Automatic Content-Based Retrieval of Broadcast News

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

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

Appears in Proceedings of ACM Multimedia 95, San Francisco, November 1995

Abstract

This paper presents current work on a video retrieval project at Cambridge University and Olivetti Research Limited (ORL). We show that statistical methods developed for text retrieval are also effective for retrieving and browsing multimedia documents. These methods allow rapid retrieval of news broadcasts by information content determined from teletext subtitles. Information retrieval results for experiments performed on a large archive of news broadcasts are presented. This is made possible by the ORL Medusa system, which allows practical recording, storage, and playback of tens of gigabytes of multimedia data. This work is a step towards practical retrieval of multimedia documents, where the information content is determined from speech recognition performed on the audio soundtrack. We describe the project background, the ORL Medusa multimedia system, and retrieval application, as well as the news broadcast corpus and methods of browsing the retrieved news stories.

Keywords

Multimedia, ATM, content-based retrieval, television news, information retrieval, text subtitles, browsing

1. 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 [1, 16]. 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.

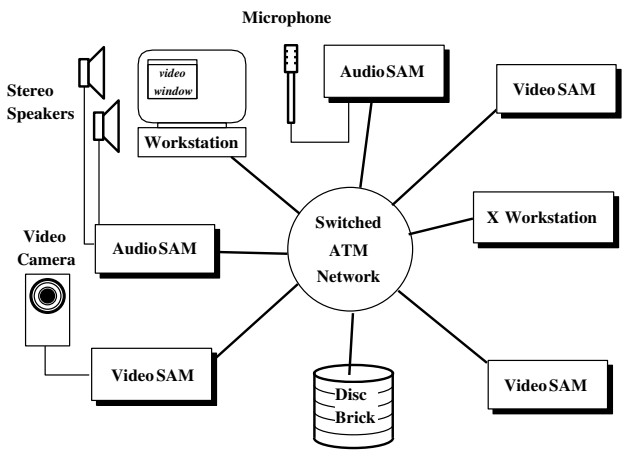

Figure 1. Connection of Medusa Modules to an ATM network.

Previous work on the VMR project demonstrated practical retrieval of audio messages using speech recognition for content identification [8, 4]. Because of the limited number of available audio messages, a much larger archive of television news broadcasts (along with accompanying subtitle transcriptions) is currently being collected. This will serve as a testbed for new methods of storing and accessing large amounts of audio/video data. The enormous potential size of the news broadcast archive dramatically illustrates the need for ways of automatically finding and retrieving information from the archive. Quantitative experiments demonstrate that Information Retrieval (IR) methods developed for searching text archives can accurately retrieve multimedia data, given suitable subtitle transcriptions. In addition, the same techniques can be used to rapidly locate interesting areas within an individual news broadcast.

Although large multimedia archives will be more common in the future, today they require a specialised and high-performance hardware infrastructure. The work presented here relies on the the Medusa system developed at ORL, which includes distributed, high-capacity multimedia repositories. This paper begins with an overview of the ORL Medusa technology. Subsequent sections describe the collection and storage of a BBC television broadcast news archive, a retrieval methodology for location of potentially relevant sections in response to users' requests, and a graphical user interface for content-based retrieval and browsing of news

Figure 2. The Medusa multimedia system.

broadcasts.

2. MEDUSA: MULTIMEDIA ON AN ATM NETWORK

The Medusa Project at ORL is a novel and extensive experiment in sending many streams of digital audio and video over a prototype 100 Megabit-per-second switched ATM (Asynchronous Transfer Mode) network [6, 9]. Some 200 network connections cover all laboratory rooms; optical fibre extends the 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 appropriate medium for multimedia applications.

Multimedia input and output capability 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 units. The Medusa hardware is constructed in a modular fashion by adding specialised option cards to a general-purpose ATM network interface called an ATMos card. The ATMos card has an ARM processor and up to 32 Mbytes of memory. A wide 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 1 shows connection of various SAMs to the network.

