Information Retrieval from Unsegmented Broadcast News Audio

Sue E Johnson, Pierre Jourlin, Karen Jones, Philip C Woodland

INTERNATIONAL JOURNAL OF SPEECH TECHNOLOGY 4, 251–268, 2001

Information Retrieval from Unsegmented Broadcast News Audio*

SUE E. JOHNSON
Engineering Department, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, UK
sej28@eng.cam.ac.uk

PIERRE JOURLIN AND KAREN SPÄRCK JONES
Computer Laboratory, University of Cambridge, Pembroke Street, Cambridge, CB2 3QG, UK
pj207@cl.cam.ac.uk
ksj@cl.cam.ac.uk

PHILIP C. WOODLAND
Engineering Department, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, UK
pcw@eng.cam.ac.uk

Received June 14, 2000; Revised March 22, 2001

Abstract. This paper describes a system for retrieving relevant portions of broadcast news shows starting with only the audio data. A novel method of automatically detecting and removing commercials is presented and shown to increase the performance of the system while also reducing the computational effort required. A sophisticated large vocabulary speech recogniser which produces high-quality transcriptions of the audio and a window-based retrieval system with post-retrieval 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 46.8%, when duplicates are scored as irrelevant, is shown to be achievable using this system, with the corresponding word error rate of the recogniser being 20.5%.

Keywords: spoken document retrieval, automatic speech recognition, story segmentation, commercial detection, information retrieval

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 in 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 given manually

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

pre-defined "stories". Many different research groups produced SDR systems, most of which generated word-level transcriptions for each "story" (document), which were then run through a standard text-based retrieval engine. Although other methods were tried, such as producing phone-level transcriptions for searching (Ng et al., 2000), or using a machine-translation approach to address the mis-match between imperfectly transcribed spoken documents and text-based queries (Franz et al., 2000), the approach of combining a standard word-level recogniser with text-based retrieval techniques proved to be the most successful (Garofolo et al., 2000). The introduction of a text-based "parallel" corpus, for example, newswire data which overlapped with the epoch of the spoken documents, allowed the effects of the transcription errors to be reduced and the difference in performance between using manually derived and automatically generated transcriptions was eliminated. Garofolo et al. (1998, 1999b, 2000) give a summary of the performance of the state of the art systems during 1997–1999.

Unfortunately, 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 the automatically generated transcriptions, retrieving on these windows, and then using window recombination after retrieval (Abberley et al., 2000; Dharanipragada and Roukos, 1997; Dharanipragada et al., 1999; Robinson et al., 1999). The second technique 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 (van Mulbregt et al., 1999), but it is also possible to use additional audio or video cues (Hauptmann and Witbrock, 1998). This paper focuses on experiments on a system which uses both ideas, exploiting properties of the audio to impose some structure on complete broadcasts, while using windowing/recombination techniques to find relevant passages during retrieval.

Many news broadcasts contain portions which are irrelevant to the information need of the user, such as musical jingles and commercials. These are generally pre-recorded and re-broadcast many times over the course of a few months, leading to portions of the audio having identical characteristics. By searching the audio for exact repetitions, many of the commercials therefore can be detected automatically. This search can be speeded up by applying techniques similar to those used in content-based audio retrieval (Foote, 1999; Wold et al., 1996), where portions of audio are represented by a set of characteristic features, and a distance measure between these feature sets is defined. However, instead of ranking the similarity of audio segments, a small threshold can be applied to find segments which are thought to be repeats, hence allowing the automatic detection of commercials on very large audio databases. This paper also describes a system for automatically detecting and eliminating commercials based on this method.

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 the method for automatically detecting and eliminating commercials, while 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 removing commercials, enforcing structure within the broadcasts and improving the retrieval and post-processing are given in Section 6. A discussion of ongoing work is presented in Section 7, and finally conclusions are given in Section 8.

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 news material had been independently marked up with manually-generated story boundaries, and the relevance assessments for the queries were for these whole stories. The story boundaries were also used to define the official story-IDs for the scoring procedure.

The queries were chosen by NIST to span a wide range of topics contained in the broadcast news, and varied in difficulty and in numbers of relevant documents. The query set included, for example:[1]

How safe are the world's drinking water supplies?
What foreign countries has Pope John Paul II visited or does he plan to visit?
What natural disasters occurred in the world in 1998 causing at least 10 deaths?

Participants had to produce {`show:time`} stamps automatically 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 were labelled as irrelevant.[2] 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 are the Average Precision (averaged over precision values computed after each relevant document is retrieved), denoted *AveP*, and R-precision (precision when the number of documents retrieved equals the number of relevant documents), denoted R-P, averaged over all the queries.[3]

The data used for the evaluation was the February 1998 to June 1998 subset of the audio from the TDT-2 corpus (Cieri et al., 1999). 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, while retrieval was *retrospective*, i.e., any data up until the end of the collection (June 30, 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 user's 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 relevance matches occurring, and hence increase overall retrieval performance.

By assuming that (usually) only commercials are repeated, a direct audio search to find segments of repeated audio can be used to indicate the probable location of many of the commercials. By then applying a set of rule-based filters to limit the chance of the identified segments being false matches, or repeated news bulletins, an accurate prediction can be made of where many of the commercials occur.

3.1. An Introduction to Direct Audio Search

To be able to search audio accurately and quickly, the data must first be sampled, and a set of features which capture the characteristics of the audio must be defined. Whilst it is possible in theory just to use the amplitude of the audio in the time-domain directly after sampling,[4] more complicated features such as FFT-based coefficients or zero-crossing rates require several samples to be grouped together into a single frame (typically computed over 25 ms with a 10 ms frame shift). A single *feature vector* is then calculated for each frame within the portion of audio being considered.

Searching for every possible match between the exact sequences of audio frames over a large database is computationally intractable, however recent work in content-based audio retrieval has shown that frames can be grouped together into larger audio *segments*, and general properties of the audio, (such as pitch and loudness) can be used to identify the acoustic "similarity" between different segments. This requires defining a way to represent the audio segments mathematically and an associated "distance" metric which encaptures the dis/similarity between segments (Foote, 1999). Similar segments thus have a low *distance* between their representations, while a distance of zero indicates identical (i.e., repeated) segments.

