# Audio Indexing and Retrieval of Complete Broadcast News Shows

**S.E. Johnson†, P. Jourlin‡, K. Spärck Jones‡ & P.C. Woodland†**

†Cambridge University Engineering Department,
Trumpington Street,
Cambridge, CB2 1PZ, UK.
{sej28,pcw}@eng.cam.ac.uk

‡Cambridge University Computer Laboratory,
Pembroke Street,
Cambridge, CB2 3QG, UK.
{pj207,ksj}@cl.cam.ac.uk

**Abstract**

This paper describes a system for retrieving relevant portions of complete broadcast news shows starting with only the audio data. A novel system of automatically detecting and removing commercials is described and shown to increase the performance of the system whilst also reducing the computational effort required. The sophisticated large vocabulary speech recogniser which produces the high-quality transcriptions and the window-based retrieval system with post-merging are also described.

Results are presented using the 1999 TREC-8 Spoken Document Retrieval data for the task where no story boundaries are known. Experiments investigating the effectiveness of all aspects of the system are described and the relative benefits of automatically eliminating commercials, enforcing broadcast structure during retrieval, using relevance feedback, changing retrieval parameters and merging during post-processing are shown. An Average Precision of 46.5%, when duplicates are scored as irrelevant is shown to be achievable using this system.

## 1 Introduction

With the ever increasing amount of information being stored in audio and video formats, it is necessary to develop efficient methods for accurately extracting relevant information from these media with little or no manual intervention. This is particularly important for the case of broadcast news since the density of important up-to-date information is generally high, but topic changes occur frequently and information on a given event will be scattered throughout the broadcasts.

Initially work done in Spoken Document Retrieval (SDR) focused on the automatic transcription of American broadcast news audio into manually pre-defined "stories", which were then run through a text-based retrieval engine (Garofolo et al., 1998; Garofolo et al., 1999). However, manually-generating story boundaries is a time consuming task and is not feasible for large, constantly updated collections.

Some recent work has therefore focused on retrieving information automatically when no manual labels for story boundaries exist. There are two main techniques used for this type of task. The first involves creating quasi-stories by using a simple windowing function across automatically generated transcriptions and then running some window recombination after retrieval (e.g. (Abberley et al., 2000; Dharanipragada and Roukos, 1997; Dharanipragada et al., 1999; Robinson et al., 1999)). The second involves attempting to find structure within the broadcast automatically, for example with story segmentation, or detection of commercials. This generally involves generating a transcription and performing the segmentation using text-based methods (e.g. (van Mulbregt et al., 1999)), but it is also possible to use additional audio or video cues (e.g. (Hauptmann and Witbrock, 1998)). This paper describes experiments on a system which uses both ideas, exploiting properties of the audio to impose some structure on complete broadcasts, whilst using windowing techniques to find relevant passages during retrieval.

Section 2 describes the framework for the experiments reported in this paper including the data set used and the method of performance evaluation. Section 3 describes a method for automatically detecting and eliminating commercials using the audio data directly, whilst the overall recognition, indexing and retrieval system is described in section 4. More details about the experimental procedure and a discussion of the scoring measures are given in section 5. Experimental results showing the effect of commercial removal, enforcing structure within the broadcasts and improving the retrieval and post-processing are given in section 6 and finally conclusions are offered in section 7.

## 2 Description of Task and Data

The experiments reported in this paper use the framework of the TREC-8 Spoken Document Retrieval (SDR) Story Unknown (SU) track (Garofolo et al., 2000). For this evaluation 500 hours of American broadcast news audio were supplied along with 50 queries and their associated human relevance assessments. The manually-generated story boundaries which had been used when producing these assessments were then used to define the official story-IDs for the scoring procedure.

Participants had to automatically produce show:time stamps for each query for the portions of audio thought to be relevant to that query. These were then mapped to the appropriate story-ID and all but the first occurrence of each story was labelled as irrelevant. Any non-story audio, such as commercials or jingles was also scored as irrelevant before the standard IR measures of *precision* (proportion of retrieved documents that are relevant) and *recall* (proportion of relevant documents that are retrieved) were calculated. The overall performance measures reported in this paper is the Average Precision (averaged over precision values computed after each relevant document is retrieved) and R-precision (precision when the number of documents retrieved equals the number of relevant documents) averaged over all the queries.[1]

The data used for the evaluation was the February 1998 to June 1998 subset of the audio from the TDT-2 corpus. It consisted of 244 hours of Cable News Network (CNN) broadcasts, 102 hours from Voice of America (VOA), 93 hours from Public Radio International (PRI) and 62 hours from the American Broadcasting Company (ABC). All recognition had to be performed *on-line*, namely not using any material broadcast after the date of the show being processed, whilst retrieval was *retrospective* i.e. any data up until the end of the collection (June 30th 1998) could be used. The use of any manually-derived story boundary information was prohibited in both tasks.

## 3 Automatic Elimination of Commercials

The overall system is designed to pick out areas in the news broadcasts which are relevant to a users request. Complete audio shows include portions other than news stories, such as commercials, which contain little content information and hence are very unlikely to be relevant to any user request. Therefore any method of automatically removing such portions of audio would not only reduce the amount of computational time needed for recognition, but would also reduce the possibility of false matches occurring, and hence increase overall retrieval performance.

A system was built to automatically detect and remove commercials within the framework of the TREC-8 SU task. The commercial detector was based on finding segments of repeated audio using a method for direct audio search (Johnson and Woodland, 2000) making the assumption that (usually) only commercials are repeated.[2] The detector used a windowing system to divide the audio into overlapping segments of 5 seconds long with a shift of 1 second between adjacent windows. Each

[1] 1 of the 50 queries was adjudged to have no relevant documents within the TREC-8 corpus and therefore was not used in the calculation of AveP and R-precision.

[2] Tony Robinson initially suggested the idea that repeated audio could indicate the presence of commercials.

window was characterised by the covariance matrix of the (wideband) PLP cepstral coefficients as used in the subsequent speech recognition process. The windows were compared to a library of windows stored from previous shows from the broadcaster (the broadcast history)[3] using a direct match based on the arithmetic harmonic sphericity distance (Bimbot and Mathan, 1993) between the windows.