Figure 2 shows a typical Medusa installation, including cameras and microphones. To the far right of the figure is a Disc Brick repository; an audio SAM sits between this Disc Brick and the room's 8-port ATM switch. On top of the switch are two video SAMs. Four microphones and four speakers are connected to the audio SAM. Each video SAM is connected to two camera heads (one can be seen sitting on top of the workstation display). Between the switch and the workstation screen is a VideoTile, a large high-quality colour LCD display connected directly to the ATM network.

An Audio SAM supports a set of high-fidelity microphones and speakers. It has two high-quality programmable CODECs providing four independent audio inputs and four independent audio outputs, each sampling at rates of up to 48 kHz, with 16 bits of precision. The Video SAM supports one or two colour cameras (or other video sources). The Disc Brick SAM uses the ATMos card to connect a RAID-3 array of discs to the network. A microkernel called ATMos runs on the ARM processor of each SAM. It is capable of the rapid task-switching required to receive and transmit ATM cells with low latency. ORL has also developed a TURBOchannel ATM interface card so that multimedia data can be processed on workstations and displayed in X windows at 25 frames per second. The network is constructed from ORL 8-port ATM switches, permitting many ATM connections to be made to each room.

2.1. The Medusa software environment

The Medusa software applications environment is a peer to peer architecture for controlling networked multimedia devices and routing streams between them [22]. A Medusa server process runs on each of the SAMs as well as on ATM networked workstations. Multimedia data flows through a series of Medusa software modules on these servers and across the network between servers. Medusa module code makes it simple to set up and dynamically modify media connections between software modules connected in a pipeline. There are three basic classes of Medusa software modules: source modules have stream outputs, sink modules have audio or video stream inputs, and pipeline modules which have both inputs and outputs. Medusa source and sink modules communicate with their SAM's ATMos device driver process, which in turn communicates audio/video data to and from the CODECs.

Pipeline modules can buffer, transform, or modify data in real time. For example, audio sample rate conversion is done with such a pipeline module. A timestamp module appends a capture time to each Medusa data block, required for audio/video playback synchronisation. An especially in-

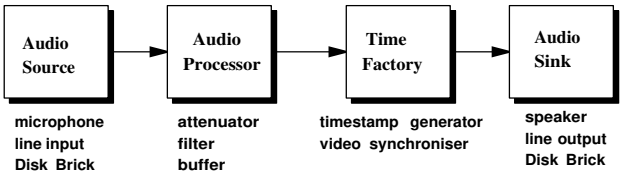

Figure 3. Example of Medusa software "module" connection.

teresting kind of module is an analyser. Examples include a reporter of audio loudness or a speech recogniser. A video analyser module reports video activity levels by calculating a measure of frame differences.

Medusa code allows location-transparent connections to be made between these sources, sinks, modifiers and analysers of multimedia data. The choice of which 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 from a source module which reads a file on a Disc Brick SAM to an audio sink module on an audio SAM via a format convertor running on either machine (Figure 3), or perhaps on a more powerful processor available elsewhere on the network.

Medusa application programs (including the user interface described in Section 5.) are usually written in Tcl/Tk, a scripting language with useful built-in X widgets [11]. Tcl has been extended at ORL to include commands for initialising Medusa software modules, making connections between them, and reading and writing controlling information to the modules. For example, an audio source module's sampling rate and quantisation accuracy may be changed on the fly.

Several large Tcl applications have been written to demonstrate the multi-stream capabilities of Medusa. Each Medusa station is equipped with four cameras, four microphones and four speakers. The Medusa phone application shows small images from all the cameras at each end of the conversation, as well as a single large image selected from the available streams. Streams can be selected manually or automatically through agent software which monitors audio and video activity.

Another Medusa application, called MDmedia (Figure 4), gives simultaneous small views of four TV channels plus a selectable large view of one of them. The video can be viewed on either a workstation or an Olivetti VideoTile, while sound is routed to the office audio SAM. The live media streams are provided by four Medusa camera SAMs and four Medusa audio SAMs which deliver four video sources, each with high quality stereo sound, plus four additional stereo audio channels from compact disc players and FM radio receivers. The news broadcast archive of Section 3.1. is made by routing the BBC1 audio and video streams to a Disc Brick.

