AN OVERVIEW OF THE AT&T SPOKEN DOCUMENT RETRIEVAL

John Choi, Don Hindle, Julia Hirschberg, Ivan Magrin-Chagnolleau,
Christine Nakatani, Fernando Pereira, Amit Singhal, Steve Whittaker

AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07932-0971

ABSTRACT

We present an overview of a spoken document retrieval system developed at AT&T Labs-Research for the HUB4 Broadcast News corpus. This overview includes a description of the intonational phrase boundary detection, classification, speech recognition, information retrieval and user interface components of the system, along with updated system assessments based on the 49-query task defined for the TREC-6 SDR track. Results from a comparative ranking study, based on queries taken from AP Newswire headlines from the same time period that the Broadcast News corpus was recorded, are presented. For the AP task, retrieval accuracy is assessed by comparing the documents retrieved from ASR generated transcriptions with those from human generated transcriptions.

1. INTRODUCTION

This paper presents an overview of a spoken document retrieval and browsing system developed at AT&T Labs Research. The system was designed for the TREC-6 SDR track [23], which involved a retrieval task consisting of 49 known-item queries submitted over approximately 47 hours of speech from the HUB4 Broadcast News corpus [5]. Automatic speech recognition is used to generate textual transcripts of the HUB4 corpus. Text-based information retrieval techniques are then applied. In addition to supporting research on information retrieval strategies for machine generated speech transcripts, the system is also a testbed for user-interface experiments on intelligent presentation of speech documents to users.

Previous work on spoken document retrieval includes a Video Mail Retrieval system [7,8], radio news broadcast retrieval using subword units [16], a retrieval system for a digital video library [25], a system for Swiss radio news [24], and the systems developed for the TREC-6 SDR track [23], *inter alia*.

2. SYSTEM OVERVIEW

An overview of the system architecture is provided in Figure 1. Speech documents, whose boundaries were prespecified for the HUB4 retrieval task, are initially processed by an intonational phrase boundary detector. The phrase defined by the boundary detector is then submitted for classification. The classifier provides a hypothesis about the channel conditions in the given phrase. Based on this hypothesis, one of several acoustic models is selected for use in the recognizer. Having generated the transcripts for the speech corpus, an information-retrieval engine indexes the transcripts for retrieval. Boundary detection, classification, recognition and indexing are all conducted off-line. The user-interface currently supports real-time query-submission and retrieval, making calls to the information-retrieval engine which returns a ranked list of hypothesized relevant documents based on the machine generated transcripts. Each of the system components is described in more detail below.

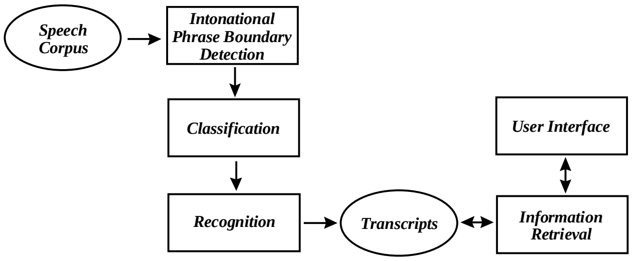


Figure 1: Overview of the spoken document system architecture

2.1. Intonational Phrase Boundary Detection

The speech documents are initially passed through an intonational phrase boundary detector [6]. The detector generates a binary classification for every 10 msec of the speech stream as either (i) occurring within an intonational phrase or (ii) occurring in the break between two intonational phrases. This decision is based on models generated by classification and regression tree techniques [1], which were trained on acoustic vectors consisting of the following parameters: fundamental frequency, RMS energy and autocorrelation-peak. The training corpus consisted of the Boston Directions Corpus [15] which was prosodically transcribed using ToBI conventions [18].

Contiguous frames classified as occurring in the break between intonational phrases are used to define phrase boundaries, with the constraint that an intonational phrase be at least 1 second in duration and no greater than 20 seconds in duration. Selection is biased toward longer phrases, so that if, e.g., a phrase boundary is detected at 5 seconds and another at 19 seconds, the boundary at 5 seconds is ignored in favor of a single larger phrase 19 seconds in length. This boundary detection scheme segmented the speech corpus into 10,480 phrases averaging 14.78 seconds in duration.