Several methods of representing the entire segment of audio have been investigated. These range from calculating the mean, variance and small-lag autocorrelation of the frame-level feature vectors (e.g., in Wold et al., 1996), through using a histogram to represent the overall distribution of each feature (e.g., in Kashino et al., 1999), to using the relative outputs from a discriminatively-trained vector quantiser in Foote (1997). The ability of the final representation to capture the specific properties of the piece of audio therefore depends on both the choice of initial features, and the method of representing these features over the entire segment. Distance metrics that have been used include the covariance-weighted Euclidean distance and the Cosine distance between the representative feature vectors (Foote, 1999).

Content-based audio retrieval systems are generally presented with a single *query* segment of audio (which we shall denote the *cue-audio*) and a static set of segments to search (which we shall denote the *library-audio*). The system then calculates the distance between the cue-audio and each segment in the library-audio and produces a ranked list of possible matches in decreasing order of similarity.

Our system for finding repeated audio to help detect commercials differs from this in three main ways. First, the database is not naturally divided into segments, so a windowing system is used to define the "segments" to be used. Second, there are no separate cue-audio segments, since each part of the database must be compared to every other part which temporally precedes it. Finally, the task is more like the conventional *filtering* IR task, in that rather than producing a ranked list of possible matches, the system must produce a binary decision as to whether any two given segments match or not. This is done by comparing the score to a small threshold (to compensate for asynchronousity between the audio event in question and the sampling, framing or segmenting) to decide if the segments match or not.

3.2. The Search Method

A system to detect areas of the audio that were re-broadcasts of previous sections within the TREC-8 database was built using the direct audio search method described in Johnson and Woodland (2000).

The data was sampled at 16 kHz and grouped into overlapping frames of 25 ms with a 10 ms frame shift. Each frame was represented by 13 mel-frequency PLP-cepstral coefficients (Woodland et al., 1997) and their first and second derivatives. These feature vectors are available at no extra computational cost since they are used in the subsequent speech recognition process. The audio was then divided into five second long overlapping segments with a shift of one second between adjacent segments. The correlation matrix, Σ, for the feature vectors was calculated for each segment and used as the mathematical representation of the audio within that segment.

Each segment was then compared to all the segments that had been broadcast more than two days previously by the same broadcaster.[5] We denote this library of previously broadcast segments the *broadcast history*. The 2-day delay between the cue-audio segment and the broadcast history was introduced to reduce the possibility of a real news story remaining in the news and being directly re-broadcast.

The distance measure used for the matching was the arithmetic harmonic sphericity (AHS) distance (Bimbot and Mathan, 1993)

$$d(X, Y) = \log\left[tr\left(\Sigma_y \Sigma_x^{-1}\right) \cdot tr\left(\Sigma_x \Sigma_y^{-1}\right)\right] - 2\log(D)$$

where *tr* is the trace of the matrix and D is the dimension of the feature vector.

This representation and distance measure was chosen because previous work in speaker clustering (Johnson and Woodland, 1998) showed it enabled acoustically similar segments to be found both accurately and quickly, and early trials into detecting repeated audio using this representation proved successful.

By comparing the inter-segment distances to a small empirically-found threshold (Johnson et al., 2000), the segments of audio thought to represent re-broadcasts were identified.

3.3. Commercial Elimination

The audio sections identified as re-broadcasts by the method described above were used as the basis for the commercial elimination system. A set of rule-based filters was applied to the postulated re-broadcasts to improve the overall performance of the system.

First safeguards were introduced to reduce the probability of stories being wrongly labelled as commercials, either due to false audio matches or to the story itself being re-broadcast by playing the same audio track during different news bulletins. This was particularly important because once the audio segments have been labelled as a commercial, they are discarded and can not be recovered. The safeguards included forcing the cue-audio to match (a) a minimum number of different segments in the broadcast history and (b) a minimum number of different preceding shows.

Second smoothing was carried out so that small sections of audio that occurred between segments already labelled as possible commercials were themselves labelled as commercials. The smoothing was restricted by ensuring the resulting commercials were less than a certain length, to reduce the probability of short stories that occurred between commercials or jingles being incorrectly smoothed out. For the case of CNN a show-grammar was also introduced which only

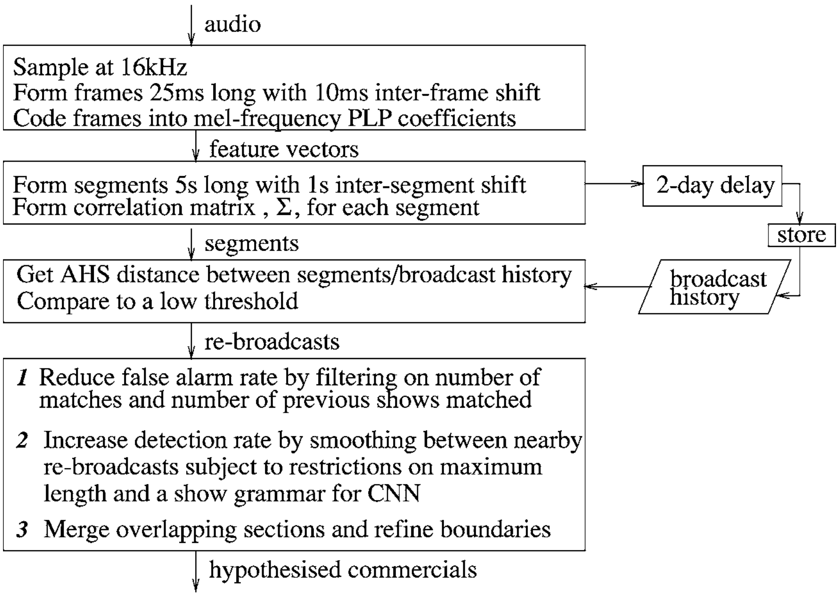
audio
Sample at 16kHz
Form frames 25ms long with 10ms inter-frame shift
Code frames into mel-frequency PLP coefficients
feature vectors
Form segments 5s long with 1s inter-segment shift
Form correlation matrix , Σ, for each segment
2-day delay
store
segments
Get AHS distance between segments/broadcast history
Compare to a low threshold
broadcast history
re-broadcasts
1 Reduce false alarm rate by filtering on number of matches and number of previous shows matched
2 Increase detection rate by smoothing between nearby re-broadcasts subject to restrictions on maximum length and a show grammar for CNN
3 Merge overlapping sections and refine boundaries
hypothesised commercials

Figure 1. The Commercial Detection Process.

allowed smoothing between commercials within certain time-limits from the start of the show. This smoothing process increased the commercial detection rate without unduly increasing the false alarm rate.