2.2. The Multimedia Repository

The Disc Brick, which is the size and shape of a small vertical format PC case, uses the ATMos network interface card plus a SCSI interface to make a RAID-3 array of discs available as a multimedia file server. The prototypes use five 2 Gbyte drives (now being extended to 4 Gbyte drives) giving a storage capacity of 16 Gbytes per unit. Four Disc Bricks are currently deployed on ORL's ATM network. When these

Figure 4. Medusa simultaneous media application.

are upgraded to 4 Gbyte drives they will deliver a total networked storage capacity of 64 Gbytes.

Medusa ATM video cameras 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. Together with the 0.5 Megabits per second required for uncompressed 16-bit audio sampled at 32 KHz, a typical half hour news broadcast amounts to about 2 Gbytes of data. Storing the video at a lower frame resolution of 88 x 64 reduces this to 0.6 Gbytes.

Using more sophisticated compression techniques (such as MPEG encoding) can reduce this figure dramatically, making practical an archive containing several hundred hours of audio/video material. At full resolution (VHS quality), MPEG compression can deliver video and audio at a total rate of 2 Mbps. Using this compression scheme, 72 hours of compressed audio/video could be stored in 64 Gbytes. As archives of these magnitudes become more common, better ways of searching and retrieving information become essential.

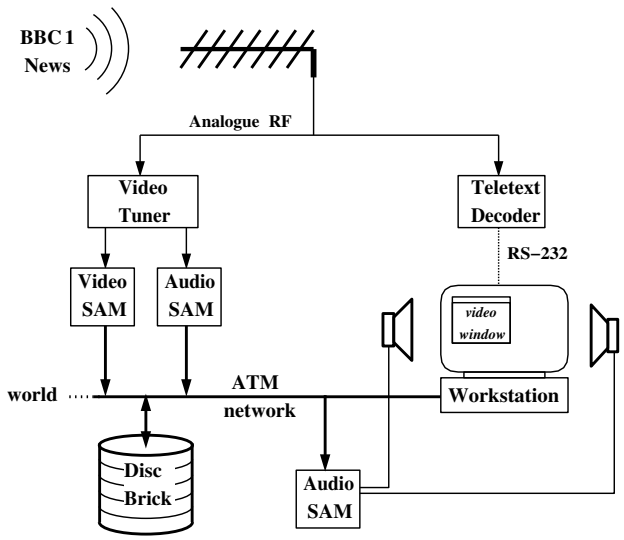

Figure 5. Block diagram of news retrieval system hardware.

3. THE NEWS BROADCAST ARCHIVE

Figure 5 shows a block diagram of the archival system. The archive of news broadcasts is stored on Disc Bricks, for nearly instantaneous retrieval and real-time playback. A daemon automatically records the 9:00 PM BBC1 evening news into the archive. Currently, nearly three months of recordings exist in the archive, with more added nightly. For the IR experiments described in Section 4.1., a one-month subset of the available broadcasts were used.

As the file is recorded, a Medusa module analyses the video stream for movement activity. Large values indicate a camera or scene change; times when this value exceeds a given threshold are recorded for later use. Concurrently, a teletext decoder receives the page 888 subtitle and time information and saves it to disk as well. These text subtitles contain a nearly complete transcription of the words spoken in the broadcast. Text data comes in the natural unit of a line, which is limited in length to the 40 characters (the width of a teletext window). This data is conditioned into an ASCII format suitable for information retrieval by eliminating noise and duplicate lines, and adding the time (accurate to within a few seconds) that each line appeared during the broadcast. Figure 6 shows a short excerpt of a teletext transcription. The transcription of a half hour news broadcast contains from 500 to 800 lines and from 3,000 to 5,000 words. Though the transcriptions are not word-perfect (most contain typos and slight omissions) they reflect the information content of the broadcast very well.

3.1. Long-term broadcast storage

Though the repository size is large, it fills quite rapidly due to the high bandwidth of full-motion video and high-fidelity audio, and thus some form of compression is desirable. To increase the size of the archive, older files can be stored in a compressed or reduced form, both for later retrieval and for speech recognition research.