The boundary detection serves two purposes. First, the intonational phrases serve as domains over which we can apply one of several acoustic models for recognition. This is discussed in more detail in section 2.2. Second, the frame classifications are also used to identify suitable browsing-units for the user interface of the spoken document retrieval system. This is based on the assumption that presentation of prosodically well-formed segments of speech to the user would be preferable to fixed-size segments which might begin and terminate in the middle of an utterance.

2.2. Classifier

To select the best acoustic model to apply to a given recognition unit, a classifier [13] is used which assigns log-likelihood scores associated with each of four categories, reflecting a four-way partitioning of the training corpus into "high-fidelity", "medium-fidelity", "low-fidelity" and "noise" categories. The high-fidelity partition consists of wideband (0-8kHz) speech recorded in the studio with no background noise. The medium partition also consists of wideband (0-8kHz) speech, but recorded in other recording environments, including field conditions, with no background noise. The low-fidelity partition consists of narrowband speech (0-4kHz) recorded from telephone interviews, again with no background noise. All speech recorded in the presence of background noise is consigned to the noise partition. Partitioning was based on labels provided with the HUB4 training corpus. The purpose of this partitioning was to minimize observed variability in channel conditions when training different acoustic models.

	D_{HIGH}	D_{MED}	D_{LOW}	D_{NOISE}	CORRECT
M_{HIGH}	123	38	0	15	69.88%
M_{MED}	3	4	0	9	25.00%
M_{LOW}	10	12	85	0	79.44%
M_{NOISE}	44	31	1	167	68.72%

Table 1: Confusion-matrix for the classification models; correct classifications are represented on the diagonal. Each row provides counts of positive classifications by model M_C for each data partition D_C. Accuracy for each model M_C is presented as percent correct.

Classification is based on full covariance Gaussian mixture models [22], initialized using vector quantization [10], and trained using the Expectation-Maximization algorithm [3]. The training input consists of 31-dimensional vectors of filter-bank coefficients in dB units, derived from Hamming windowed frames of 20 msec with a frame advance rate of 10 msec. The filter-bank coefficients are computed by taking the base 10 logarithm from short term power spectra in the 0-8kHz band from a mel-scaled bank of filters.

Log-likelihood scores are computed for each of the four classification models. Recognition then proceeds with the acoustic model associated with the classification that provides the highest log-likelihood score. When tested on the three hour HUB4 development test partition, overall accuracy for the classifier was assessed at 69.91%, normalizing for the number of observations in each category. Within category accuracy scores, along with a confusion matrix profiling errors, are reported in Table 1.

2.3. Recognizer

The recognizer incorporates a standard time-synchronous beam search algorithm with continuous density, three-state, left-to-right, context-dependent hidden Markov phone models. The models connecting phone HMMs to word sequences are implemented in the general framework of weighted finite-state transducers [14,17]. The probabilities defining the transduction from context-dependent phone sequences to word sequences are estimated on word level grapheme-to-phone mappings generated by a set of heuristics. These mappings are typically one-to-one, although multiple pronunciations are defined for less than 5% of the 29K words considered in the training phase.

Recognition hypotheses are output in the form of word lattices which are derived from model lattices by transducer composition [14]. Given a phone model p for transition a from state s to state s' matching the input from time t to time t', the corresponding model lattice arc goes from node (s,t) to node (s',t'). In the implementation, for any such t', only the best t is recorded for reasons of run-time efficiency and lattice size. The resulting lattice reduction does not seem to have negative consequences for recognition accuracy.

Acoustic observations serving as input to the HMMs consist of 39-dimensional vectors taken from 20 msec analysis frames. The frame advance rate is 10 msec. Each acoustic vector contains the first 13 normalized mel-frequency cepstral coefficients, along with their first and second time derivatives.