Finally the boundaries of the postulated commercials were refined to take into account the coarseness of the initial windows. The commercial detection process is illustrated in Fig. 1 and further details can be found in Johnson et al. (2000).

3.4. Results for the Commercial Elimination Stage

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 the "reference" labels are not an exact reflection of the story/commercial distinction, because a few commercials have been labelled erroneously 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.

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

The complete system consists of five main stages. First the commercials are eliminated, then the audio is transcribed using a large vocabulary automatic speech recogniser. This stage also discards portions of the audio thought to correspond to silence, pure music or pure noise.

Since the data are provided as a continuous audio stream, and no document/story boundaries are known, the transcriptions are then split into equal-length overlapping passages or *windows*. By assuming no commercials exist within a single story, the location of some of the story boundaries can be assumed from the presence

Table 1. Amount of data rejected during the commercial elimination stage. Results are given for the actual system (C-1) and a less conservative system (C-2) for comparison.

	Non-Stories		Stories		Total	
Broadcaster	%cat	Time	%cat	Time	%cat	Time
C-1						
ABC	65.5%	12.8 hrs	0.02%	28 s	20.48%	12.8hrs
CNN	35.7%	26.2 hrs	0.46%	2822 s	11.03%	27.0 hrs
PRI	16.6%	1.9 hrs	0.10%	297 s	2.16%	2.0 hrs
VOA	5.0%	0.5 hrs	0.04%	132 s	0.49%	0.5 hrs
Total	36.3%	41.4 hrs	0.23%	0.9 hrs	**8.42%**	**42.3 hrs**
C-2						
ABC	70.6%	13.8 hrs	0.07%	107 s	22.12%	13.8 hrs
CNN	59.0%	43.3 hrs	1.73%	3.0 hrs	18.91%	46.2 hrs
PRI	22.4%	2.6 hrs	0.14%	416 s	2.92%	2.7 hrs
VOA	6.0%	0.6 hrs	0.06%	208 s	0.58%	0.6 hrs
Total	52.9%	60.2 hrs	0.81%	3.2 hrs	**12.63%**	**63.4 hrs**

The percentage of each category rejected, *%cat*, and the amount of broadcast time this corresponds to, are both given. For example, $C1$ eliminated 65.5% of all non-stories from ABC, which corresponds to 12.8 hours of broadcast time.

of commercials. This information is used during this stage to make sure no window spans a known story boundary.

The retriever is then run on the windows to provide a ranked list of scores which aim to indicate the potential relevance of each window. Finally, temporally close windows are merged, subject to the known story boundaries, to reduce the number of duplicate hits generated from any given story. A block diagram of our complete system is given in Fig. 2 and more details can be found in Johnson et al. (2000).

4.1. The Transcription System

After the commercial detection and elimination stage, the audio is automatically split into segments of between 1 and 30 seconds, thought to be acoustically homogeneous, i.e., containing only one

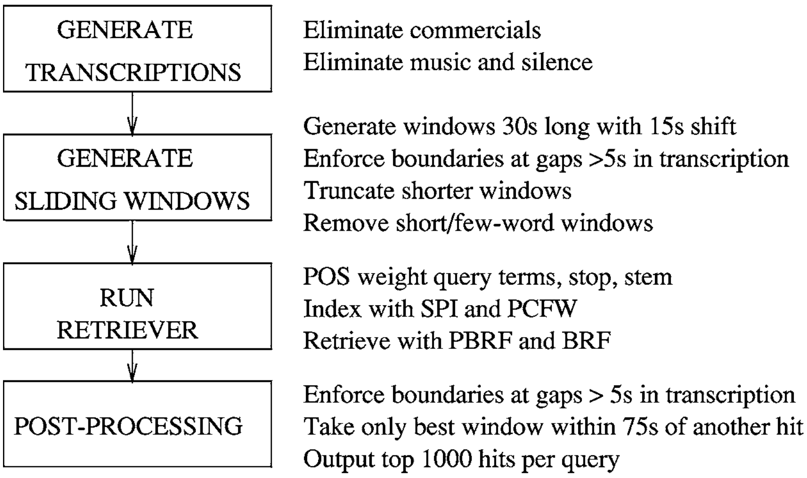
GENERATE TRANSCRIPTIONS
Eliminate commercials
Eliminate music and silence
GENERATE SLIDING WINDOWS
Generate windows 30s long with 15s shift
Enforce boundaries at gaps >5s in transcription
Truncate shorter windows
Remove short/few-word windows
RUN RETRIEVER
POS weight query terms, stop, stem
Index with SPI and PCFW
Retrieve with PBRF and BRF
POST-PROCESSING
Enforce boundaries at gaps > 5s in transcription
Take only best window within 75s of another hit
Output top 1000 hits per query

Figure 2. The Complete Whole-Show Spoken Document Retrieval System.

speaker and audio/noise condition (Hain et al., 1998). These segments are labelled by gender of speaker (male/female) and bandwidth (telephone/wideband). 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. The overall system HTK-p2 consisted of two passes through the data.

The first pass used the bandwidth but not the gender labels of the segments, employing gender-independent, bandwidth-specific triphone HMM acoustic 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.

The second pass used gender and bandwidth dependent triphone models which had been adapted to the data using maximum likelihood linear regression (MLLR) (Gales and Woodland, 1996). The MLLR adaptation was applied to automatically clustered groups of segments and used word-level adaptation supervision from the first pass. A 108,000-word vocabulary with 4-gram language model was used to generate the final one-best output. This transcription, denoted HTK-p2, gave a WER of 20.5% on the scored subset. A detailed breakdown of the error rates is shown in Table 2.

4.2. The Windowing System

To use the retrieval system, the continuous audio has to be split into discrete documents. Since no story-boundary information is given, this can be done in two ways. The first involves using information gathered automatically from the audio (or video if it were available, (Hauptmann and Witbrock, 1998)), or from the text in the transcriptions (van Mulbregt et al., 1999), to postulate where topic changes are likely to occur. A document is then defined as a continuous section of audio covering a single topic, and standard retrieval techniques can be used.

Table 2. Word Error Rates (WER) on the 10-hour subset of TREC-8 evaluation data used for scoring, for the 1-pass (HTK-p1) and 2-pass (HTK-p2) system.

Recogniser	%Corr.	%Sub.	%Del.	%Ins.	%WER
HTK-p1	77.3	18.5	4.2	3.9	26.6
HTK-p2	82.4	14.0	3.7	2.9	**20.5**

The percentage of words *correct*ly recognised along with the *substitution, deletion* and *insertion* rates are also given.