Offline software MPEG compression can reduce the video data rate by a factor of between 10 and 20, reducing storage requirements to as little as 150 Mbytes per news broadcast. Alternatively, using half size images at a quarter of the frame rate results in an immediate 16x reduction in bandwidth, leaving a video stream that is still useful if not crisp. More drastic compression can be obtained by recording only representative still frames, either at regular intervals or at times indicated by video activity like a new scene.

```
21:02/42 administration is considering air
21:02/42 strikes, without the impediment of
21:02/45 the UN. The White House has been
21:02/47 agonising over what to do in a
21:02/50 series of meetings. Warren
21:02/52 Christopher said that a more
21:02/54 aggressive air campaign is being
21:02/55 considered. But the Americans have
21:02/57 not considered an European request
21:02/58 to commit helicopters to carry in
21:03/01 reinforcements to the safe areas.
```

Figure 6. Example portion of teletext transcription.

The audio stream can be substantially compressed as well. For eventual use in speech recognition experiments, it is desired to retain a high-quality audio signal; even so, decimating the audio stream to 16kHz mono and using a lossless LPC coding technique [14] can still reduce the audio storage requirements to 125 Kbits per second, a reduction factor of 8.

4. NEWS BROADCAST RETRIEVAL

We now describe an innovative broadcast retrieval application that integrates information retrieval techniques and multimedia random-access and browsing capabilities. To retrieve relevant portions of news broadcasts, a user enters a typed *request* consisting of one or more search words (*terms*) or a natural-language sentence. Information retrieval methods are used to locate potentially relevant broadcasts from the associated text subtitles. Once found, similar techniques are used to identify promising areas within a single broadcast. This information is displayed graphically, enabling interesting portions of the video broadcast to be visually located.

4.1. Information Retrieval

Information retrieval (IR) techniques satisfy a user's information need by retrieving potentially relevant documents from an archive. Given a text request and a document (such as broadcast subtitles), a matching score can be computed from the search terms common to both. The higher the score, the greater the document's potential *relevance* to the request. Statistical techniques like those described here have proven their utility on extremely large text corpora, and should continue to be successful on archives that are orders of magnitude larger than those considered here [18].

4.1.1. Match score computation

Before a matching score can be computed, the transcriptions and requests are conditioned as follows:

1. Common non-discriminating words (e.g. "the," "a") from a "stop list" are removed [20].

2. To reduce word form variations that inhibit retrieval matching, hyphens are removed and terms are suffix-stripped to stems using a standard algorithm [12]. Thus "managing," "manager," and "manage" become "manag" (sic).

These are standard practice in current IR systems, and result in *queries* and documents that are simple sequences of term stems.

The simplest match score is the number of terms common to both query and document. This is referred to as *coordination level* or *unweighted* (*uw*) scoring. More sophisticated methods obtain better retrieval performance by weighting the search terms. For example, the *collection frequency weight* (*cfw*) for search term i is,

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

where N is the total number of documents and $n[i]$ is the number of documents that contain term i. Thus, terms occurring in a small number of documents are favoured, as discriminating selectors. For this weighting scheme, the score is the sum of weights for each term which occurs in both the query and the document. More sophisticated scoring methods normalize scores by document length or account for the number of times each term occurs in a document. A recently developed weighting scheme incorporating both these factors is the *combined weight* (*cw*), computed as

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

where $tf(i,j)$ is the frequency of term i in document j, $ndl(j)$ is the length of document j normalised by the average document length, and K is a tuning constant determined empirically [13] .

Once all documents have been scored for a particular query, they may be ranked by score such that the higher-scoring (and hopefully more relevant) documents are found at the top of the list. The ranked list can be used directly in the user interface (as in Section 5.), or to evaluate retrieval performance [20]. The scores themselves may be used to locate interesting portions in a broadcast (as in Section 5.1.).

4.1.2. Broadcast Segmentation

Ideally, retrieval would be performed on individual stories within a news broadcast. But because automatic story-level segmentation is not yet practical (see Section 6.), our approach here is to construct "pseudo-stories" by considering a number of adjacent lines as a *segment* [5]. Scores can be computed for each segment just as for a story. The segment width (number of adjacent lines) may be varied to optimise retrieval performance, and segments may be overlapped to provide finer time granularity. Figure 7 shows an example of overlapping segments. Retrieval experiments using these segments are described in the next section.