Two sets of acoustic models have been trained. The first set consists of four acoustic models – $A1_{HIGH}$, $A1_{MED}$, $A1_{LOW}$ and $A1_{NOISE}$ – based on the four-way partitioning of the training corpus described in section 2.2. Each of these models was bootstrapped from a single model trained on the channel 1 WSJ training corpus. The second set of models consists of two acoustic models – $A2_{WB}$ and $A2_{NB}$ – trained on wideband speech (the union of the "high-fidelity", "medium-fidelity" and "noise" partitions), and narrowband speech (the "low-fidelity" partition), respectively. The $A2_{WB}$ model was bootstrapped from $A1_{HIGH}$. In addition, the training corpus for $A2_{WB}$ was expanded to include the “spontaneous speech” partition of the channel 2 WSJ training corpus. This additional training data was weighted by a factor of 0.1 in calculating means and variances; the HUB4 training data was weighted by a factor of 0.9. The WSJ data was not included in defining the context-dependency models. The $A2_{NB}$ model was bootstrapped from a narrowband model trained on the SWITCHBOARD corpus [4].

Training iterations in both sets consisted of eigenvector rotations to decorrelate the training data [11], k-means clustering, normalization of means and variances based on maximum-likelihood, and Viterbi alignment to re-segment the data. The output probability distributions in the HMMs consist of a weighted mixture of Gaussians with diagonal covariance. Each mixture in the *A1* set of models contains at most 8 components. In the *A2* set of models, each mixture contains at most 12 components.

2.4. Language Models

Language modeling was based on a 116 million word corpus consisting of text designated for SDR language modeling and the transcriptions from the training partition. The 20K most frequent words in the training corpus comprise our lexicon. The least frequent of the 20K words appear 121 times in the training corpus. In addition, two pseudo words – *pause* and *unknown* – were added to the lexicon. *pause* is freely insertable at any point in recognition, and *unknown* is used for words outside the vocabulary.

Two Markov language models were trained. A standard Katz [9] backoff bigram model was constructed from the 6.1 million bigrams observed in the training corpus. When tested on the retrieval test corpus, this model exhibits an out-of-vocabulary rate of 2.2% and perplexity of 200.

A backoff trigram model was also constructed based on the 9.4 million trigrams observed in the training corpus. This model showed an out-of-vocabulary rate of 2.2% and a perplexity of 144 on the three hour development test corpus. From this model, a more compact trigram language model was constructed following the procedures described in [20]. In particular, trigrams and bigrams were discarded from the model in cases where the difference between the model prediction and the backed-off prediction is less than a threshold *T*:

$$f * (P_O - P_B) < T$$

where *f* is the observed *n*-gram frequency, P_O is the *n*-gram prediction and P_B is the backed-off *(n-1)*-gram prediction. In our case, a threshold of 20 was used. This technique reduces the number of trigrams in the model by 85% and the number of bigrams in the model by 83%, with a concomitant increase in perplexity of 18% to 169. Compared to the bigram model, this compacted trigram model has 16% lower perplexity and is 60% smaller.

2.5. Information Retrieval

We use the SMART retrieval system which is a text processing system based on the vector space model [19,2]. SMART automatically generates weighted vectors for any given text using the following indexing scheme:

- Tokenization: The text is first tokenized into individual words and other tokens.
- Stop word removal: Common functions words (like *the*, *of*, *an*, …), also called stop words, are removed from this list of tokens. The SMART system uses a predefined list of 571 stop words.
- Stemming: Various morphological variants of a word are normalized to the same *stem* [12]. Usually simple rules for suffix stripping are used in this process.
- Weighting: The term (word) vector, thus created for a text, is weighted using *tf*, *idf*, and length normalization considerations.

We use the *Lnu* term weighting scheme to assign weights to the terms of a document [21]:

$$\frac{(1 + ln(tf)) / (1 + ln(average\ tf))}{0.8 * pivot + 0.2 * (\#\ of\ unique\ terms)}$$

where *tf* is the number of times a term occurs in the text, and average *tf* is the average of the *tf*s of all the terms in a document. The average number of unique terms in a document (computed *across the entire collection*) is used as the *pivot*. The user queries are also indexed using the above steps, and are weighted using *ltn* weights [21]:

$$(1 + ln(tf)) * idf$$

where *tf* is once again the frequency of a word in the query and *idf* is *ln*(*N*/*df*) (*N* is the total number of documents in the collection, and *df* is the number of documents that contain the word).

