computed, the phone lattice may be rapidly scanned to find instances of an arbitrary search word. Phonetic decompositions are easily found from a either a dictionary or by a rule-based algorithm. Figure 2 shows a lattice generated for the single utterance "manage" where the correct path is shown in gray. For clarity, acoustic scores and

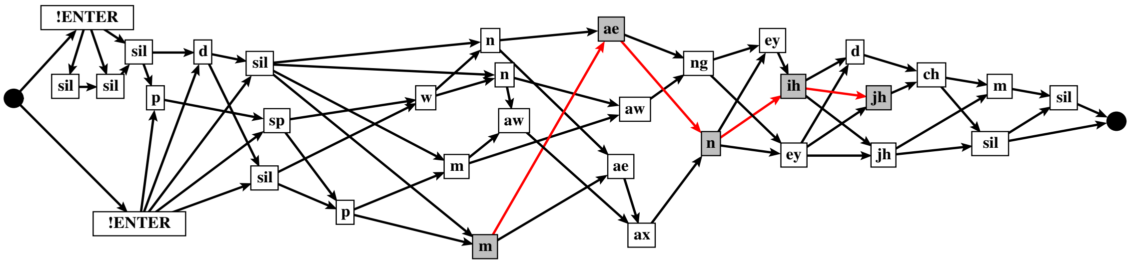

Figure 2. Phone lattice for word "manage" (m ae n ih jh)

start/end times are not shown, though nodes are arranged in roughly chronological order.

Lattice Generation In the VMR system, phone lattices are generated using a set of speaker independent tied-state right biphone HMMs. These were trained on the WSJ-CAM0 British English database parameterised using 12 MFCC coefficients plus energy and 1st and 2nd differentials. When tested on the standard VMR message corpus[4], the phone recognition accuracy was 52%.

Decoding uses a simple extension to the token passing implementation of the basic Viterbi algorithm[5]. In this scheme, each partial state/frame alignment path is represented by a token which is propagated from state to state at each time step. Paths are iteratively extended by examining the tokens for all connecting states, adding the transition and output log probabilities, and then propagating only the best token. The token propagated into a phone q is selected by choosing the most likely token exiting from all connecting phones p_i. Each token records its history in a chained list of phone transition records. Every time a token t transits from phone p_i to another phone q, the identity of p_i (and the current time) is appended to t's history list.

A simple way to generate multiple phone hypotheses is to allow each state to hold multiple tokens and then to record at each phone boundary not just the phone p_i holding the best token, but the set of phones $\{p_i\}$ holding the N-best tokens. To do this efficiently, it is necessary to discard the least likely tokens in a set of tokens with equivalent histories. In the implementation used here, two histories are regarded as equivalent if they end in the same phone.

The depth of the generated lattices (number of phones in the lattice per utterance phone) is controlled by the normal beam search pruning mechanism and it is a key factor in determining system performance. If the lattice is too shallow, performance will be poor due to deleting phones from wanted keywords. On the other hand, if the lattice is too deep, too many phone sequences become possible, most of which will be incorrect. Furthermore, the storage requirements and search time increase substantially with lattice depth.

Phone Lattice Scanning Once computed, a phone lattice may be rapidly searched to find keywords. To simplify searching and storage, an assumption is made that any phone starting at time $t+1$ may follow a phone ending at time t. Ignoring the detailed lattice connectivity results in large storage and I/O savings by preserving just the start and stop times, acoustic score, and phone identity for each lattice edge.

The actual search strategy used to find all examples of a keyword in a lattice is as follows. A list is kept of candidate phone sequences in the lattice which match (i.e. are a prefix of) the keyword being sought. For each candidate c, a record is kept of the total acoustic score, the current end time of the sequence and the next phone p_c required to match the keyword.

At each time frame t the following steps are taken:

1. For each complete candidate sequence c ending at time t, calculate a normalised acoustic score for the sequence, record the keyword instance and delete c.
2. For each incomplete candidate sequence c ending at time t, if the lattice contains an instance of p_c starting at time t then extend the candidate sequence otherwise delete it.
3. For each instance of p_s in the lattice where p_s is the first phone in the keyword, create a new candidate sequence.