4.2. IR Experiments

To properly evaluate retrieval performance requires both a set of requests and a set of relevance assessments for each request. The relevance assessments are a manually-generated

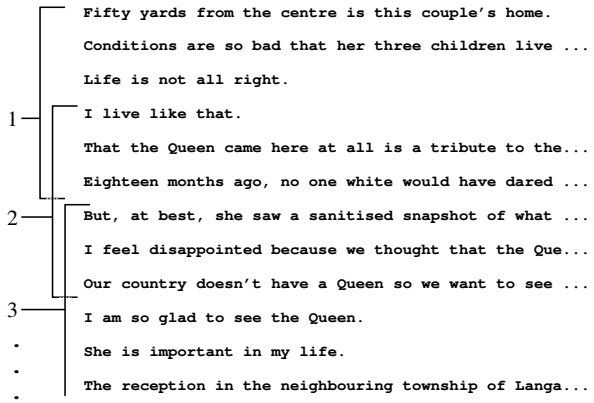

Figure 7. Overlapping retrieval segments.

list showing which documents are relevant to each query. This allows the automatic retrieval performance to be evaluated.

Search Requests: In operation, a user enters a search request expressing their actual information need. For preliminary tests, however, headlines from teletext news pages (quite separate from the broadcast subtitles) were used as requests. Each news page contains a short headline and one sentence about a single news item. These were collected over the same period as the news broadcasts used for these experiments. For example,

```
OVERSEAS TRADE DEFICIT SHOCKS CITY
Britain's balance of payments was
in the red for the first three
months of the year, figures show.
```

Removal of stop words and suffix stripping the remainder produces the following search query,

```
oversea trade deficit shock citi
britain balanc payment
red three
month year figur show
```

Relevance Assessments: To evaluate retrieval performance for each request, the relevance of all archived documents must be determined by manual inspection. Because this is impractical for large archives, a standard practice is to manually assess only a small fraction of the archive. A suitable subset can be formed automatically by pooling the highest scoring documents retrieved by several independent weighting schemes. (This assumes that the different weighting schemes captures all relevant documents.)

A further complication is that the principal retrieval items are segments, rather than actual stories or entire broadcasts. Because obtaining relevance assessments requires expensive human effort, it is not practical to gather separate assessments for all possible segment widths and overlaps. A compromise was to gather relevance assessments for complete news stories, manually extracted from each broadcast. Because news stories are relatively short and homogeneous, it was then assumed that each segment would have the same

	Total terms	Average terms	Relevance Scheme	
			(1)	(2)
width 12	2685	34.1	54	55
width 24	1335	67.0	51	54
width 36	886	99.7	46	52
width 48	659	132.3	39	51

Table 1. Statistics of fixed length segments.

relevance as the story with which it coincides. A fixed rule assigns a relevance to segments that partially overlap a relevant story.

4.2.1. Measuring IR Performance

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 indicator.

4.2.2. Experiment Design

Teletext subtitles for July 1995 news broadcasts were manually segmented at the story level. This resulted in 319 separate story documents. Teletext headline pages were collected from three categories: *UK news*, *UK politics*, and *International news and politics*. Eight headline pages for each category were collected daily. The request set was chosen by selecting at random one page from each category for each day. This produced a total of 78 queries, 26 from each category. The following procedure was used to generate a suitably varied set of requests. Headlines for each category were grouped into pairs separated by approximately two weeks. Groups of 6 headlines were then formed by taking one pair from each category, ensuring that all headlines were separated by at least four days. Finally the ordering of the headlines in each group was permuted using a Latin square technique [19].

A ranked list of all stories was formed for each search query using the *cfw*and *cw*weighting schemes. The top 40 stories from the merged sets form the potential relevance set. For each of the 6 headlines (queries) in a group, a volunteer assessor was asked to assess the relevance of the 40 stories, presented in a randomised order. 13 assessors produced 78 assessment sets, used to judge the effectiveness of the automatic retrieval experiments presented in the next Section.