If *Q* is the query vector and D_i is the vector representation for document-*i*, a numerical (inner-product) similarity between the query and the document is computed as

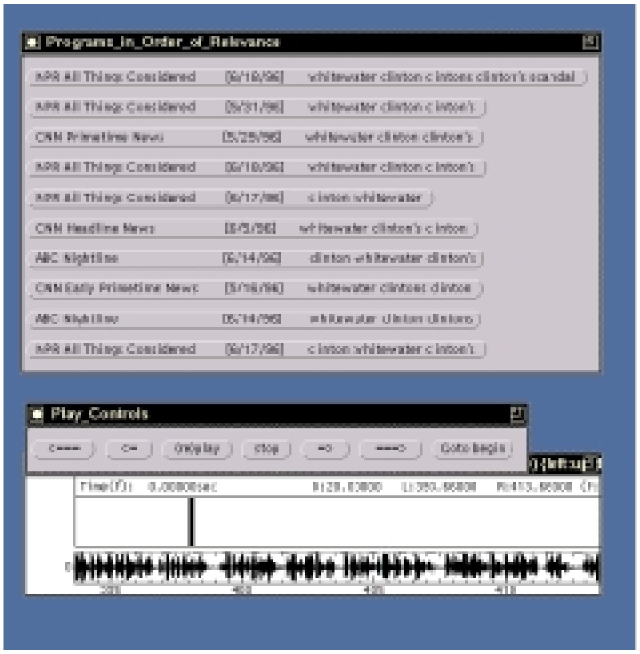
$$Sim(Q, D_i) = \sum_{common\ terms\ t_j} q_j * d_{ij}$$

where t_j is a term present in both the query and the document, q_j is the weight of term t_j in the query, and d_{ij} is its weight in document-*i*. The documents in the information base are ranked by their decreasing similarity to the query and are presented to the user in this order.

2.6. User Interface

The user interface design is based on extensive user testing of a simple browser on a voicemail database and consists of three main components: an Programs Overview window, a Speech

Feedback window and a Player window. These are shown in Figure 1 for the query “whitewater clinton scandal”. Queries are submitted textually at a shell prompt.



NPR All Things Considered	[6/18/96]	whitewater clinton clintons clinton's scandal
NPR All Things Considered	[5/31/96]	whitewater clinton clinton's
CNN Primetime News	[5/29/96]	whitewater clinton clinton's
NPR All Things Considered	[6/18/96]	whitewater clinton clinton's
NPR All Things Considered	[6/17/96]	clinton whitewater
CNN Headline News	[6/5/96]	whitewater clinton's clinton
ABC Nightline	[6/14/96]	clinton whitewater clinton's
CNN Early Primetime News	[5/16/96]	whitewater clintons clinton
ABC Nightline	[6/14/96]	whitewater clinton clintons
NPR All Things Considered	[6/17/96]	clinton whitewater clinton's

Figure 2: The graphical user interface for spoken document retrieval and browsing.

The Programs Overview window presents the speech search results as a relevance ranked list of stories, named by the news program in which they occur. The top 10 most relevant stories are displayed. For each story, the program title (e.g., “NPR All Things Considered”), the date of the program and all instances of keywords in the story that matched the query are displayed. Clicking on one of the program-story buttons loads the corresponding speech into the Speech Feedback window, along with a time-aligned cursor which shows the location of the story that contains the query terms. The Player window provides controls for navigation and play within the program displayed in the Speech Feedback window and includes the following: a play button which plays from the point selected in the Speech Feedback window; a stop-play button; a move-to-beginning button; skip-forward buttons which skip forward an intonational phrase or paratone; skip-back buttons which skip backwards an intonational phrase or paratone. A paratone is a unit larger than the intonational phrase whose pause boundaries are typically longer in duration than those that delimit intonational phrases. Like the intonational phrases, the paratones are automatically detected labeled using the boundary-detection model described in section 2.1.

