Audio-Indexing For Broadcast News

Satya Dharanipragada, Martin Franz, Salim Roukos

IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

ABSTRACT

In this paper we describe the IBM Audio-Indexing System which is a combination of a large vocabulary speech recognizer and a text-based information retrieval system. Our speech recognizer was used to produce the baseline transcripts for the NIST SDR97 evaluation. We report the performance of the system on the SDR-97 "known item retrieval" task and on a more pertinent TREC-style Audio-Indexing task.

1. Introduction

The goal of an audio-indexing system is to provide the capability of searching and browsing through audio content. The system is formed by integrating information retrieval methods with large vocabulary continuous speech recognition. A large vocabulary continuous speech recognition system is used to to produce time aligned transcripts of the speech. Information retrieval techniques are then employed on these recognized transcripts to identify locations in the text that are relevant to the search request. These locations with time alignments then specify regions of the speech that are relevant for the request.

In this paper we give a description of the speech recognition and information retrieval systems that constitute our Audio Indexing System and report the performance of the system on the SDR97 "known item retrieval" task and on a more pertinent TREC-style Audio-Indexing task.

2. System Description

Our current Audio-Indexing system consists of two components: (1) A large vocabulary continuous speech recognition system, and (2) a text-based information retrieval system. Below we give a brief description of these two components.

2.1. Speech Recognition System

The recognition system is based on the large vocabulary continuous speech recognition system described in [1, 2, 3]. The system uses acoustic models for subphonetic units with context-dependent tying. The instances of context-dependent sub-phone classes are identified by growing a decision tree from the available training data and specifying the terminal nodes of the tree as the relevant instances of these classes. The acoustic feature vectors that characterize the training data at the leaves are modeled by a mixture of Gaussian pdf's, with diagonal covariance matrices. Each leaf of the decision tree is modeled by a 1-state Hidden Markov Model with a self loop and a forward transition. The IBM system expresses the output distributions on the state transitions in terms of the rank of the leaf instead of in terms of the feature vector and the mixture of Gaussian pdf's modeling the training data at the leaf. The rank of a leaf is obtained by computing the log-likelihood of the acoustic vector using the model at each leaf, and then ranking the leaves on the basis of their log-likelihoods.

For this SDR track evaluation we, on purpose, trained our recognizer on data that is different from the SDR (HUB4) training data. This enabled us to judge the performance of our system under mismatch conditions where one wishes to retrieve, as is often the case in practice, spoken documents that come from a domain different from the domain that the speech recognizer is trained on. The acoustic space is parameterized by 60 dimensional feature vectors which are obtained by performing a Linear Discriminant Analysis on a 9 frame window of 24 dimensional cepstral coefficients vectors. The decision tree for identifying the context-dependent sub-phone classes was grown using the WSJ data. The decision tree has around 6000 leaves. An initial set of Gaussians, approximately 35,000 in number, were also trained using the WSJ data. This was our initial system, which we will refer to as **SR-0**. In order to adapt our system to the broadcast domain we ran a MAP adaptation using approximately 1100 studio quality sentences from the 1994 HUB4 (NPR Market Place) data. This system which we will refer to as **SR-1** was used to produce the baseline transcripts for the SDR97 track.

For the language model we use a deleted interpolation trigram model which was also trained on the WSJ corpus with a 64K cased vocabulary. The language model has a perplexity of 253.3 [1] on the WSJ test set.

[1] The language model was trained with the sentence boundary

2.2. Information Retrieval System

An Information Retrieval System typically works in two phases, the *document indexing* phase and *query-document matching* phase. In the document indexing phase each document in the collection is processed to yield a document description, also known as a document-index, which stands in its place during the retrieval. In our system this processing involves part-of-speech tagging of the text, followed by a morphological analysis of the text, followed by removal of function words using a standard *stop word* list. This is in contrast to the simple stemming and filtering used by most of the current systems. Morphological analysis is a form of linguistic signal processing which has great utility in natural language processing. For instance during morphological analysis, among other decompositions, verbs are decomposed into units designating person, tense and mood of the verb plus the root of the verb. Similarly, nouns are decomposed into their roots with (possibly) a tag indicating the plural form. The written request is processed in an identical fashion to yield a *query*. For example, given the request

```
Security arrangements in Hebron involving
international peace-keepers.
```

the following query is obtained after the processing is done.

```
security arrangement Hebron to_involve
international peace-keepers
```