Safeguards were introduced to reduce the probability of stories being wrongly discarded, either due to false matches or to the story itself being rebroadcast by playing the same audio track during different news bulletins. These included forcing the match to occur a minimum number of times and in more than one preceding show and introducing a delay between the current show and the broadcast history.

Smoothing was then carried out to relabel sections of audio between matches as commercials, conditional on the resulting commercial being less than a maximum allowable length and, for the case of the CNN shows, fitting within a show grammar. Finally the boundaries of the postulated commercials were refined to take into account the courseness of the initial windows. This process is illustrated in Figure 1 and more details can be found in (Johnson et al., 2000).

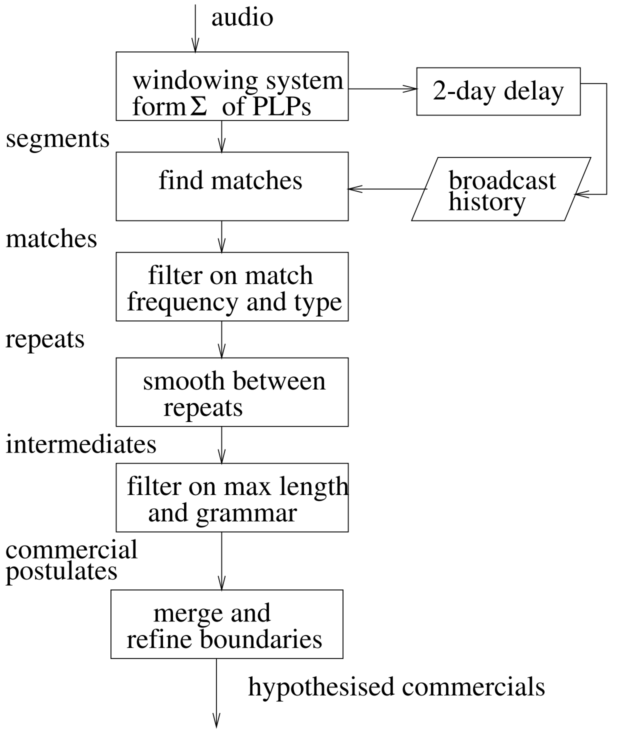


**Figure 1:** The Commercial Detection Process

Since the audio was eliminated at an early stage and could not be recovered later during processing, a very conservative system, `C-1`, which removed 8.4% of the audio, was used for the TREC-8 SDR evaluation. A contrast run, `C-2`, which removed 12.6% of the audio, was later made to see the effect of relaxing the tight constraints on the system. The breakdown of data removed using these systems compared to the manually-generated story labels is given in Table 1. Note that these "reference" labels are not an exact reflection of the story/commercial distinction, since a few commercials have been erroneously labelled as stories and some portions of actual news have not had story labels added and hence are wrongly scored as commercials; however they offer a reasonable indicator of the performance of the commercial detector.

[3] In theory all the data in the test collection could be used (in an unsupervised way) for the library, but this was not allowed within the TREC-8 SU evaluation framework, as recognition was an *on-line* task.

| | Broadcaster | Non-Stories | Stories | Total |
|---|---|---|---|---|
| C 1 | ABC | 12.8hrs=65.5% | 28s=0.02% | 12.8hrs=20.48% |
| | CNN | 26.2hrs=35.7% | 2822s=0.46% | 27.0hrs=11.03% |
| | PRI | 1.9hrs=16.6% | 297s=0.10% | 2.0hrs= 2.16% |
| | VOA | 0.5hrs= 5.0% | 132s=0.04% | 0.5hrs= 0.49% |
| | ALL | 41.4hrs=36.3% | 0.9hrs=0.23% | 42.3hrs= 8.42% |
| C 2 | ABC | 13.8hrs=70.6% | 107s=0.07% | 13.8hrs=22.12% |
| | CNN | 43.3hrs=59.0% | 10640s=1.73% | 46.2hrs=18.91% |
| | PRI | 2.6hrs=22.4% | 416s=0.14% | 2.7hrs= 2.92% |
| | VOA | 0.6hrs= 6.0% | 208s=0.06% | 0.6hrs= 0.58% |
| | ALL | 60.2hrs=52.9% | 3.2hrs=0.81% | 63.4hrs=12.63% |

**Table 1:** Amount of data rejected during Commercial Elimination

The results in Table 1 show that automatic commercial elimination can be performed very successfully for ABC news shows. More false rejection of stories occurs with CNN data, due to the frequency of short stories, such as sports reports, occurring between commercials. The amount of commercial rejection with the VOA data is low, due mainly to the absence of any VOA broadcast history from before the test data. However, overall the scheme worked well, since 97.8% of the 42.3 hours of data removed with the `C-1` system (and 95.0% of the 63.4 hours removed by the contrast `C-2` run) were labelled as non-story in the reference.

## 4 The Complete System

A block diagram of our complete system is given in Figure 2. After the commercial detection and elimination stage, the audio is automatically split into segments of between 1 and 30 seconds, and labelled by gender and bandwidth. A further 34 hours of data classified as music and silence was discarded during this segmentation process.

The main transcription system used a continuous mixture density, tied-state cross-word context-dependent HMM system based on the CUHTK-Entropic 1998 Hub4 10xRT system (Odell et al., 1999) and is described in more detail in (Johnson et al., 2000). The data was coded into cepstral coefficients and cepstral mean normalisation was applied. A 2-pass system was implemented, the first pass used gender-independent, bandwidth-specific triphone models with a 60,000 word 4-gram language model. The output from this pass, denoted `HTK-p1`, gave a word error rate (WER) of 26.6% on the 10 hour scored subset of the TREC-8 SDR data.

A second pass used MLLR-adapted gender and bandwidth dependent triphone models with a 108,000 word trigram mixture language model to generate lattices from which a one-best output was made using a 4-gram model. This transcription, denoted `HTK-p2`, gave a WER of 20.5% on the scored subset.

The transcriptions were then divided into windows of length 30 seconds, with a 15 second shift between adjacent windows. Knowledge about the broadcast accumulated from the segmentation and commercial detection phases was incorporated into the windowing system by enforcing boundaries when a gap of over 5 seconds appeared in the transcriptions. Such gaps were thought to indicate the presence of either pure music (such as in a jingle), or commercials and hence offer a reasonable indicator of where a change in story might occur within the broadcast. Finally very short windows (less than a certain duration or number of words) were removed before retrieval.