An alternative method involves splitting the audio into fixed-length passages or *windows*. The retrieval system is then run to produce a ranked list of windows, and post-processing this list to merge windows thought to originate from the same news story increases the performance of the system by reducing the number of duplicate hits generated (Abberley et al., 2000; Dharanipragada et al., 1999).

Our method combined these approaches. We used a basic windowing system but incorporated the knowledge of broadcast structure gained from the commercial elimination and segmentation stages by enforcing hard breaks where gaps of more than five seconds occurred in the transcription. Such gaps were thought to indicate the presence of either pure music (such as in a jingle) or commercials and hence offered a reasonable indicator of where a change in story might occur within the broadcast. This meant no window could be formed, and no post-processing could occur, across a postulated commercial break. Finally, very short windows (less than a certain duration or number of words) were removed before the retrieval stage.

Experiments showed the best length of window was 30 seconds, with a 15 second inter-window shift (Johnson et al., 2000). The 50% overlap between adjacent windows reduced the edge effects caused by windowing, by allowing different contexts of the words to be retained. The windowing process is illustrated in Fig. 3.

4.3. The Retrieval System

Our retriever was based on the Okapi model described in Robertson and Spärck Jones (1997).[6] The windows and queries were first *stopped* (function words such as "a", "the", "of" etc. are removed) and *stemmed* (the words are reduced to their linguistic root), using Porter's algorithm (Porter, 1980) with a manually-generated list of exceptions.

The combined-weight score, $cw(t_i, d_j)$, was computed for each query term, t_i, and each window, d_j,

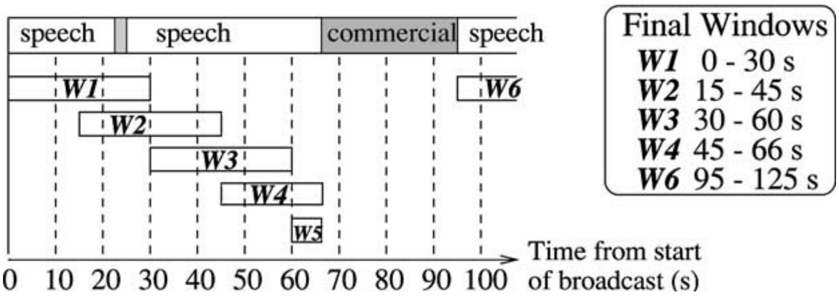

Figure 3. Windowing the transcriptions before the retrieval stage. The standard windows were 30 seconds long with inter-window shift of 15 seconds. Hard boundaries were enforced where gaps in the transcription indicated the presence of a commercial or jingle. Preceding windows (e.g., *W*4) were truncated, whilst very small windows (e.g., *W*5) were removed completely.

using the standard formula:

$$cw(t_i, d_j) = \frac{(\log N - \log n(t_i)) \, . \, tf(t_i, d_j).(K+1)}{K \, . \, (1 - b + b \, . \, ndl(d_j)) + tf(t_i, d_j)} \qquad (1)$$

where N is the number of windows in the collection, $n(t_i)$ is the number of different windows the term t_i occurs in, $tf(t_i, d_j)$ is the number of times term t_i occurs in window d_j, $ndl(d_j)$ is the length of window d_j normalised by the average window length over the collection, and K and b are tuning parameters. Summing the combined-weight scores for each query term gives an overall score for each window. The retriever then produces a ranked list of all the windows in decreasing order of score.

Blind Relevance Feedback (BRF) can be used to add terms to the query to help try to capture the information need of the user. The top R windows returned by the retriever are assumed to be relevant, and the T terms with the highest Offer Weight (Robertson and Spärck Jones, 1997) from these windows are added to the query before running the new query on the test-collection to get the final ranked list.

Our system included several modifications to this standard approach. First the query terms were weighted according to their part-of-speech to give more emphasis to nouns, which were thought to contain more information content than adjectives and verbs. Second, parallel collection frequency weighting (PCFW) was used to obtain more robust estimates of the collection frequency weights. This is equivalent to deriving the N and $n(t_i)$ terms in equation 1 from the union of the test and parallel collections.

Third semantic poset indexing (SPI) (Jourlin et al., 1999a) was used to capture some semantic information about geographical locations and unambiguous nouns within the windows. A poset is a Partially Ordered SET which allows semantically equivalent words (those which have the same meaning) and sub-categorisations (words whose meaning is a subset of other words) to be captured. For example, queries asking about “England”, would also be concerned with stories about “London”,[7] whilst information about “flu” is also relevant to “influenza”. By adding such related terms to the windows, the potential for a match with a relevant query is increased.

Finally, two stages of blind relevance feedback (BRF) were used. The first was on the parallel collection (PBRF), which does not contain transcription errors and has known story boundaries, thus allowing robust terms related to the same topic to be identified more easily. The second stage used the PBRF-expanded query on the test collection. This helps more syntactic relations to be captured (since the documents are short, possibly multi-topic, windows) and potentially allows the recovery from systematic transcription errors.

4.4. The Post-Processing Stage

Since most news stories span several windows, it is probable that a single relevant story will give rise to more than one high-scoring window in the ranked list returned by the retriever. This implies many of the matching windows form *duplicate* hits of higher-ranking windows that originated from the same source story. This is undesirable because the user does not wish to see the same story twice, and is heavily penalised within the TREC-8 SU scoring framework (see Section 5 for more details).

To reduce the number of duplicate hits produced, a post-processing stage was introduced that attempts to identify the windows returned by the retriever which originated from the same source story for each query. All but the highest scoring window from each postulated story were then eliminated before producing the final ranked list. This process was called window-*merging*.

Two retrieved windows were labelled automatically as belonging to the same source story if they originated from within a certain time period, T_m, in the same broadcast. The inferred structure of the broadcast gained during commercial detection and acoustic segmentation also was used in this stage by again enforcing hard story boundaries at gaps of more than five seconds in the transcriptions.

5. Experiments and Scoring Measures

Experiments were conducted on the 1998 TREC-8 SU test data, which consisted of 21,754 manually-labelled *stories* and 6,294 portions of audio between the stories that were not considered to be information-bearing. These *non-stories* were mainly commercials and filler portions between two stories.[8] 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.[9]

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. While 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.