4.2.3. Experimental Results

This section presents experimental retrieval results for stories and segments of differing widths. Though the story results are not used directly, they serve as a benchmark for segment performance. Examination of the relevance assessments showed that no stories were judged relevant for 19 of the requests. There are two possible reasons for this. Relevant stories may not have been present in the retrieved story pool, or, more likely, there were no relevant stories in the archive because the headline pages typically cover

	Weighting Scheme		
	uw	*cfw*	*cw*
Avg. precision	0.662	0.714	0.821

Average story length = 129.9 terms

Table 2. Retrieval results for manually partitioned stories.

Average precision		Weighting Scheme		
		uw	*cfw*	*cw*
width 12	(1)	0.267	0.298	0.307
	(2)	0.343	0.376	0.391
width 24	(1)	0.326	0.352	0.407
	(2)	0.413	0.450	0.500
width 36	(1)	0.352	0.346	0.401
	(2)	0.432	0.454	0.538
width 48	(1)	0.280	0.303	0.298
	(2)	0.421	0.452	0.532

Table 3. Retrieval results for fixed size segments.

many more stories than the daily 30-minute news broadcasts. These 19 requests were excluded from the experimental request set leaving a total of 59 requests. As before, the total number of stories in the archive was 319. In all the following experiments the average query length was 16.1 terms.

Table 2 gives retrieval results for manually segmented news stories. These results indicate that the IR scheme is able to retrieve stories quite successfully. The relatively high average precision means that large majority of high-ranking stories were genuinely relevant. It can also be observed that the more elaborate term weighting schemes improve retrieval performance, as found in previous experiments on a different corpus [17].

When using segments for retrieval, an important consideration is the appropriate choice of segment width and overlap. Experiments were performed over a range of segment widths to find the most useful width. Examination of the manually-segmented stories showed the average story length to be 47 lines, though some stories are significantly longer than this (several hundred lines) and many are much shorter (three or four lines). The best segment width is likely to be a trade off between segments that are too small to contain sufficient search terms, and those too large to adequately capture short stories. To investigate this, teletext transcriptions were divided into segments of width 12, 24, 36 and 48 lines. The appropriate amount of segment overlap is not obvious, so a 50% overlap was used, as in [2]. The first segment starts at the beginning of the news broadcast, with the next segments following at intervals of half the segment width. Overlapping segments by this amount ensures that any semantically coherent group of lines less than half the segment width will be included intact in at least one segment.

To assess segment retrieval, a means must be found to estimate segment relevances. As discussed earlier, this must be generated automatically from the story relevance assessments. Two relevance assignment rules were considered in this investigation, one using a more stringent and a the other a looser interpretation of relevance assignment:

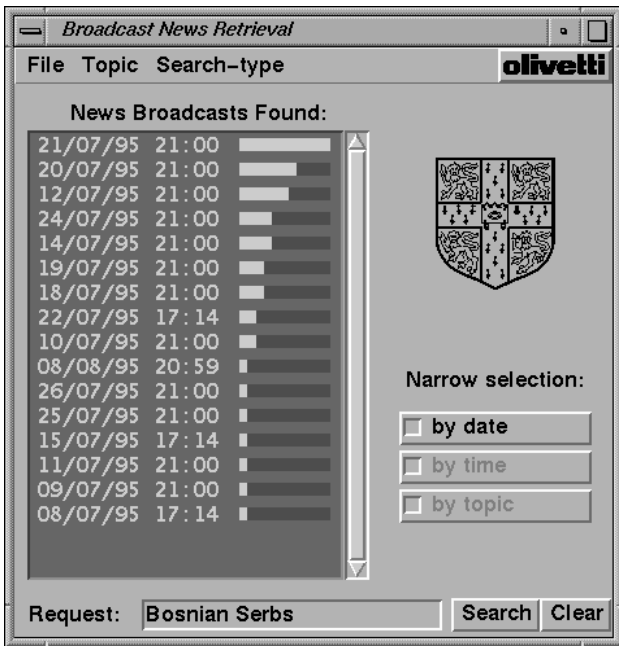

Figure 8. News retrieval application.

(1) A segment is judged relevant to a query only if more than 50% of it is part of a single relevant story. In this case the relevance of a segment is associated with at most one story.

(2) A segment is judged relevant if it meets criterion (1) and additionally if it partially overlaps a relevant story. In this case the relevance of a segment may be associated with at most two stories.