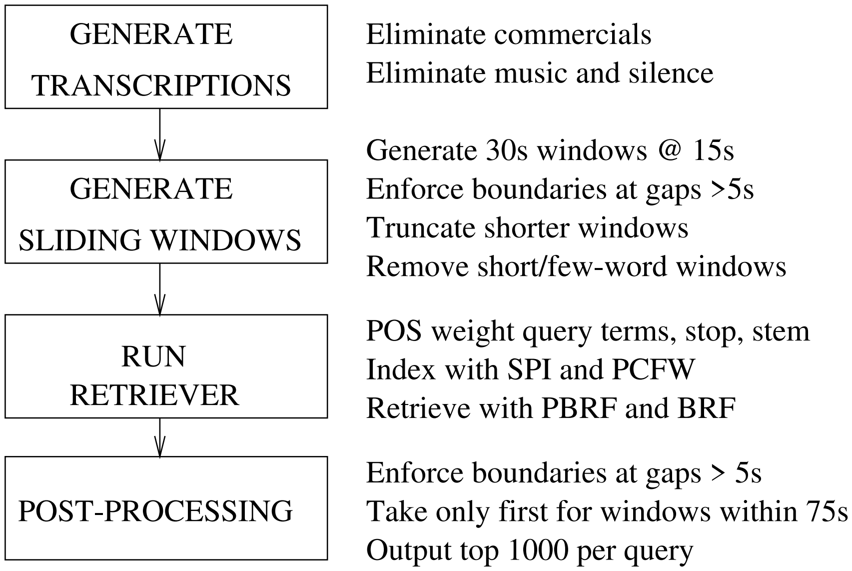


**Figure 2:** The Whole-Show Retrieval System

Our Okapi-based retriever was used with traditional stopping and Porter stemming. Also included was a stemming exceptions list, and part-of-speech weighting for the query terms. Semantic poset indexing (Jourlin et al., 1999a) was used to capture some semantic information about geographical locations and unambiguous nouns, whilst parallel collection frequency weighting (Johnson et al., 2000) was used to obtain more robust estimates of the collection frequency (inverse document frequency) weights. Both traditional and parallel blind relevance feedback were used to add terms to the query during retrieval. A more detailed system description can be found in (Johnson et al., 2000).

Finally a post-processing stage was implemented to try to reduce the number of multiple hits (duplicates) from each story source. When two retrieved windows originated from within a certain time period in the same show, the one with the highest score was retained, whilst the other was discarded. However, the inferred structure of the broadcast was again used, by enforcing hard breaks at gaps of more than 5 seconds in the audio.

## 5 Experiments and Scoring Measures

Experiments were conducted on the 1998 TREC-8 SU test data, which consisted of 21,754 *stories* and 6,294 portions of audio between the stories which were not considered to be information-bearing. These *non-stories* were mainly commercials and filler portions between two stories. From the 50 queries and relevance assessments given, there were a total of 1,818 relevant stories, 1,085,882 non-relevant stories and 314,700 non-stories[4].

The scoring method for the TREC-8 SU evaluation mapped the show:time stamps given in the ranked list produced by the system, to a story-ID. All non-stories were scored as irrelevant and the first occurrence of each *relevant* story was scored as relevant. The difficulty arises when considering how to deal with duplicate hits from the same story. The method used in the evaluation scored all duplicates as irrelevant, irrespective of whether they represented a relevant story or not. Whilst this does reflect a real scenario to some degree, in that a user does not want to be presented with the same story more than once, it is a rather harsh scoring measure, and the reduction of duplicates seems to affect the score considerably more than an increase in the number of relevant documents found.

[4] 6,294 non-stories multiplied by 50 queries.

An alternative suggestion was to remove all duplicates before scoring, but this is also unsatisfactory since it encourages systems to over-generate story hits and produce many duplicates, which the user would not want to see. For example, suppose there were 50 relevant stories for a given query. Since the retriever returns the top 1000 matches, there would be no disincentive to produce 5 (or even 10) matches per story providing that there were not more than another 150 (or 50) non-relevant stories returned by the retriever.

In this paper, we use the official TREC-8 SU scoring procedure and quote the AveP and R-precision when all duplicates are scored as irrelevant. However, we supplement this figure by quoting the *%retrieved* (proportion of the entire set which has been retrieved) of relevant stories, (RS), non-relevant stories (NRS) and non-stories (NS), and give the number of duplicates. These latter measures are especially interesting, since they can be given at any stage of the system and unlike the Average and R-precision, are not influenced by how duplicates are scored.

The system described in section 4 gave an AveP of 41.47% on `HTK-p2` and 41.50% on `HTK-p1`, in the TREC-8 SU evaluation, the R-precision being 41.98% and 41.63% respectively.[5]

# 6 Experimental Results

## 6.1 Effect of Commercial Elimination

A second run of the `HTK-p1` system with *no commercial elimination* was made to allow experiments which investigated the effects of automatically detecting and removing commercials to be conducted.

Two strategies for eliminating the commercials were compared. The first removed the sections of audio corresponding to the automatically labelled commercials before recognition, as in our original system. The second removed any windows returned by the retriever which occurred in a postulated commercial break, before the final post-processing stage, and thus could be applied to any retrieval system on any set of transcriptions.

The results before post-processing from applying no commercial elimination (-), the TREC-8 evaluation system (`C-1`) which removed 8.4% of the data, and the less conservative run (`C-2`) which removed 12.6% of the data are given in Table 2. The `%retrieved` for relevant stories (RS), non-relevant stories (NRS) and non-stories (NS) is given along with the number of duplicates (#Dup), before the final post-processing stage. The effect of removing the commercials before generating the transcriptions (BT) and after retrieving the windows (AR) is shown.

| BT | AR | RS | NRS | NS | #Dup |
|---|---|---|---|---|---|
| - | - | 94.7 | 39.3 | 25.6 | 734,897 |
| - | C-1 | 94.7 | 39.2 | 20.3 | 703,071 |
| - | C-2 | 94.4 | 39.0 | 17.4 | 690,069 |
| **C-1** | - | 94.2 | 39.1 | 18.6 | 697,143 |
| C-1 | C-2 | 93.9 | 38.9 | 16.0 | 686,162 |

**Table 2:** %Retrieved of Relevant Stories (RS), Non-Relevant Stories (NRS), Non-Stories (NS) and number of duplicates (#Dup) before post-processing, when removing commercials before recognition (BT) or after retrieval (AR) on `HTK-p1` transcriptions

[5]The AveP for our complete story-known system for the TREC-8 evaluation was 55.29% on `HTK-p2` and 54.51% on `HTK-p1`.