3. SYSTEM EVALUATION

We evaluate the retrieval effectiveness of our system using two different tasks: the TREC-6 SDR task and an AP Newswire headlines task. Both tasks, described in sections 3.1 and 3.2, involve two different sets of transcriptions. The first set (*Trans1*) was generated by a recognizer using the *A1* model set and the bigram language model. Subsequently, another set of transcriptions (*Trans2*) was generated by using the $A2_{WB}$ model, the $A1_{LOW}$ model and the compact trigram language model.[1] The word error rates associated with the two recognizers on the three hour development test partition and the 47 hour test partition are presented in Table 2.

	Word Error Rate	
	Recognizer 1 ***(Trans1)***	**Recognizer 2** ***(Trans2)***
Devtest (3 hours)	50.50%	36.64%
Test (47 hours)	42.70%	30.09%

Table 2: Word error rates associated with the two different recognizers on both the three hour development-test partition and the 47 hour test partition.

3.1. TREC-6 SDR Task

The first benchmark we use is the TREC-6 SDR task. For this task, each broadcast-show has been manually divided into several documents. There are 49 user queries with each query having a unique “answer document” in the collection of 1452 documents (as indexed by SMART). Two queries, numbered SDR43 and SDR48, have two answer documents in the collection. The aim of the retrieval system is to rank the answer document for a query as high up in rank as possible (in response to the query). This task has also been referred to as “known item searching”. For the two queries with more than one answer document, the rank of the answer document with a better rank (lower absolute rank) is used in the evaluations.

It is generally accepted that this is a relatively easy task compared to other benchmark retrieval tasks. For most of the queries, a typical IR system running on the human transcription of the speech consistently retrieves the answer document within the top few ranks. This pattern holds true even when retrieval is done using machine generated transcription, albeit less accurately. This ease of retrieval makes it difficult to conduct rigorous comparisons of multiple retrieval approaches. Given the ease of retrieval from this collection, using any single evaluation measure does not exemplify meaningful differences between two retrievals. For this reason, we use *five* different evaluation measures in this study to compare the retrieval effectiveness of retrieval from various transcriptions.

Evaluation Measures. The evaluation measure that we use are:

- **E1**: *Number of queries for which the answer document is ranked 1*. This indicates “perfect” retrieval. High values for this evaluation measure are desired as they

[1] The training of the $A2_{NB}$ model was incomplete at the time the second set of transcripts was generated.

indicate that a retrieval system is doing perfect retrieval for many queries.

- **E2**: *Number of queries for which the answer document is ranked within the top 5*. Perfect retrieval might be too strict an evaluation measure. From a user's perspective, if an answer document is retrieved at rank 2, then the system is still doing a good job. This evaluation measure credits a system if the answer is found within the top five speech documents retrieved.
- **E3**: *Mean answer rank*. This measure uses the answer ranks for all the queries. A system that has a low mean answer rank is better.
- **E4**: *Mean answer rank, after removing outliers*. Since there are only 49 queries, if one of the answers is retrieved at a poor rank, the *mean answer rank* measure for the entire system suffers noticeably. For example. If one of the answers is ranked 400 for a system, the *mean answer rank* of the system falls by almost 8. For this reason, we allow each system to ignore one to two of its worst queries for which the answer is retrieved at a very poor rank when computing the *mean answer rank*.
- **E5**: *Mean reciprocal rank*. This is the known item search variant of the non-interpolated average precision score. Its value is computed as:

$$\frac{\sum_{i=1}^{49} \left(1 \,/\, \textit{answer rank for query}_i \right)}{49}$$

Higher values of this measure are better. We must point out that this measure is heavily governed an answers rank. For example, if the answer is at rank 1, this measure assigns a credit of 1.0 to the system; whereas if the answer is ranked 2, the credit assigned is just half. It is unclear if, from a user's perspective, ranking an answer at rank 1 is 100% better than ranking the answer at rank 2. On the other hand, this measure makes very little distinction between an answer ranked at low ranks (1/100 is not much different from 1/300 in absolute terms).