In general our retrieval system uses a 2-pass approach. However, for SDR97 evaluation we employed just the first-pass. In the first pass, given a query, a matching score is computed for each document and the documents are ranked according to this score. The scoring function is simply a weighting scheme that takes into account the number of times each query-term (n-grams in general) occurs in the document normalized with respect to the length of the document. Normalization is essential to remove the bias towards longer documents. The scoring function also favors terms that are specific to a document and thus rare (and hence more significant) across the documents. We use the following version of the Okapi formula [4], for computing the matching score between a document d and a query q:

$$S(d,q) = \sum_{k=1}^{Q} c_q(q_k) \frac{c_d(q_k)}{\alpha_1 + \alpha_2 \frac{l_d}{\bar{l}} + c_d(q_k)} \, idf(q_k).$$

Here, q_k is the kth term in the query, Q is the number of terms in the query, $c_q(q_k)$ and $c_d(q_k)$ are the counts of the kth term in the query and document respectively, l_d is the length of the document, $\bar{l}$ is the average length of the documents in the collection, and $idf(q_k)$ is the inverse document frequency for the term q_k which is given by:

$$idf(q_k) = log(\frac{N - n(q_k) + 0.5}{n(q_k) + 0.5}),$$

where N is the total number of documents and $n(q_k)$ is the number of documents that contain the term q_k. The inverse document frequency term thus favors terms that are rare among documents. We use a linear combination of unigram and bigram scores in the first pass with weights 0.8 and 0.2 respectively. For unigrams $\alpha_1 = 0.5$ and $\alpha_2 = 1.5$ were used and for bigrams $\alpha_1 = 0.05$ and $\alpha_2 = 0.05$ were used.

In the second pass we re-rank the documents by training a probabilistic relevance model for documents, using the top-ranked documents from the first pass as training data. Details of the second pass can be found in [6].

markers and OOV words included, however, they were not included in the perplexity computation.

3. The VOA Evaluation Corpus

Our Audio-Indexing evaluation corpus consists of approximately 20 hours of radio news broadcasts from the Voice of America covering the time period between May to June 1996. Each day only three broadcasts starting at a different hour and spaced roughly 8 hours apart were downloaded from their internet site. This was done to ensure that the broadcasts are not too similar in content and also to ensure that the collection had several different speakers. The entire collection has about 10 main speakers (both male and female anchors) with several more speakers (correspondents, interviewees etc.) contributing short segments. Each broadcast is typically 6 or 10 mins long and begins with a signature announcement followed by the signature music. A typical news bulletin usually consists of several news stories and often includes reports from correspondents over the telephone line and brief interviews with foreign speakers of English.

The entire speech collection is recognized with a large vocabulary speech recognizer to produce transcripts along with time-alignments for each word in the transcripts. Unlike in the standard information retrieval scenario where the text collection is segmented into documents with each document usually discussing a specific topic/story, story segmentation is not automatically available in this application. We, thus, need a scheme to segment the transcripts into stories. One method is to apply standard topic identification schemes to automatically segment the text into topics, however, a more simplistic solution to this problem is to break the transcript into overlapping segments of a fixed number of words and treat each segment as a separate document. We adopt such an approach in our experiment here, with 100 words in each document, resulting in 3412 documents in the collection.

# queries	53
average length in words	10
average number of relevant documents per query	11

Table 1: Query statistics

3.1. The Test Collection

Evaluating an information retrieval systems requires search requests, together with assessments of the relevance of each document to each of these requests. The search requests were collected from independent sources such as newspapers and other news broadcasts appearing during the same period of time. This method of collecting search requests is similar to the TREC evaluation and in general they form a better test for the information retrieval system than "known item retrieval", where users are asked to compose queries after reading the documents. We compiled 85 requests in this manner. Judging the relevance of each document for each of these queries is a time-consuming task. Instead, we took the following approach. We ran our information retrieval system on the document collection with each of these search requests and made relevance judgment of only the top 30 ranked documents for each query. We found that only 53 of the 85 requests had any relevant documents, which can be attributed to the small size of the database. We discarded the requests that did not have any relevant documents from our evaluation set. The query statistics are shown in Table 1.