These results show that the `%retrieved` for non-stories can be greatly reduced by the automatic removal of commercials. When applying the conservative `C-1` system after retrieval, the `%retrieved` for non-stories and the number of duplicates can both be considerably reduced, without affecting the `%retrieved` for relevant stories. Further reductions in the retrieval of irrelevant and duplicate information can be made by using the less conservative `C-2` run or pre-filtering the audio, but at a slight cost to the `%retrieved` for relevant stories. The retrieval results after the post-processing stage are given in Table 3.

| BT | AR | RS | NRS | NS | #Dup | AveP | R-P |
|---|---|---|---|---|---|---|---|
| - | - | 77.5 | 3.52 | 2.50 | 2550 | 41.00 | 40.96 |
| - | C-1 | 78.1 | 3.72 | 1.76 | 2658 | 41.22 | 41.34 |
| - | C-2 | 77.9 | 3.80 | 1.44 | 2720 | 41.13 | 41.50 |
| **C-1** | - | 77.6 | 3.76 | 1.62 | 2667 | $\boxed{41.50}$ | 41.63 |
| C-1 | C-2 | 77.6 | 3.84 | 1.32 | 2730 | 41.42 | $\boxed{41.77}$ |

**Table 3:** %Retrieved, number of duplicates and % Average and R-precision after post-processing when eliminating postulated commercials, on `HTK-p1` transcriptions.

Filtering out windows thought to correspond to commercials after retrieval can be performed using any retriever on any set of transcriptions. For example, the results when applying the technique to the TREC-8 transcriptions from LIMSI (Gauvain et al., 2000), which have a word error rate of 21.5%, are shown in Table 4.

| AR | RS | NRS | NS | #Dup | AveP | R-P |
|---|---|---|---|---|---|---|
| - | 77.0 | 3.48 | 2.61 | 2610 | 40.19 | 41.12 |
| **C-1** | 77.4 | 3.68 | 1.73 | 2710 | $\boxed{40.75}$ | 41.79 |
| C-2 | 77.3 | 3.78 | 1.53 | 2701 | 40.49 | $\boxed{41.94}$ |

**Table 4:** %Retrieved, number of duplicates and % Average and R-precision after post-processing when filtering out postulated commercials, on LIMSI's transcriptions.

These results show that the AveP can be increased by 1.4% relative on the transcriptions from LIMSI and 0.5% relative on the complete `HTK-p1` transcriptions by filtering the windows returned by the retriever using the `C-1` postulated commercial breaks. Both the R-precision and `%retrieved` for relevant stories also increase with a large drop in `%retrieved` for non-stories for this case. Using the `C-2` postulated commercials gave a further increase in R-precision but led to a decrease in the relevant story `%retrieved` and AveP on both sets of transcriptions.

Despite the drop in the `%retrieved` for relevant stories before post-processing when the commercials are eliminated before recognition, the results in Table 3 show that better precision can be obtained when the commercial elimination is performed at the front-end of the system.

Implementing the `C-1` commercial removal system before recognition thus produced a relative increase of 1.2% AveP and 1.6% R-P over the full `HTK-p1` transcriptions whilst also reducing the amount of computational time required by 8.4%.

### 6.2 Enforcing Broadcast Structure

Some automatically derived knowledge of the structure of the broadcast was used during both the window generation and post-processing stages of our system. This was implemented by enforcing hard breaks (such that no window could be generated across such a break during pre-processing, and no merge could take place over such a break during post-processing) whenever a gap of over 5 seconds occurred in the transcriptions. It was felt that such a gap would only be generated during the commercial elimination or segmentation stages and thus would indicate the presence of either a commercial, or pure music such as a jingle.

**6.2.1 Breaks in Post-Processing** An experiment was conducted to observe the effects of altering the length of gap required to enforce such breaks during post-processing using our `HTK-p2` transcriptions[6] and the results for 3, 5 and 10 seconds are given in Table 5.

| Gap | RS | NRS | NS | #Dup | AveP | R-P |
|---|---|---|---|---|---|---|
| 3s | 78.2 | 3.66 | 1.66 | 3587 | 40.81 | 40.92 |
| **5s** | **78.4** | 3.74 | 1.68 | 2707 | $\boxed{41.47}$ | 41.98 |
| 10s | 78.3 | 3.76 | 1.67 | 2504 | 41.44 | 42.01 |
| $\infty$ | 78.3 | 3.77 | 1.67 | 2422 | 41.44 | 42.01 |

**Table 5:** Effect of changing the gap required in the transcriptions to enforce a hard break during post-processing, on `HTK-p2` transcriptions. ($\infty \equiv$ not enforced)

These results show that although many merges have been prevented by enforcing hard breaks at gaps of 5 seconds in the transcriptions, (leading to an increase in the number of duplicates), the overall results are practically unaffected. There is a very slight increase in relevant story `%retrieved`, due to distinct relevant stories which occur across a hard boundary no longer being incorrectly merged. However, some non-stories and non-relevant stories which would have been merged if no hard breaks had been enforced, now remain as separate entities. Since duplicates are scored as irrelevant, this practically counteracts the gain from not merging distinct relevant stories.