An alternative suggestion was to remove all duplicates before scoring, but this is also unsatisfactory since it offers no incentive to attempt to reduce the number of multiple-hits generated per story. For example, suppose there were 50 relevant stories for a given query. Since the retriever returns the top 1000 matches, it could give five entries in the ranked list for each of 200 different stories. Providing no more than 150 of these stories were non-relevant, the system would not be judged any differently to an obviously superior system which produces only a single entry in the ranked list for each relevant story.

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.

5.1. Significance Tests

The Matched-Pair Sign Test is used to measure the statistical significance of differences between systems (van Rijsbergen, 1979). This uses the average precision *computed for each query independently* to compare the number of queries for which one system performed better than another to the number of queries which performed worse. Consistent improvement across all queries, even if only by a very small amount, therefore gives a more significant result than a potentially larger increase in AveP (averaged across all queries) caused by an improvement in only a few queries. A significance level of lower than 5% is considered statistically significant, and the level is reported.

5.2. TREC-8 Evaluation Results

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.[10] These results confirm the findings made with SDR systems where the story boundaries are given in advance, that the fall-off in AveP with WER is gentle for WERs below approximately 40% (Johnson et al., 2000; Garofolo et al., 2000). This is in part due to the inclusion of compensatory methods such as BRF and the use of a parallel text corpus.

Real user-based systems can benefit from increased standards of automatic transcription, even if the AveP is unaltered, since the documents presented to the user contain fewer transcription errors and thus are easier to read and more reliable, thereby reducing the dependence of the user on the original audio.

Table 3. %Retrieved of Relevant Stories (RS), Non-Relevant Stories (NRS), Non-Stories (NS) and number of duplicates (#Dup), when removing commercials before transcription (BT) or after retrieval (AR) using the `HTK-p1` transcriptions. Average and R-Precision are also given after the post-processing stage. Results are given for both the original `C-1` and less conservative `C-2` commercial elimination systems.

BT	AR	RS	NRS	NS	#Dup	AveP	R-P
Results before post-processing							
–	–	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	–	–
Results after post-processing							
–	–	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	$\boxed{41.77}$

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 to be conducted that investigated the effects of automatically detecting and removing commercials.

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 obtained before post-processing from applying no commercial elimination (−), the TREC-8 evaluation system (`C-1`) that removed 8.4% of the data, and the less conservative run (`C-2`) that removed 12.6% of the data, are given in the first half of Table 3. 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.

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 both can be reduced, considerably 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 the second half of Table 3.

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.

Table 4. %Retrieved of Relevant Stories (RS), Non-Relevant Stories (NRS), and Non-Stories (NS), number of duplicates (#Dup), and % Average and R-Precision after post-processing when filtering out postulated commercials after retrieval using `LIMSI`'s transcriptions.

	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	$\boxed{41.94}$

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 (both statistically significant at the 0.1% level), 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 suggest that a better AveP can be obtained when the commercial elimination is performed at the front-end of the system, although this is not statistically significant.

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, while also reducing the amount of data transcribed and associated computational time 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 five 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[11], and the results for 3, 5 and 10 seconds are given in Table 5.

Table 5. Effect of changing the gap required in the transcriptions to enforce a hard break during the post-processing stage, using the HTK-p2 transcriptions. (∞ ≡ not enforced). %Retrieved of RS, NRS and NS, number of duplicates and Average and R-Precision are given after post-processing.

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
∞	78.3	3.77	1.67	2422	41.44	42.01

These results show that although many merges have been prevented by enforcing hard breaks at gaps of five 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 that occur across a hard boundary no longer being incorrectly merged. However, some non-stories and non-relevant stories that 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 five seconds occurred in the transcriptions, and any extremely short windows (<8s or ≤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.

These results show that using the structural information derived from segmentation and commercial elimination increases the %retrieved for relevant stories while also reducing the %retrieved for non-stories and number of duplicates both before and after post-processing. However, although the R-precision

Table 6. Effect of enforcing hard breaks during window generation when a gap of >5s exists in the HTK-p2 transcriptions. %Retrieved of RS, NRS and NS, number of duplicates and Average and R-Precision are given.

Breaks	RS	NRS	NS	#Dup	AveP	R-P
Before post-proc.						
Y	96.4	39.98	18.80	717829	–	–
N	96.0	39.39	27.42	752913	–	–
After post-proc.						
Y	78.4	3.74	1.68	2707	41.47	$\boxed{41.98}$
N	78.3	3.43	2.39	3801	$\boxed{41.71}$	40.07

increases by 4.7% relative, the AveP decreases by 0.6% relative (statistically significant at the 0.1% level) with a corresponding increase of 9% relative in non-relevant story `%retrieved`. It therefore appears that using five 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),[12] so this was removed for subsequent experiments.

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., 1999b; Jourlin et al., 2000). 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.

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 also increase. 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 (not statistically significant) when including SPI is thought to be due to the inclusion of semantically related words adding considerably more non-relevant or non-stories than relevant stories during retrieval. This unexpected result needs further investigation, but in the meantime SPI was removed for subsequent experiments.[13]

Table 7. Effect of including semantic poset indexing (SPI) using the `HTK-p2` transcriptions. Results are given for %retrieved of RS, NRS and NS, number of duplicates and Average and R-Precision.

SPI	RS	NRS	NS	#Dup	AveP	R-P
Before post-proc.						
Y	96.0	39.39	27.42	752913	–	–
N	95.8	37.52	24.92	698461	–	–
After post-proc.						
Y	78.3	3.43	2.39	3801	41.71	40.07
N	79.2	3.43	2.40	3764	$\boxed{43.42}$	43.35

6.3.2. Blind Relevance Feedback (BRF). During blind relevance feedback a certain number of terms, t, are added to the query while making the assumption that the top r documents returned by running the retriever on the test collection are relevant. The TREC-7 SDR and adhoc collections (Voorhees and Harman, 1999) were used for the case where story boundaries are known for parameter-tuning experiments which resulted in the values of $t = 5$ and $r = 10$ being chosen. 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.

Table 8. Effect of altering the number of terms added, t, from assuming the top r windows retrieved are relevant, during the BRF stage. %Retrieved of RS, NRS and NS, number of duplicates and Average and R-Precision are given after post-processing for the `HTK-p2` transcriptions.

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

The results given in Table 8 show that including the BRF stage within the system improved the AveP by 4.6% relative (significant at the 1% level). 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, but *not significantly*, by adding more terms to the query during the feedback process.[14] 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 feedback on data that are not windowed.