If one retrieval run consistently outperforms another on most of the above measures, then, despite the shortcomings of this task, we have strong reason to believe that the better retrieval run is indeed superior.

Results. We performed retrieval using the 49 queries on the first transcription set *Trans1*, the second transcription set *Trans2* and the human transcriptions *Human*. The results are shown in Table 3. The -1 in the **E4** row indicates that each run was allowed to skip its one worst query. Similarly the -2 in the very next row indicates that the two worst queries were skipped for each run. The numbers in the parentheses represent improvement in retrieval (in the case of positive numbers) as compared to the baseline performance associated with *Trans1*.

Table 3 shows that retrieval on *Trans2* outperforms retrieval on *Trans1* for each and every evaluation measure. Using *Trans2*, e.g., 32 out of the 49 queries get their answer document retrieved at rank 1, as opposed to just 29 for *Trans1*. Similar improvements are obtained for other measures. These results are reassuring as they indicate that when a speech recognizer improves, a speech retrieval system should improve as well. As expected, retrieval from recognizer generated transcripts is still poorer than retrieval from human generated transcripts.

	Trans1	**Trans2**	**Human**
E1	29	32	35
	--	(+3)	(+6)
E2	37	39	45
	--	(+2)	(+8)
E3	25.80	10.39	7.39
	--	(+15.41)	(+18.41)
E4	9.83	6.10	2.63
(-1)	--	(+3.73)	(+7.20)
E4	6.06	5.21	2.00
(-2)	--	(+0.81)	(+4.06)
E5	0.6703	0.7149	0.8020
	--	(+6.65%)	(+19.65%)

Table 3: Retrieval results for TREC-6 SDR task.

3.2. AP Newswire Queries

To exercise the system and expand our evaluation protocol, we devised another task to compare the retrieval from the three transcription sets. We selected 94 AP Newswire headlines, published between May 10, 1996 and June 20, 1996, the same period from which the broadcast news shows were recorded. We use these headlines, which are typically short, as potential user queries for retrieval. However, since we do not have relevance assessments for these headlines, we cannot use the standard IR evaluation measures (such as average precision) to evaluate our retrieval. Since most speech retrieval work aspires to achieve results comparable to the results of doing retrieval from the human transcription of speech, in this task, we use the ranking for the human transcriptions as a "gold standard" and evaluate how close retrieval from speech comes to this gold standard.

Evaluation Measures and Results. As a first test, we compare the average number of common documents retrieved per query (within the top *K* ranks) from the human transcriptions and the machine transcriptions. A higher average number of common documents would signify a higher correlation between retrievals from the *Human* set of transcriptions and those from the *Trans1* and *Trans2* sets. This, in turn, would suggest that the corresponding speech retrieval run is more effective.

We test this measure for the *K* values 1, 5, 10, 20, 30, 50, and 100. The results are shown in Figure 2. We observe that retrieval from *Trans2* is more closely correlated to the human transcription than retrieval from *Trans1* for all *K* ranks on this evaluation measure. On average, *Trans2* retrieves about 75% more in common to *Human* whereas *Trans1* only retrieves about 70%.

We use the following additional test to compare rankings from the human and the machine transcriptions. We assume that the top *K* documents as retrieved from the human transcriptions are relevant for the corresponding query. Given this assumption, we can compute the average precision for retrieval from the machine transcriptions. Note that by this assumption, retrieval from human transcriptions always has an average precision of 100%.

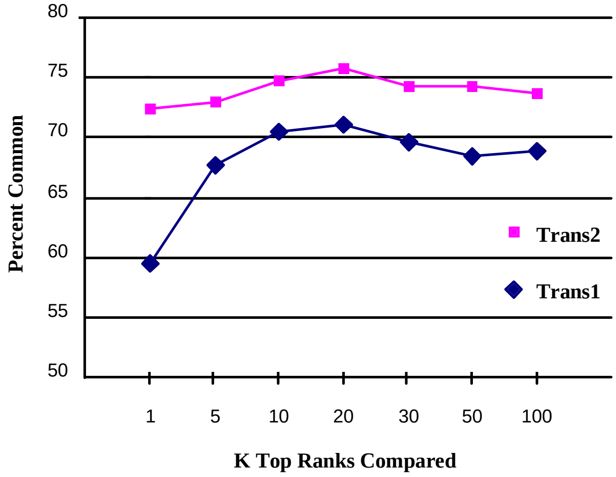


Figure 3: Percent of common documents in top *K* ranks.