In both cases, stories of less than half the segment length contained completely within a single segment are always judged not relevant. It is thus possible that some queries for which there are one or more relevant stories may have no relevant segments. This relevance assignment relies on the assumption of story homogeneity, meaning that the distribution of matching terms is roughly equal across a story.

4.2.4. Discussion

Table 1 presents statistics for the fixed length segments. The last two columns displays the total number of queries for which at least one segment is relevant. It can be seen that smaller segments result in more queries, because making the segment width smaller will result in more segments that meet the relevance assignment criteria.

Table 3 gives segment-level retrieval results. Although segment retrieval is less precise than manually-partitioned story retrieval, it is still quite effective and improved by the standard term weighting schemes. It is likely that the poorer segment-level results are due to inhomogeneous term distributions across stories. The best segment size for this task is somewhere between 24 and 36 lines depending on the IR method used; a width of 24 was used for the applications of Section 5.. As can be expected from the more relaxed relevance criterion, average precision values are higher for the type (2) relevance assignments.

Figure 9. Video Browser application.

5. THE NEWS BROADCAST RETRIEVAL USER INTERFACE

The broadcast news user interface is displayed in Figure 8. In operation, the user types a search request, and the resulting score for each broadcast is computed as the maximum of the segment scores within the broadcast, as in Section 4.1.. The interface then displays a list of the broadcasts ranked by score, with the scores shown as bar graphs; broadcasts with identical scores are ranked by date. (A similar approach has been used for the retrieval of text articles from a newspaper archive [15].) In its simplest form, the search resembles a "video grep" that returns a list of segments containing a particular search term. However the statistical IR methods used here are much more powerful than a simple keyword search (as used in [10]); multiple-term or natural language queries are not only possible but desirable as they will provide better retrieval performance.

Given a ranked list of broadcasts, the user must still investigate the high-ranking broadcasts to either find the relevant one(s) or to determine that the search was ineffective and that a new search is required. This is a non-trivial problem as it is not simple to determine whether a 30-minute news broadcast contains desired information without watching it in its entirety. While there are convenient methods for the graphical browsing of text, eg scroll bars, "page-forward" commands, and word-search functions, existing video playback interfaces almost universally adopt the "tape recorder" metaphor. To scan an entire broadcast, it must be auditioned from start to finish to ensure that no parts are missed. Even if there is a "fast forward" button, it is generally a hit-or-miss operation to find a desired section in a lengthy broadcast. We find that a far better method is to locate interesting portions of a broadcast from the segment relevance scores.

5.1. A Video Browser

The browser is an attempt to represent a dynamic time-varying process (the audio/video stream) by a static display that can be taken in at a glance [3]. A broadcast 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 broadcast; for example, events at the beginning appear on the left side of the bar and short-duration events are short. Figure 9 shows a news browser exploiting this technique. Segments are represented along the black "timebar" as brighter-coloured rectangles. The brightness of a segment region is proportional to the IR score for the query, so that higher-scoring segments appear brighter and stand out. The entire broadcast may thus be visually scanned for high-scoring segments without any actual playback. The figure shows the results for a query concerning Bosnia; as is evident from the timebar, the most relevant part of the broadcast is near the beginning (the slight relevance indication at the very right of the timebar is due to the recap of major headlines at the end of the broadcast). Portions of the broadcast can be selected for playback by dragging over part of the bar; this lets the user selectively play regions of interest rather than the entire message. In the figure, a short portion about seven minutes into the broadcast is selected for playback. The subtitle transcriptions are displayed in a scrolling list at the bottom. The visible part of this list follows the selected area of the timebar, such that the subtitles corresponding to the beginning of the currently-selected region are displayed, as shown in the figure. In addition, the user may scroll through the subtitles independently, and double-click on any subtitle line to begin video playback from that point.

6. FUTURE WORK AND CONCLUSIONS

The work presented here, including the news broadcast archive, is a necessary first step towards spoken document retrieval. Our experience with the present text data retrieval gives us important information about the nature of the news discourse (for example the optimal segment width) that would be difficult to determine otherwise.