6.3.3. Parallel Blind Relevance Feedback (PBRF). Parallel blind relevance feedback (PBRF) involves running BRF on a parallel collection of documents. This parallel corpus is generally text-based and thus not affected by transcription errors, has natural story boundaries and thus is not affected by windowing, and is larger than the test collection, thus offering more accurate statistics. For these experiments, the parallel collection consisted of 51,715 stories extracted from the L.A. Times, Washington Post and New York Times over a 6-month period that included the test data collection period.

Another experiment was conducted to see the effect of changing the t and r parameters for the PBRF 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 (significant at the 2.1% level).

6.3.4. Altering the Retrieval Parameters. The combined-weight formula used in the retriever, given by Eq. (1), contains two parameters, b and K, that 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 from development runs for the story-known case. 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 $b, \ldots$ 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"

Table 9. Effect of altering the number of terms added, t, from assuming the top r retrieved documents are relevant, during the PBRF stage. %Retrieved of RS, NRS and NS, number of duplicates and Average and R-Precision are given after post-processing for the `HTK-p2` transcriptions.

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

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.

The results, given in Table 10, confirm that R-P can be increased by 6.4% relative, with a corresponding improvement of 2.8% relative in AveP (significant at the 0.2% level), by setting $b = 0$ for the retrieval on the windowed test collection.[15] A further small (not statistically significant) increase in AveP can be obtained by increasing K.

6.4. *Optimising the Post-Processing*

The post-processing stage reduces the number of duplicate hits from the retriever output by *merging* some of the retrieved windows as described in Section 4.4. 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.

Table 10. Effect of altering the b and K parameters in the combined-weight formula when retrieving on the windowed test-collection. %Retrieved of RS, NRS and NS, number of duplicates and Average and R-Precision are given after post-processing for the `HTK-p2` transcriptions.

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 11. Effect of changing the merge length, T_m, during post-processing, with hard boundaries forced at gaps of more than five seconds in the transcriptions. %Retrieved of RS, NRS and NS, number of duplicates and Average and R-Precision are given after post-processing for the `HTK-p2` transcriptions.

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
∞	76.1	3.74	2.00	1677	46.41	$\boxed{48.05}$

6.4.1. Altering the Merge Length. In an attempt to eliminate duplicates, the post-processing stage 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. This generally meant that if one retrieved window started within one minute of the end of another retrieved window, they were viewed as originating from the same story. It was also hoped that enforcing hard breaks when gaps of over five seconds in the transcriptions occurred would help reduce the problem of over-merging.[16]

An experiment was conducted to find the effect on performance of varying T_m. The results are illustrated in Fig. 4 and summarised in Table 11. These results show that both Average and R-Precision increase monotonically towards an asymptote as the merge time is

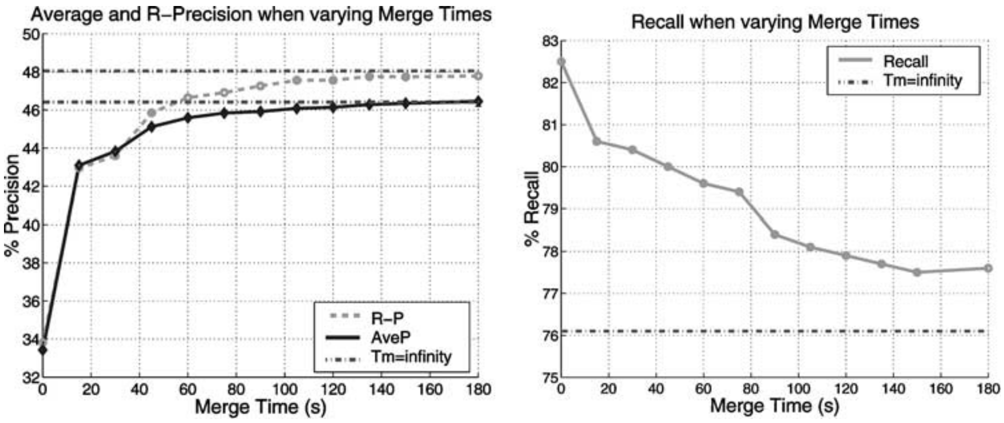

Figure 4. Effect on Average Precision, R-Precision and Recall of changing the merge time, used to decide if retrieved windows originated from the same story, during the post-processing stage.

Table 12. Effect of changing the merge length, T_m, during post-processing, with *no* hard boundaries enforced ($\infty \equiv$ taking only the top-ranking retrieved window per show). %Retrieved of RS, NRS and NS, number of duplicates and Average and R-Precision are given after post-processing for the HTK-p2 transcriptions.

T_m(s)	RS	NRS	NS	#Dup	AveP	R-P
15	80.5	3.40	1.51	6880	43.34	43.14
75	79.0	3.79	1.67	2123	46.20	47.30
180	75.6	3.94	1.73	397	46.66	48.06
210	75.0	3.96	1.73	242	46.71	48.42
240	74.5	3.97	1.72	143	$\boxed{46.80}$	$\boxed{48.46}$
270	74.0	3.98	1.71	96	46.76	48.42
∞	66.4	4.06	1.51	0	44.26	47.27

increased, suggesting that the "best" system would use a large merge time of around three minutes.[17] 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 $T_m = 180$s, 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.[18] 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.

The corresponding results when no hard breaks are forced in the merging procedure are given in Table 12. Again the AveP and R-P increase as the merge time is increased, with a slight increase over the corresponding case with boundaries enforced (significant at the 0.1% level for $T_m = 180$s). However, when the merge time is increased above four minutes, the AveP begins to drop, unlike when hard boundaries are enforced. This confirms that when reasonably accurate postulates for the commercials are available, enforcing hard breaks during post-processing helps guard against over-merging.

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 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 entering the post-processing stage for each cut-off level, are given in Table 13 using $T_m = 75$s.

Table 13. Effect of varying the low score threshold, θ_{LS}, of the retrieved windows. %Retrieved of RS, NRS and NS and number of duplicates are given before and after post-processing along with the number of retrieved windows and the final Average and R-Precision for the HTK-p2 transcriptions.