Once again, we test this measure for the *K* values 1, 5, 10, 20, 30, 50, and 100. The results, shown in Figure 3, again confirm that retrieval from *Trans2* is more effective for all *K* ranks, as measured by average precision.

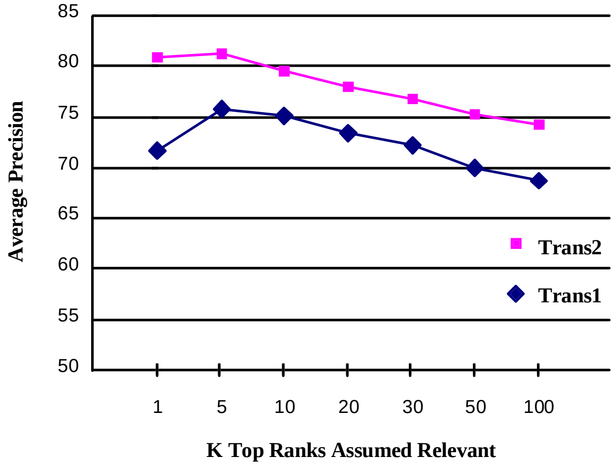


Figure 4: Average precision for retrieval from machine generated transcriptions if the top K ranks of human generated transcriptions are assumed relevant.

4. CONCLUSIONS

We have provided an overview of a spoken document retrieval system designed for the HUB4 corpus and presented experimental results which provide some metric of retrieval efficiency. Two general conclusions can be drawn from these results. First, spoken document retrieval is a tractable problem. As shown in Figures 3 and 4, retrieval from automatic transcriptions is about 70-80% as effective as retrieval from human transcriptions. This number is in agreement with other studies that have compared the retrieval effectiveness on both automatically generated transcriptions and human generated transcriptions [8,25].

The second conclusion we draw is simply that better recognition contributes to better retrieval. While this seems intuitively transparent, we are pleased that our empirical findings support this hypothesis. The relatively small size of the HUB4 corpus for information retrieval purposes initially forced us to interpret our early results with some skepticism. However, with the introduction of the various other evaluation measures outlined in this paper, and the consistency with which the trends were evident along all these measures, we find that speech retrieval systems do stand to gain from better recognition.

We are encouraged by these results and continue to explore various approaches to increase retrieval efficiency and to resolve the ‘out-of-vocabulary’ problem, which was not addressed in this paper. Preliminary results from work with word lattices which provide multiple recognition hypotheses suggest that the use of lattices increases word recall, albeit with noticeable cost in word precision, and that this increase results in better retrieval. We have also conducted experiments exploring the use of subword units and hope to expand this line of research in the coming months. The system is also being utilized to explore different user interfaces to maximize the utility of spoken document retrieval for browsing broadcast news recordings as well as other application domains.

ACKNOWLEDGEMENTS

We wish to thank Andrej Ljolje, Mehryar Mohri and Michael Riley for their help in the development of the recognizer.

REFERENCES

1. Breiman, L, Friedman, J. H., Olshen, R. A., and Stone, C. J., *Classification and Regression Trees*, Chapman and Hall, 1984.
2. Buckley, C., “Implementation of the SMART information retrieval system”, *Technical Report TR85-686*, Department of Computer Science, Cornell University, Ithaca, NY, 1985.
3. Dempster, A. P., Laird, N. M., and Rubin, D. B.,

"Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society*, B39(1), 1977, pp. 1-38.

4. Godfrey, J. J., Holliman, E. C., and McDaniel, J., "SWITCHBOARD: Telephone speech corpus for research and development", *Proceedings of the ICASSP 92*, 1992, pp. 517-520.