**6.2.2 Breaks in Window Generation** The initial windows were generated by moving a 30s sliding window with a 15s shift across the transcriptions. Boundaries were again forced where a break of over 5 seconds occurred in the transcriptions and any extremely short windows ( $<$8s or $\leq$16 words) were removed before retrieval. A contrast run was performed which made no use of the inferred broadcast structure and simply generated windows of length 30s with shift 15s. The results are given in Table 6.

| Breaks | Post | RS | NRS | NS | #Dup | AveP | R-P |
|---|---|---|---|---|---|---|---|
| HB | B | 96.4 | 39.98 | 18.80 | 717829 | - | - |
| - | B | 96.0 | 39.39 | 27.42 | 752913 | - | - |
| **HB** | A | 78.4 | 3.74 | 1.68 | 2707 | 41.47 | $\boxed{41.98}$ |
| - | A | 78.3 | 3.43 | 2.39 | 3801 | $\boxed{41.71}$ | 40.07 |

**Table 6:** Effect before (B) and after (A) post-processing of enforcing a hard break (HB) during window generation when a gap of $>$5s exists, on `HTK-p2` transcriptions

These results show that using the structural information derived from segmentation and commercial elimination increases the `%retrieved` for relevant stories whilst also reducing the `%retrieved` for non-stories and number of duplicates both before and after post-processing. However, although the R-precision increases by 4.7% relative, the AveP decreases by 0.6% relative with a corresponding

[6]Since it was shown in section 6.1 that retrieval performance increased when the `C-1` scheme was used to filter the audio directly, the following experiments are on our final transcriptions, namely `HTK-p2`, which use this strategy and have a lower word error rate of 20.5%.

increase of 9% relative in non-relevant story `%retrieved`. It therefore appears that using 5 second gaps in the audio to restrict the initial window generation in the way described is not beneficial for retrieval (when measured by AveP, scoring duplicates as irrelevant)[7], so this was removed for subsequent experiments.[8]

### 6.3 Changing the Retriever

The retrieval strategy was developed for the case where story boundaries are known and was tested on the TREC-7 SDR data (Garofolo et al., 1999; Jourlin et al., 1999a). Experiments were conducted to see if the devices and parameter sets used during retrieval generalised well to the story unknown task on the larger TREC-8 SU collection.

**6.3.1 Semantic Poset Indexing (SPI)** Semantic poset indexing was incorporated into our system to allow semantic relationships between words to be captured (Jourlin et al., 1999a). Specifically, we use geographic trees to encode relationships between place names, and related unambiguous nouns are extracted automatically from WordNet (Fellbaum, 1998).

Although results on many sets of transcriptions for the TREC-7 SDR data showed that SPI gave a small but consistent improvement in AveP (Jourlin et al., 1999b), this did not appear to be the case when SPI was included within our complete TREC-8 story-known evaluation system (Johnson et al., 2000). An experiment was therefore conducted to see the effect of removing SPI from our story-unknown system.

| SPI | Post | RS | NRS | NS | #Dup | AveP | R-P |
|---|---|---|---|---|---|---|---|
| Y | B | 96.0 | 39.39 | 27.42 | 752913 | - | - |
| N | B | 95.8 | 37.52 | 24.92 | 698461 | - | - |
| **Y** | A | 78.3 | 3.43 | 2.39 | 3801 | 41.71 | 40.07 |
| N | A | 79.2 | 3.43 | 2.40 | 3764 | $\boxed{43.42}$ | 43.35 |

**Table 7:** Effect before (B) and after (A) post-processing of including semantic poset indexing, on `HTK-p2` transcriptions

The results, given in Table 7, show that including SPI does slightly increase relevant story `%retrieved` before post-processing. However, the non-relevant story and non-story `%retrieved` and the number of duplicates are also increased. After post-processing, the number of duplicates remains slightly higher for the SPI case, and the `%retrieved` for relevant stories drops. The decrease in AveP of 3.9% relative when including SPI is thought to be due to the inclusion of semantically related words adding significantly more non-relevant or non-stories than relevant stories during retrieval.[9] This unexpected result needs further investigation, but in the meantime SPI was removed for subsequent experiments.

**6.3.2 Blind Relevance Feedback (BRF)** During blind relevance feedback a certain number of terms, $t$, are added to the query whilst making the assumption that the top $r$ documents returned by running the retriever on the test collection are relevant. The values of $t = 5$ and $r = 10$ used in our system were chosen from experimental results for the case where story boundaries are known,

[7] It is not clear that increasing AveP to the detriment of other measures always increases performance from the point of view of real users, for example those concentrating only on high ranked documents.

[8] There was still a very small gain (0.1%) for the straight forward windowing when using the structural information to enforce hard breaks during post-processing.

[9] Note that there is a complicated interaction between the use of SPI and other techniques such as blind relevance feedback.

using both the TREC-7 SDR and adhoc collections (Voorhees and Harman, 1999). An experiment was conducted to see if these values generalised to the story-unknown case, where for example, the number of documents was much greater and the average document length much less.

| r | t | RS | NRS | NS | #Dup | AveP | R-P |
|---|---|---|---|---|---|---|---|
| 5 | 5 | 78.8 | 3.44 | 2.38 | 3719 | 42.69 | 43.48 |
| **10** | **5** | 79.2 | 3.43 | 2.40 | 3764 | 43.42 | 43.35 |
| 15 | 5 | 79.0 | 3.43 | 2.39 | 3765 | 42.83 | 43.59 |
| 20 | 5 | 78.9 | 3.43 | 2.40 | 3784 | 43.04 | 43.16 |
| 10 | 3 | 78.7 | 3.43 | 2.41 | 3773 | 42.40 | 42.71 |
| **10** | **5** | 79.2 | 3.43 | 2.40 | 3764 | 43.42 | 43.35 |
| 10 | 10 | 79.4 | 3.45 | 2.34 | 3742 | $\boxed{44.28}$ | 44.23 |
| 10 | 12 | 79.3 | 3.45 | 2.34 | 3745 | 44.21 | 44.33 |
| 10 | 15 | 79.5 | 3.45 | 2.33 | 3733 | 44.20 | $\boxed{44.50}$ |
| - | 0 | 78.4 | 3.45 | 2.37 | 3687 | 41.52 | 42.93 |

**Table 8:** Effect of altering BRF parameters, on `HTK-p2` transcriptions

The results given in Table 8 show that including blind feedback within the system improved the AveP by 4.6% relative. The value of $r$ chosen from experiments with a story-known system, seems to generalise well to the story-unknown case despite the different nature of the documents in both cases. However, AveP could be increased further by adding more terms to the query during the feedback process.[10] This increase in performance may be due to slightly sub-optimal values being used originally, or because of differences when moving from the story-known to story-unknown task, for example there are more "documents" and simple blind feedback now captures relatively more short-term dependencies than the parallel blind feedback which is not windowed.