θ_{LS}	RS	NRS	NS	#Dup	# windows	AveP	R-P
Results before post-proc.							
0.1	96.5	44.20	30.41	871,428	1,448,864		
1	95.4	33.05	21.80	574,486	1,003,679		
5	89.8	10.38	5.58	127,084	259,030		
7	86.4	6.30	3.46	73,592	154,450		
10	82.1	3.22	1.66	38,499	80,196		
12	76.1	2.14	1.11	25,873	53,969		
Results after post-proc.							
0.1	79.4	3.64	1.83	3261		45.84	46.91
1	79.4	3.64	1.83	3256		45.84	46.91
5	78.4	3.43	1.73	5238		45.82	46.91
7	77.2	3.17	1.66	6445		45.77	46.91
10	77.0	2.35	1.15	11919		45.78	46.91
12	72.7	1.78	0.86	12591		45.63	46.69

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 (not statistically significant). For real systems, where speed of retrieval is important, the higher threshold thus should be used during post-processing.

7. Ongoing Work

Recent work in *document* expansion has shown that adding statistically related terms to the retrieval file from the parallel corpus prior to retrieval can improve performance for the story-known task (Singhal and Pereira, 1999). We successfully exploited this technique within the framework of the Probabilistic Model for our story-known system for the TREC-8 evaluation (Johnson et al., 2000) and are currently working on trying to port these ideas into the story-unknown task.

Finally, TREC-9 SDR will allow a text-based non-lexical information exchange file which can be used to store information automatically extracted from the audio that is not currently available in the recogniser transcriptions, e.g., speaker changes, music, audio repeats (Garofolo et al., 2000). It is hoped that combining this information with text-based cues could help locate story boundaries more accurately and further aid retrieval. Ongoing work covering these topics will be presented at TREC-9 (Johnson et al., 2001).

8. Conclusions

This paper has described a system for retrieving relevant portions of complete broadcast news shows when only the audio data are 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, while 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[19] 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 merge 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 had little effect on performance.

Experiments in retrieval showed that blind relevance feedback continued to be beneficial both using a parallel corpus and the test collection itself. However semantic poset indexing, which had been found useful in earlier tests on other data (Jourlin et al., 2000), was not helpful for this collection. Post-processing experiments showed precision could be increased at a cost to recall by performing more merging, while the speed of post-processing could be increased, with little loss in Average 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 46.8% on the TREC-8 story-unknown data set. This performance level is quite respectable when compared to the AveP of 56.7% achieved on the same test collection in the unrealistically favourable case where the manually-defined story boundaries used for scoring have been previously supplied (Johnson et al., 2000).

Appendix

Abbreviations

ABC	American Broadcasting Company
AHS	Arithmetic Harmonic Sphericity
ASR	Automatic Speech Recognition
AveP	(Mean) Average Precision
BRF	Blind Relevance Feedback
CFW	Collection Frequency Weight
CNN	Cable News Network
CUHTK	Cambridge University HTK

FFT	Fast Fourier Transform
HMM	Hidden Markov Model
HTK	HMM Toolkit
MLLR	Maximum Likelihood Linear Regression
NIST	National Institute for Standards and Technology
NRS	Non-Relevant Story
NS	Non-Story
PBRF	Parallel Blind Relevance Feedback
PCFW	Parallel Collection Frequency Weight
PLP	Perceptual Linear Prediction
POS	Part of Speech
Poset	Partially Ordered Set
PRI	Public Radio International
R-P	(Mean) R-Precision
RS	Relevant Story
RT	Real-Time
SDR	Spoken Document Retrieval
SPI	Semantic Poset Indexing
SU	Story-boundary Unknown
TDT	Topic Detection and Tracking
TREC	Text REtrieval Conference
VOA	Voice of America
WER	Word Error Rate

Notes

1. The complete query set and TREC-8 SDR evaluation specification are available through Garofolo et al. (1999a).
2. See Section 5 for a discussion of this scoring strategy.
3. One 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.
4. Sampling is usually done at 16 kHz in the front-end of an ASR system.
5. 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.
6. See also (Spärck Jones et al., 2000).
7. But not vice versa. Queries specifically about London are not concerned with general stories about England.
8. Although the manually generated story boundaries were used during the scoring procedure, they were not supplied to the SDR systems.
9. 6,294 non-stories multiplied by 50 queries.
10. 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`.
11. 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%.
12. 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.
13. Note that there is a complicated interaction between the use of SPI and other techniques such as blind relevance feedback.
14. Although again the parameter set which gave the best R-precision did not also give the best AveP.
15. The values used for the PBRF stage, which only uses the parallel corpus, were not changed.
16. See Section 6.2.1.
17. Using different merge times for different data sources will be investigated in the future.
18. The user's reaction to the presence of duplicates may also be influenced by the type of material being broadcast.
19. On a single processor of a dual processor Pentium III 550 MHz running Linux.

References

Abberley, D., Renals, S., Robinson, T., and Ellis, D. (2000). The THISL SDR system at TREC-8. In E.M. Voorhees and D.K. Harman (Eds.), *The Eighth Text REtrieval Conference (TREC-8)*. NIST Special Publication 500-246. Gaithersburg, MD; Department of Commerce, National Institute of Standards and Technology, pp. 699–706.

Bimbot, F. and Mathan, L. (1993). Text-free speaker recognition using an arithmetic harmonic sphericity measure. *Proc. Eurospeech'93*, Berlin, Germany, Vol. 1, pp. 169–172.

Cieri, C., Graff, D., Liberman, M., Martey, N., and Strassel, S. (1999). The TDT-2 text and speech corpus. *Proc. DARPA 1999 Broadcast News Workshop*, Herndon, VA, pp. 57–60.

Dharanipragada, S., Franz, M., and Roukos, S. (1999). Audio indexing for broadcast news. In E.M. Voorhees and D.K. Harman (Eds.), *The Seventh Text REtrieval Conference (TREC-7)*. NIST Special Publication 500–242. Gaithersburg, MD: Department of Commerce, National Institute of Standards and Technology, pp. 115–119.

Dharanipragada, S. and Roukos, S. (1997). Experimental results in audio indexing. *Proc. DARPA 1997 Speech Recognition Workshop*, Chantilly, VA.

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

Foote, J.T. (1997). Content-based retrieval of music and audio. *Multimedia Storage and Archiving Systems II, Proc. SPIE*, 3229, pp. 138–147.

Foote, J.T. (1999). An overview of audio information retrieval. *Multimedia Systems*, *7*(1):2–10.