5. Graff, D., Wu, Z., MacIntyre, R., and Liberman, M., "The 1996 broadcast news speech and language-model corpus", *Proceedings of the 1997 DARPA Speech Recognition Workshop*, 1997.

6. Hirschberg, J., and Nakatani, C., "Using machine learning to identify intonational segments", *Proceedings of the AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, 1998, (forthcoming).

7. Jones, G. J. F., Foote, J. T., Sparck-Jones, K., and Young, S. J., "Video mail retrieval: The effect of word spotting accuracy on precision", *Proceedings of ICASSP 95*, Vol. 1, 1995, pp. 309-312.

8. Jones, G. J. F., Foote, J. T., Sparck-Jones, K., and Young, S. J., "Retrieving spoken documents by combining multiple index sources", *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996, pp. 30-38.

9. Katz, S. M., "Estimation of probabilities from sparse data from the language model component of a speech recognizer", *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1987, pp. 400-401.

10. Linde, Y., Buzo, A., and Gray, R. M., "An algorithm for vector quantization design", *IEEE Transactions on Communications*, 28(1), 1980, pp. 84-95.

11. Ljolje, A., "The importance of cepstral parameter correlations in speech recognition", *Computer Speech and Language*, 1994.

12. Lovins, J. B., "Development of a stemming algorithm", *Mechanical Translation and Computational Linguistics*, 1-2(11), 1968, pp. 11-31.

13. Magrin-Chagnolleau, I., Parthasanathy, S., and Rosenberg, A., "Automatic labeling of broadcast news into different sound classes using gaussian mixture models", manuscript in preparation.

14. Mohri, M., Riley, M., Hindle, D., Ljolje, A., and Pereira, F. C. N., "Full expansion of context-dependent networks in large vocabulary speech recognition", *Proceedings of ICASSP 98*, 1998, forthcoming.

15. Nakatani, C., Grosz, B. and Hirschberg, J., ``Discourse structure in spoken language: studies on speech corpora'', *Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, 1995.

16. Ng, K., and Zue, V., "Subword unit representations for spoken document retrieval", *Proceedings of Eurospeech 97*, 1997, pp. 1607-1610.

17. Pereira, F., and Riley, M., "Speech recognition by composition of weighted finite automata", in Roche, E., and Schabes, Y., (eds), *Finite-State Language Processing*, MIT Press, Cambridge, 1997, pp. 431-453.

18. Pitrelli, J., Beckman, M., and Hirschberg, J., "Evaluation of prosodic transcription labeling reliability in the ToBI framework", *Proceedings of the 3rd International Conference on Spoken Language Processing*, Vol. 2, 1994, pp. 123-126.

19. Salton, G., (ed.), *The SMART Retrieval System - Experiments in Automatic Document Retrieval*. Prentice Hall, Inc., Englewood Cliffs, NJ, 1971.

20. Seymore, K., and Rosenfeld, R., "Scalable backoff language models", *Proceedings of the Fourth International Conference on Spoken Language Processing*, 1996.

21. Singhal, A., Buckley, C., and Mitra, M., "Pivoted document length normalization", *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996, pp. 21-29.

22. Titterington, D. M., Smith, A. F. M., and Makov, U. E., *Statistical Analysis of Finite Mixture Distributions*, John Wiley and Sons, 1985.

23. Vorhees, E. M., and Harman, D. K., "Overview of the sixth Text REtrieval Conference (TREC-6)", in Vorhees, E. M., and Karman, D. K. (eds.), *Proceedings of the Sixth Text Retrieval Conference (TREC-6)*, 1998, forthcoming.

24. Wechsler, M., and Schauble, P., "Indexing methods for a speech retrieval system", in van Rijsbergen, C. J. (ed.), *Proceedings of the MIRO Workshop*, 1995.

25. Witbrock, M. J., and Hauptmann, A. G., "Using words and phonetic strings for efficient information retrieval from imperfectly transcribed spoken documents", *Proceedings of the 2nd ACM International Conference on Digital Libraries*, 1997, pp. 30-35.