**6.3.3 Parallel Blind Relevance Feedback (PBRF)** Another experiment was conducted to see the effect of changing the $t$ and $r$ parameters for the parallel blind relevance feedback stage. The values of $t = 7$ and $r = 20$ used in our system were chosen in the same way as the BRF parameters and the results, given in Table 9, show that these generalise well, with the inclusion of PBRF leading to a relative increase of 13.5% in AveP.

| r | t | RS | NRS | NS | #Dup | AveP | R-P |
|---|---|---|---|---|---|---|---|
| 10 | 7 | 79.0 | 3.47 | 2.30 | 3635 | 43.44 | 45.05 |
| 15 | 7 | 80.1 | 3.46 | 2.30 | 3737 | 43.92 | 44.47 |
| **20** | **7** | 79.4 | 3.45 | 2.34 | 3742 | $\boxed{44.28}$ | 44.23 |
| 25 | 7 | 79.8 | 3.45 | 2.32 | 3785 | 44.02 | $\boxed{45.61}$ |
| 30 | 7 | 80.5 | 3.44 | 2.34 | 3799 | 43.29 | 44.98 |
| 20 | 5 | 79.1 | 3.43 | 2.38 | 3786 | 43.22 | 42.96 |
| **20** | **7** | 79.4 | 3.45 | 2.34 | 3742 | $\boxed{44.28}$ | 44.23 |
| 20 | 10 | 80.6 | 3.47 | 2.26 | 3736 | 44.23 | 44.68 |
| - | 0 | 77.2 | 3.48 | 2.30 | 3611 | 39.01 | 40.40 |

**Table 9:** Effect of altering PBRF parameters, on `HTK-p2` transcriptions

[10] Although again the parameter set which gives best R-precision does not also give best AveP.

**6.3.4 Altering the Retrieval Parameters** The combined-weight formula used in the retriever contains two parameters, $b$ and $K$, which modify the influence of document length and term frequency respectively (Robertson and Spärck Jones, 1997). The values of $b = 0.5$ and $K = 1.0$ used in our system were chosen for the story-known case and an experiment was conducted to see if these values generalised well to the story-unknown case. In particular (Robertson and Spärck Jones, 1997) describe the parameter $b$ as

*"The constant b, ... modifies the effect of document length. If b=1 the assumption is that documents are long simply because they are repetitive, while if b=0 the assumption is that they are long because they are multitopic"*

For the story-unknown case, no prior information about the document lengths is available, so we assume that longer documents would contain more topics, implying that $b$ should be set to 0. Increasing $K$ means that more emphasis is placed on the term frequencies, so a word that occurs many times in a document becomes relatively more important.

| b | K | RS | NRS | NS | #Dup | AveP | R-P |
|---|---|---|---|---|---|---|---|
| 0.0 | 1.0 | 79.3 | 3.64 | 1.84 | 3271 | $\boxed{45.50}$ | $\boxed{47.04}$ |
| 0.25 | 1.0 | 80.3 | 3.50 | 2.18 | 3643 | 45.50 | 45.05 |
| **0.5** | **1.0** | 79.4 | 3.45 | 2.34 | 3742 | 44.28 | 44.23 |
| 1.0 | 1.0 | 78.1 | 3.31 | 2.78 | 3934 | 39.86 | 40.50 |
| 0.0 | 0.75 | 79.1 | 3.64 | 1.85 | 3244 | 45.05 | 46.05 |
| 0.0 | 1.0 | 79.3 | 3.64 | 1.84 | 3271 | 45.50 | $\boxed{47.04}$ |
| 0.0 | 1.25 | 79.4 | 3.64 | 1.83 | 3261 | $\boxed{45.84}$ | 46.91 |
| 0.0 | 1.5 | 79.6 | 3.64 | 1.82 | 3277 | 44.96 | 45.34 |

**Table 10:** Effect of altering b and K parameters, on `HTK-p2` transcriptions

The results, given in Table 10, confirm that AveP can be increased by 2.8% relative (and R-P by 6.4% relative) by setting $b = 0$ for the retrieval on the windowed test collection.[11] A further small increase in AveP can be obtained by increasing K.

### 6.4 Optimising the Post-Processing

The post-processing stage attempts to reduce the number of duplicate hits from the retriever output by merging some windows. Various rules can be applied to define when and how the merges should take place. Here we investigate the effect of two of the alternatives.

**6.4.1 Altering the Merge Length** In an attempt to eliminate duplicates, the post-processing merges all stories originating from the same broadcast whose midpoints occur within a certain time scale, $T_m$. Changing this parameter models the trade-off between over-generating hits from the same story and over-combining hits from different (neighbouring) stories. It was felt that the probability of two adjacent stories being relevant to the same query would be small (although related to the number of hits returned by the retriever and the number of relevant documents for the query) and hence a fairly large merge time of 75 seconds was used during the evaluation. It was also hoped that enforcing hard breaks when gaps of over 5 seconds in the audio occurred would help reduce the problem of over-merging.[12].

[11] The values used for the PBRF stage, which only uses the parallel corpus, were not changed.
[12] See section 6.2.1

An experiment was conducted to find the effect on performance of varying $T_m$. The results are illustrated in Figure 3 and summarised in Table 11.