The scores are normalised by scaling the log likelihood of phone sequence corresponding to the keyword instance by the log likelihood of the best possible phone sequence spanning the duration of the hypothesised keyword. Deep lattices will result in many hypotheses for a given word, so overlapping word hypotheses are eliminated by discarding all but the highest-scoring one.

The detailed implementation of the above scanning procedure involves a speed/memory tradeoff. In the VMR system, a 1-D lookup table is used where each element points to a linked list of phones starting at that time, in effect hashing on the phone's start time. The linked list is then traversed to determine whether the desired phone starts at the given time. Although this is slower than using a full 2-D table, it is much more memory-efficient and it is still very fast, producing hypotheses in the order of a thousand times faster than would be obtained by word-spotting directly on the source audio waveforms.

3. INFORMATION RETRIEVAL VIA ACOUSTIC INDEXES

An audio message retrieval system functions much like a conventional text retrieval system in that a specific user request is used to locate promising audio documents.[1] In operation, a user enters a text request for information consisting of one or more search keys. Common function words (such as "and," "a," "the") having little information content are removed and the remaining words are reduced to

[1]This type of audio document retrieval is rather different from much of the related research on audio "topic" identification, where much broader subject classification is typical and the classes are predefined.

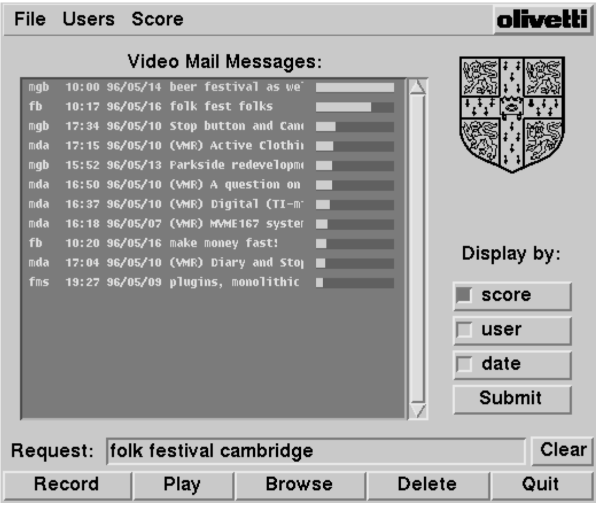

Figure 3. Video Mail User Interface application

stems using a standard algorithm [6] in order to remove word variations that inhibit matching. Once processed, a request is referred to as a search *query* and the stemmed words which it contains are called *terms*.

Given a query consisting of a set of terms, the score for each message is based on the frequency of term occurrence. However, in text retrieval, it has been shown that weighting term occurrences according to the global statistics of term distribution can lead to improved performance. We have investigated the behaviour, for the speech case, of both unweighted and two different types of weighted matching; the results confirmed that, as with text, best performance is obtained with the following *combined weight* scheme:

$$w(i,j) = \frac{c(i) \times f(i,j) \times (K+1)}{K \times l(j) + f(i,j)} \quad (1)$$

where $w(i,j)$ represents the weight of term i in message j. $f(i,j)$ is the number of occurrences of term i in message j and $l(j)$ is the normalised message length [7]. The *collection frequency weight* $c(i) = \log N/n[i]$ where N is the total number of messages and $n[i]$ is the number of messages that contain term i. The main ideas of this weighting scheme are that terms will occur frequently in relevant messages, terms which occur in a small number of messages should be favoured since they will be more discriminating, and messages should be normalised for length since long messages will naturally have more term occurrences independently of their relevance. The combined weight constant K must be determined empirically; and we use $K = 1$.

Given the above weighting scheme, the score for each message j in the corpus is computed by summing over all the term weights and messages are then ranked in score order.

4. USER INTERFACE

The primary user interface consists of a window into which search queries are typed and ranked lists of retrieved messages are displayed. Figure 3 shows an example of this where the ranked list of messages is the result of the query "folk festival cambridge." The bars to the right of the messages graphically indicate the relative score of each message.

After the ranked list of messages is displayed, the user can invoke the video browser shown in Figure 4. This represents the time evolution of the current video mail message by a static horizontal time-line onto which keyword occurrences have been superimposed. In the browser shown, the time-line is the black bar and the scale indicates time in seconds. During playback, or when pointed at with the mouse, a word hit is highlighted and its name is displayed. (In the Figure, the word "festival" has just been played.) The message may be played starting at any time simply by clicking at the desired time in the time bar; this lets the user selectively play regions of interest, rather than the entire message.