Franz, M., McCarley, J., and Ward, R. (2000). Ad hoc, cross-language and spoken document information retrieval at IBM. In E.M. Voorhees and D.K. Harman (Eds.), *The Eighth Text REtrieval Conference (TREC-8)*. NIST Special Publication 500-246. Gaithersburg, MD: Department of Commerce, National Institute of Standards and Technology, pp. 391–398.

Gales, M.J.F. and Woodland, P.C. (1996). Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, *10*:249–264.

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 E.M. Voorhees and D.K. Harman (Eds.), *The Sixth Text REtrieval Conference (TREC-6)*. NIST Special Publication 500-240. Gaithersburg, MD: Department of Commerce, National Institute of Standards and Technology, pp. 83–92.

Garofolo, J.S., Auzanne, C.G.P., Voorhees, E.M., and Spärck Jones, K. (1999a). The 1999 TREC-8 spoken document retrieval (SDR) track evaluation specification. Available via http://www.nist.gov/speech/tests/sdr/sdr99/sdr99.htm.

Garofolo, J.S., Voorhees, E.M., Auzanne, C.G.P., Stanford, V.S., and Lund, B.A. (1999b). 1998 TREC-7 spoken document retrieval track overview and results. In E.M. Voorhees and D.K. Harman (Eds.), *The Seventh Text REtrieval Conference (TREC-7)*. NIST Special Publication 500-242. Gaithersburg, MD: Department of Commerce, National Institute of Standards and Technology, pp. 79–90.

Garofolo, J.S., Auzanne, C.G.P., and Voorhees, E.M. (2000). The TREC spoken document retrieval track: A success story. *Proc. Recherche d'Informations Assistee par Ordinateur (RIAO) 2000, Content-Based Multimedia Information Access*, Paris, France, Vol. 1, pp. 1–20.

Gauvain, J.-L., de Kercadio, Y., Lamel, L., and Adda, G. (2000). The LIMSI SDR system for TREC-8. In E.M. Voorhees and D.K. Harman (Eds.), The Eighth Text REtrieval Conference (*TREC-8*). NIST Special Publication 500–246. Gaithersburg, MD: Department of Commerce, National Institute of Standards and Technology, pp. 475–482.

Hain, T., Johnson, S.E., Tuerk, A., Woodland, P.C., and Young, S.J. (1998). Segment generation and clustering in the HTK broadcast news transcription system. *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Landsdowne, VA, pp. 133–137.

Hauptmann, A. and Witbrock, M. (1998). Story segmentation and detection of commercials in broadcast news video. *Proc. Advances in Digital Libraries (ADL '98)*, Santa Barbara, CA, pp. 168–179.

Johnson, S.E., Jourlin, P., Spärck Jones, K., and Woodland, P.C. (2000). Spoken document retrieval for TREC-8 at Cambridge University. In E.M. Voorhees and D.K. Harman (Eds.), The Eighth Text REtrieval Conference (TREC-8). NIST Special Publication 500-246. Gaithersburg, MD: Department of Commerce, National Institute of Standards and Technology, pp. 197–206.

Johnson, S.E., Jourlin, P., Spärck Jones, K., and Woodland, P.C. (2001). Spoken document retrieval for TREC-9 at Cambridge University. *Proc. 'The Ninth Text REtrieval Conference (TREC-9)*, to appear.

Johnson, S.E. and Woodland, P.C. (1998). Speaker clustering using direct maximisation of the MLLR-adapted likelihood. *Proc. 5th International Conference on Spoken Language Processing*, Sydney, Australia, Vol. 5, pp. 1775–1778.

Johnson, S.E. and Woodland, P.C. (2000). A method for direct audio search with applications to indexing and retrieval. *Proc. 2000 IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, Vol. 3, pp. 1427–1430.

Jourlin, P., Johnson, S.E., Spärck Jones, K., and Woodland, P.C. (1999a). General query expansion techniques for spoken document retrieval. *Proc. ESCA Workshop on Extracting Information from Spoken Audio*, Cambridge, England, pp. 8–13.

Jourlin, P., Johnson, S.E., Spärck Jones, K., and Woodland, P.C. (1999b). Improving retrieval on imperfect speech transcriptions. *Proc. 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, pp. 283–284.

Jourlin, P., Johnson, S.E., Spärck Jones, K., and Woodland, P.C. (2000). Spoken document representations for probabilistic retrieval. *Speech Communication*, *32*(1):21–36.

Kashino, K., Smith, G., and Murase, H. (1999). Time-series active search for quick retrieval of audio and video. *Proc. 1999 IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, AZ, pp. 2993–2996.

Ng, C., Wilkinson, R., and Zobel, J. (2000). Experiments in spoken document retrieval using phoneme n-grams. *Speech Communication*, *32*(1):61–77.

Odell, J.J., Woodland, P.C., and Hain, T. (1999). The CUHTK-entropic 10xRT broadcast news transcription system. *Proc. DARPA 1999 Broadcast News Workshop*, Herndon, VA, pp. 271–275.

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

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. *Proc. Eurospeech 99*,Budapest, Hungary, pp. 1267–1270.

Singhal, A. and Pereira, F. (1999). Document expansion for speech retrieval. *Proc. 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, pp. 34–41.

Spärck Jones, K., Walker, S., and Robertson, S.E. (2000). A probabilistic model of information retrieval: Development and comparative experiments, Parts 1 and 2. *Information Processing and Management*, *36*(6):779–840.

van Mulbregt, P., Carp, I., Gillick, L., Lowe, S., and Yamron, J. (1999). Segmentation of automatically transcribed broadcast news text. *Proc. DARPA 1999 Broadcast News Workshop*, Herndon, VA, pp. 77–80.

van Rijsbergen, C.J. (1979).*Information Retrieval*, 2nd ed. Stoneham: MA Butterworths.

Voorhees, E.M. and Harman, D.K. (1999). Overview of the seventh text REtrieval conference (TREC-7). In E.M. Voorhees and D.K. Harman (Eds.), *The Seventh Text REtrieval Conference (TREC-7)*. NIST Special Publication 500-242. Gaithersburg, MD: Department of Commerce, National Institute of Standards and Technology, pp. 1–24.

Wold, E., Blum, T., Keslar, D., and Wheaton, J. (1996). Content-based classification, search and retrieval of audio. *IEEE Multimedia*, Fall 1996:27–36.

Woodland, P.C., Gales, M.J.F., Pye, D., and Young, S.J. (1997). The development of the 1996 HTK broadcast news transcription system. *Proc. DARPA 1997 Speech Recognition Workshop*, Chantilly, VA, pp. 73–78.