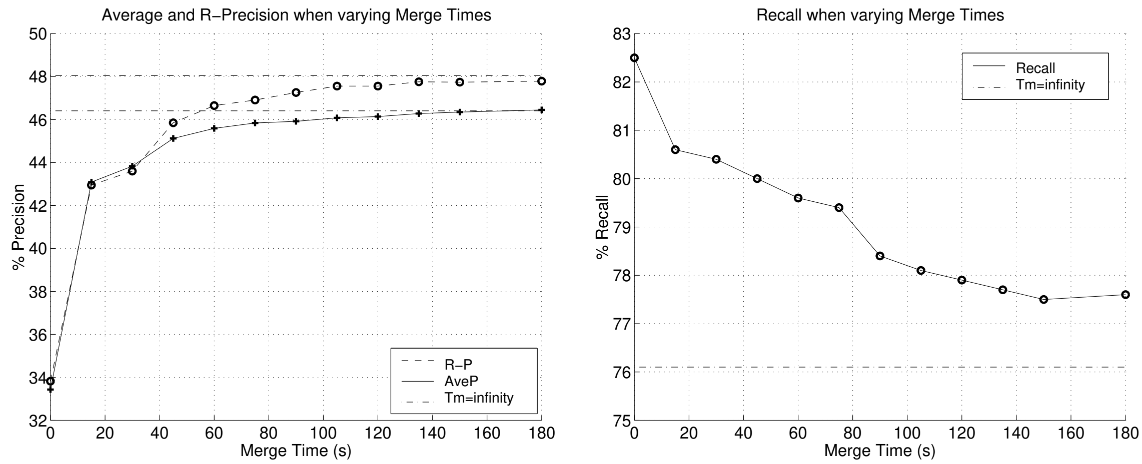


**Figure 3:** Effect on AveP, R-P and recall of changing the merge length during post-processing

| $T_m$ (s) | RS | NRS | NS | #Dup | AveP | R-P |
|---|---|---|---|---|---|---|
| 0 | 82.5 | 2.13 | 1.00 | 22188 | 33.43 | 33.82 |
| 15 | 80.6 | 3.30 | 1.63 | 7571 | 43.09 | 42.95 |
| 30 | 80.4 | 3.42 | 1.70 | 6030 | 43.83 | 43.60 |
| 45 | 80.0 | 3.52 | 1.78 | 4678 | 45.12 | 45.85 |
| 60 | 79.6 | 3.59 | 1.81 | 3896 | 45.59 | 46.65 |
| **75** | 79.4 | 3.64 | 1.83 | 3261 | 45.84 | 46.91 |
| 90 | 78.4 | 3.67 | 1.85 | 2877 | 45.92 | 47.26 |
| 120 | 77.9 | 3.71 | 1.88 | 2343 | 46.14 | 47.56 |
| 135 | 77.7 | 3.73 | 1.89 | 2184 | 46.28 | 47.76 |
| 150 | 77.5 | 3.73 | 1.91 | 2065 | 46.35 | 47.74 |
| 180 | 77.6 | 3.75 | 1.91 | 1874 | $\boxed{46.45}$ | 47.79 |
| $\infty$ | 76.1 | 3.74 | 2.00 | 1677 | 46.41 | $\boxed{48.05}$ |

**Table 11:** Effect of changing the merge length during post-processing ($\infty \equiv$ whole show merged)

The results show that both Average and R-Precision increase monotonically towards an asymptote as the merge time is increased suggesting that the "best" system would use a large merge time of around 3 minutes.[13] However, although merging dramatically decreases the number of duplicates, hence allowing the lower scoring relevant stories to gain a higher rank in retrieval (thus increasing precision), some distinct relevant stories are also being recombined (thus reducing relevant story `%retrieved`).

Although the precision values have reached an asymptote when $T_m = 180s$, relevant story `%retrieved` (i.e. *recall*) falls further when the merge time continues to be increased. Which value of $T_m$ to use therefore depends on the relative importance of precision and recall to the user and in particular how they feel about seeing duplicates. It is felt that the $T_m$ of our system (75s) offers a reasonable compromise between the rising precision and falling recall when merging is increased.

[13]Using different merge times for different data sources will be also investigated in the future.

#### 6.4.2 Reducing the Retrieval of Non-Relevant Stories

If the number of non-relevant stories returned during retrieval could be reduced without affecting the retrieval of relevant stories, then the post-processing stage could be both speeded up and improved due to a lower false alarm rate. A threshold was thus applied to the document scores during retrieval to ensure that only the windows with the best match for any given query were used in further post-processing. The results including the number of windows enterring the post-processing stage for each cut-off level, are given in Table 12 using $T_m = 75s$.

| | Post | RS | NRS | NS | #Dup | # windows |
|---|---|---|---|---|---|---|
| **0.1** | B | 96.5 | 44.20 | 30.41 | 871,428 | 1,448,864 |
| 1 | B | 95.4 | 33.05 | 21.80 | 574,486 | 1,003,679 |
| 5 | B | 89.8 | 10.38 | 5.58 | 127,084 | 259,030 |
| 7 | B | 86.4 | 6.30 | 3.46 | 73,592 | 154,450 |
| 10 | B | 82.1 | 3.22 | 1.66 | 38,499 | 80,196 |
| 12 | B | 76.1 | 2.14 | 1.11 | 25,873 | 53,969 |

| | Post | RS | NRS | NS | #Dup | AveP | R-P |
|---|---|---|---|---|---|---|---|
| **0.1** | A | 79.4 | 3.64 | 1.83 | 3261 | 45.84 | 46.91 |
| 1 | A | 79.4 | 3.64 | 1.83 | 3256 | 45.84 | 46.91 |
| 5 | A | 78.4 | 3.43 | 1.73 | 5238 | 45.82 | 46.91 |
| 7 | A | 77.2 | 3.17 | 1.66 | 6445 | 45.77 | 46.91 |
| 10 | A | 77.0 | 2.35 | 1.15 | 11919 | 45.78 | 46.91 |
| 12 | A | 72.7 | 1.78 | 0.86 | 12591 | 45.63 | 46.69 |

**Table 12:** Effect both before (B) and after (A) post-processing of varying the low score threshold just before post-processing

By increasing the low score threshold from 0.1 to 10, the final number of duplicates is increased, due to fewer intermediate windows being available for merging, and the `%retrieved` for relevant stories drops. However, the number of windows entering the post-processing stage can be reduced from 1,448,864 to 80,196 with a drop of less than 0.1% in AveP. For real systems, where speed of retrieval is important, the higher threshold should thus be used during post-processing.

### 6.5 Future Experiments

It has been shown that windowing the parallel corpus (rather than using the naturally existing document boundaries) before implementing query expansion using parallel blind relevance feedback can improve performance for this task on the TREC-7 data. (Robinson et al., 1999). Future work will therefore investigate whether this improvement extends to the TREC-8 data when implemented within the framework of our system.