Figure 4. Prototype mail browser

5. EXPERIMENTAL EVALUATION

The performance of the VMR system was tested on the VMR message corpus which is a structured collection of audio training data and information–bearing audio messages[4]. The latter consists of 5 hours of spontaneous messages from 15 speakers with topics chosen to be rich in occurrences of an associated fixed set of 35 keywords. All messages are orthographically transcribed, hence it is possible to compare audio-based retrieval with conventional text-based retrieval.

Table 1 summarises the performance of the VMR system in terms of Average Retrieval Precision operating under a number of contrasting conditions. The first two rows give the retrieval performance obtained using the text transcriptions rather than the audio, and the next two rows show the performance using conventional keyword spotting with speaker dependent (SD) whole-word models and speaker independent (SI) sub-word models. The remaining rows give results for phone-lattice based spotting using both SD and SI models. The final row corresponds to the set-up used in the operational VMR system. The column "Abs" shows the absolute Average Precision achieved using the standard VMR1 collection of queries and relevance sets[4], and the column "Rel" presents the same information relative to the scores obtained using the text transcriptions rather than the audio. Where appropriate, the Figure of Merit(FOM)

Indexing Method	#key words	Precision		FOM
		Abs.	Rel.	
Text	All	0.72	100.0%	—
Text	35	0.36	49.9%	—
SD Whole-word	35	0.32	44.0%	81.5%
SI Subword	35	0.30	41.8%	69.9%
SD Phone Lattice	35	0.30	42.1%	73.6%
SI Phone Lattice	35	0.25	34.4%	60.4%
SD Phone Lattice	All	0.50	68.9%	—
SI Phone Lattice†	All	0.47	65.5%	—

Table 1. Relative retrieval performance (Average Precision) and Figure of Merit (FOM) for various indexing methods on the VMR Message Corpus. † denotes indexing method used in operational VMR system.

scores for the word spotting are also shown.

As can be seen from the FOM scores, the SD models perform significantly better than the SI models and phone-lattice based word spotting is a little worse than conventional keyword spotting. These results also show that for a given set of search terms, average retrieval precision for the various model sets follows the same general trend as the FOM scores.

However, the comparison between using the fixed keyword set and the open set shows that artificially limiting the number of terms in a query severely degrades performance. As shown by the second line of Table 1, using text transcriptions, retrieval performance is halved by restricting the available search terms to the 35 keywords and as shown in the lower part of the table, the same effect is observed for the speech-based retrieval. (This would be even more dramatic if the message collection had not been designed in a way that makes the 35 keywords useful search keys for it.) Hence, even though the phone lattice spotting is less accurate than the fixed-keyword spotting, it substantially outperforms the fixed-keyword retrieval because of the availability of additional search terms.

6. CONCLUSIONS AND FUTURE WORK

This paper has presented an overview of the Video Mail Retrieval system which represents a first step towards open-vocabulary audio indexing for multimedia document retrieval. Phone lattices have been shown to provide a good intermediate representation of the words in the audio soundtrack. They can be precomputed, they allow genuinely open-keyword spotting and, when combined with a term-weighted IR metric, they give robust document retrieval.

However, significant challenges remain, especially in terms of scalability to much larger archives. Although phone lattices can be searched quickly, they still require significant search effort. In the future, we expect to combine phone lattice based spotting with large vocabulary transcription and our preliminary experiments in this area have shown that this combination can give better retrieval performance than either technique alone [8].

In a similar fashion to the CMU Informedia project [9], we have also developed a version of the VMR system which uses close-caption text transmissions to index broadcast news material [10]. This allows a user to browse rapidly through several months of broadcast news looking for items of interest. We believe that this prototype is a compelling demonstration of the benefits to the user of access to news items "on demand". We now plan to extend this system to incorporate speech-based indexing and although the use of speech is inferior to that of text, the results presented here suggest that it will nevertheless yield adequate retrieval performance.

ACKNOWLEDGEMENTS

The VMR project was supported by the UK DTI Grant IED4/1/5804 and SERC (now EPSRC) Grant GR/H87629 under the SALT programme. Olivetti Research Limited is an industrial partner in the VMR project.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.