Previous work, by the VMR group and by others, has shown that spoken document retrieval using speech recognition is becoming practical [8, 7]. Thus the work presented here is only the first step towards general audio and video retrieval; future work in this project will use large-vocabulary speech recognition to determine document content. The teletext transcriptions not only serve as a useful benchmark with which to compare future purely speech-based retrieval efforts, but should also be extremely valuable for training speech recognition models (which are highly data-intensive).

Even with the available transcriptions, there are some interesting problems that remain to be solved. If the news broadcasts could be segmented into thematically coherent stories, retrieval and browsing would be greatly simplified. There is much information that can be used to hypothesize story boundaries; for example analysis of video activity [23], statistical analysis of term frequencies in the transcription [5], and even talker identification methods to indicate when a new talker is speaking [21].

Although not used here, it is planned to add the video activity information to the browser, such that a "thumbnail" image is displayed on the timebar for every large activity peak (corresponding to a new scene or image). In addition, areas of moderate activity (indicating camera or subject motion) can also be highlighted, allowing them to be discriminated from the more static images of the news presenters.

As the technology to capture and store multimedia becomes increasingly widespread, multimedia archives will become larger and less manageable. The techniques presented in this paper address practical content-based retrieval and browsing of large amounts of multimedia information. Our experience with retrieving from a large multimedia archive shows that text-based methods are not only successful at locating documents, but are also a powerful tool for locating information within a document. Additional work on automatic transcription generation remove the need for subtitles or other manually-generated annotation, meaning nearly any spoken data may be retrieved by its content.

7. ACKNOWLEDGEMENTS

Broadcast images and text ©BBC 1995 and have been reproduced by Cambridge University with permission. Olivetti Research Limited is an industrial partner of the VMR project. This project is supported by the UK DTI Grant IED4/1/5804 and SERC (now EPSRC) Grant GR/H87629.

REFERENCES

[1] 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.

[2] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using SMART : TREC 3. In D.K. Harman, editor, *Proceedings of TREC 3*, pages 69–80. National Institute of Standards and Technology, Gaithersburg MD,, 1995.

[3] G. Cruz and R. Hill. Capturing and playing multimedia events with STREAMS. In *Proceedings ACM Multimedia 94*, pages 193–200, San Francisco, October 1994. ACM.

[4] 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.

[5] M. Hearst and C. Plant. Subtopic structuring for full-length document analysis. In *Proc. SIGIR 93*, pages 59–68, Pittsburgh, 1993. ACM Press.

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

[7] D. A. James. *The Application of Classical Information Retrieval Techniques to Spoken Documents*. PhD thesis, Cambridge University, February 1995.

[8] 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.

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

[10] C. J. Lindblad, D. J. Wetherall, and D. L. Tennenhouse. The VuSystem: A programming system for visual processing of digital video. In *Proceedings ACM Multimedia 94*, pages 307–314, San Francisco, October 1994. ACM.

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

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

[13] S. E. Robertson and K. Spärck Jones. Simple, proven approaches to text retrieval. Technical report, Cambridge University Computer Laboratory, Dec 1994.

[14] Tony Robinson. SHORTEN: Simple lossless and near-lossless waveform compression. Technical Report TR156, Cambridge University Engineering Department, December 1994.

[15] M. Sanderson and K. van Rijsbergen. NRT (News Retrieval Tool). *EP-ODD, Electronic Publishing, Origination, Dissemination & Design*, 1994.

[16] 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.

[17] K. Spärck Jones, G. J. F. Jones, J. T. Foote, and S. J. Young. Retrieving spoken documents: VMR project experiments. Technical Report 366, Cambridge University Computer Laboratory, May 1995.

[18] Karen Spärck Jones. Reflections on TREC. Technical Report 347, Cambridge University Computer Laboratory, August 1994.

[19] J. M. Tague. The pragmatics of information retrieval experimentation. In K. Spärck Jones, editor, *Information Retrieval Experiment*, chapter 5, pages 59–102. Butterworths, 1981.

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

[21] 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.

[22] 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.

[23] H. J. Zhang, S. Y. Tan, S. Smoliar, and G. Yihong. Automatic parsing and indexing of news video. *Multimedia Systems*, 2:256–266, 1995.