We also hope to investigate whether the benefits of using parallel blind relevance feedback for *document* expansion on the TREC-8 story-known task (Johnson et al., 2000) can be translated into better performance for the story-unknown case.

## 7 Conclusions

This paper has described a system for retrieving relevant portions of complete broadcast news shows when only the audio data is available.

A novel method of automatically detecting and eliminating commercials by directly searching the audio was used and was shown to increase performance for the TREC-8 story unknown task, whilst reducing the computational effort required by around 8% when implemented before recognition. Applying the automatically determined commercial boundaries as a filter after retrieval was also shown to improve performance on other sets of transcriptions.

A sophisticated large vocabulary speech recogniser was used to eliminate sections of audio corresponding to pure music and produce high quality transcriptions. Our final recognition system, using a 108,000 word vocabulary, ran in 13xRT[14] and gave a WER of 20.5%, with the 60,000 word first-pass output giving 26.6% WER in 3xRT.

A windowing system was used to create quasi-documents on which the retrieval engine was run. A post-processing stage was then used to recombine windows thought to originate from the same story source by removing windows which were broadcast within a certain time of a higher scoring window. It was shown that incorporating the information about the structure of the broadcast gained from commercial elimination and segmentation, during the post-processing stage increased performance by a small amount, although no gain was found when using this information during window generation.

Experiments in retrieval showed that blind relevance feedback continued to be beneficial, but that semantic poset indexing, which had been found useful in earlier tests on other data (Jourlin et al., 1999a), was not helpful for this collection. Post-processing experiments showed precision could be increased at a cost to recall by performing more merging, whilst the speed of post-processing could be increased with little loss in precision, by using only the higher scoring windows from the retriever.

Combining the various techniques described in this paper has been shown to produce a system capable of giving an AveP of 46.5% on the TREC-8 story-unknown data set.

## Acknowledgements

This work is in part funded by an EPSRC grant reference GR/L49611.

## References

Abberley, D., Renals, S., Robinson, T., and Ellis, D. (2000). The THISL SDR System at TREC-8. To appear. In *Proc. TREC-8*, Gaithersburg, MD.

Bimbot, F. and Mathan, L. (1993). Text-Free Speaker Recognition using an Arithmetic Harmonic Sphericity Measure. In *Proc. Eurospeech'93*, volume 1, pages 169–172, Berlin, Germany.

Dharanipragada, S., Franz, M., and Roukos, S. (1999). Audio-Indexing for Broadcast News. In *Proc. TREC-7, NIST SP 500-242*, pages 115–119, Gaithersburg, MD.

Dharanipragada, S. and Roukos, S. (1997). Experimental Results in Audio Indexing. In *Proc. DARPA 1997 Speech Recognition Workshop*, Chantilly, VA.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Garofolo, J. S., Auzanne, C. G. P., and Voorhees, E. M. (2000). 1999 TREC-8 Spoken Document Retrieval Track: Overview, Results and Analyses. To appear. In *Proc. TREC-8*, Gaithersburg, MD.

Garofolo, J. S., Voorhees, E. M., Auzanne, C. G. P., Stanford, V. S., and Lund, B. A. (1999). 1998 TREC-7 Spoken Document Retrieval Track Overview and Results. In *Proc. TREC-7, NIST SP 500-242*, pages 79–90, Gaithersburg, MD.

[14]On a single processor of a dual processor Pentium III 550MHz running Linux.

Garofolo, J. S., Voorhees, E. M., Stanford, V. M., and Spärck Jones, K. (1998). TREC-6 1997 Spoken Document Retrieval Track Overview and Results. In *Proc. TREC-6, NIST SP 500-240*, pages 83–92, Gaithersburg, MD.

Gauvain, J.-L., de Kercadio, Y., Lamel, L., and Adda, G. (2000). The LIMSI SDR System for TREC-8. To appear. In *Proc. TREC-8*, Gaithersburg, MD.

Hauptmann, A. and Witbrock, M. (1998). Story Segmentation and Detection of Commercials in Broadcast News Video. In *Proc. Advances in Digital Libraries (ADL '98)*, pages 168–179, Santa Barbara, CA.

Johnson, S. E., Jourlin, P., Spärck Jones, K., and Woodland, P. C. (2000). Spoken Document Retrieval for TREC-8 at Cambridge University. To appear. In *Proc. TREC-8*, Gaithersburg, MD.

Johnson, S. E. and Woodland, P. C. (2000). A Method for Direct Audio Search with Applications to Indexing and Retrieval. To appear. In *ICASSP'2000*, Istanbul, Turkey.

Jourlin, P., Johnson, S. E., Spärck Jones, K., and Woodland, P. C. (1999a). General Query Expansion Techniques for Spoken Document Retrieval. In *Proc. ESCA Workshop on Extracting Information from Spoken Audio*, pages 8–13, Cambridge, England.

Jourlin, P., Johnson, S. E., Spärck Jones, K., and Woodland, P. C. (1999b). Improving Retrieval on Imperfect Speech Transcriptions. In *Proc. ACM SIGIR '99*, pages 283–284, Berkeley, CA.

Odell, J. J., Woodland, P. C., and Hain, T. (1999). The CUHTK-Entropic 10xRT Broadcast News Transcription System. In *Proc. 1999 DARPA Broadcast News Workshop*, pages 271–275, Herndon, VA.

Robertson, S. E. and Spärck Jones, K. (1997). *Simple, Proven Approaches to Text Retrieval*. Technical Report TR-356 Cambridge University Computer Laboratory.

Robinson, A., Abberley, D., Kirby, D., and Renals, S. (1999). Recognition, Indexing and Retrieval of British Broadcast News with the THISL System. In *Proc. Eurospeech 99*, pages 1267–1270, Budapest, Hungary.

van Mulbregt, P., Carp, I., Gillick, L., Lowe, S., and Yamron, J. (1999). Segmentation of Automatically Transcribed Broadcast News Text. In *Proc. DARPA 1999 Broadcast News Workshop*, pages 77–80, Herndon, VA.

Voorhees, E. M. and Harman, D. K. (1999). Overview of the Seventh Text REtrieval Conference (TREC-7). In *Proc. TREC-7, NIST SP 500-242*, pages 1–24, Gaithersburg, MD.