4. Performance of the Speech Recognition System

The performance of the above system was tested on a test set composed of two 10 min VOA broadcasts and the results are shown is Table 2. The decoding speed, based on an IBM RS6000/590 machine, is about 30×real-time. On the WSJ test set the above system has a WER of 14.3%.

Corpus	WER (%)
WSJ	14.3
VOA	30.2

Table 2: Performance of **SR-0** on VOA and WSJ.

On HUB4 test data, System-1 had an average WER of about 50%. The WER under different acoustic conditions and speaking styles are shown in Table 3. The higher error rate on the VOA and the HUB4 test sets can be attributed to several reasons: (1) the VOA and HUB4 speech has a large proportion of spontaneous speech whereas the WSJ speech is mainly read speech, (2) the VOA speech is of a lower bandwidth (11KHz) and the HUB4 speech has different acoustic conditions than the WSJ speech, and, (3) the language model is not tuned to the VOA or HUB4 corpus.

Acoustic/Speaking conditions	WER (%)	# words
Baseline	32.2	120,714
Spontaneous	50.5	88,169
Telephone	63.3	69,595
Speech+Music	64.0	19,903
Degraded acousic conditions	46.9	46,369
Non-native speakers	38.0	1,942
Other	69.6	54,867
Overall	50.0	401,559

Table 3: Performance of **SR-1** on the HUB4 test set.

5. Performance of the Information Retrieval System On Clean Text

For the SDR track, "known item retrieval" performance was evaluated. Overall there were 49 topics or queries and 1451 documents. The performance of our system on reference transcripts is summarized in Table 4

Mean rank:	11.84
Mean reciprocal rank:	0.7923
Known items found at rank:	
≤ 1	37
≤ 5	41
≤ 10	46
≤ 20	47
≤ 100	47
Not found:	0

Table 4: IR system performance on reference transcripts

More often, however, retrieval performance is measured by two measures *precision* and *recall*. Precision is defined as the percentage of the retrieved documents that are relevant to the query and recall is defined as the percentage of the total number of relevant documents that are retrieved. These two measures can be traded off, one for the other. Often a single *average precision* number is computed by first computing the average of the precision at different recall rates for each query, and then by averaging this number across all queries. A more practi-

Total number of documents	Avg. Precision
140	83%
175000	29%

Table 5: IR system performance on TREC4

cal measurement, however, is the precision when a fixed number of documents (often small, between 10 and 20) are retrieved. Another commonly used measure is the rank of the highest-ranked relevant document for each query and the percentage of queries that have relevant documents within a given range of the ranked list of retrieved documents.

We evaluated the performance of our system on a small subset and the entire TREC4 document-collection. The results are tabulated in Table 5.

6. Combining Speech recognition with Information retrieval

The known item retrieval performance on the baseline transcripts produced by **SR-1** is shown in Table 6. In terms of the mean rank, we find a 156% degradation in performance, whereas in terms of the mean reciprocal rank the degradation is 12.6%. This clearly shows that the mean reciprocal rank is a better measure of performance of the system than the mean rank since the mean rank tends to be heavily influenced by outliers [2].

Mean rank:	30.31
Mean reciprocal rank:	0.6921
Known items found at rank:	
≤ 1	30
≤ 5	39
≤ 10	41
≤ 20	42
≤ 100	46
Not found:	0

Table 6: IR system performance on **SR-1** output

We also conducted a TREC-style evaluation of our system using the VOA corpus described in the previous section. All the results reported here are based on the speech recognition system **SR-0**. Figure 1 shows the precision vs recall rate for our audio-indexing system, averaged over the 53 queries. The average pecision after the first pass is computed to be 69.92%. With the second pass the average precision increases to 72.83%, which represents a relative increase of about 4.1%.

As described earlier, another way of presenting the retrieval performance is by plotting the precision vs the number of retrieved documents. This is shown in Figure 2. For example, the precision when the top 10 documents are retrieved is 57.92%. With the second pass this improves to 62.26% which represents a 7% relative improvement in performance.

A third method of measuring the retrieval preformance is by the percentage of queries that have relevant documents within a given range of the ranked list of retrieved documents. This is shown in Table 7. We find, for example, that after the first pass, 87% of the queries have at least one relevant document in the top 5 documents and 96% of the queries have at least one relevant document in the top 10 documents.

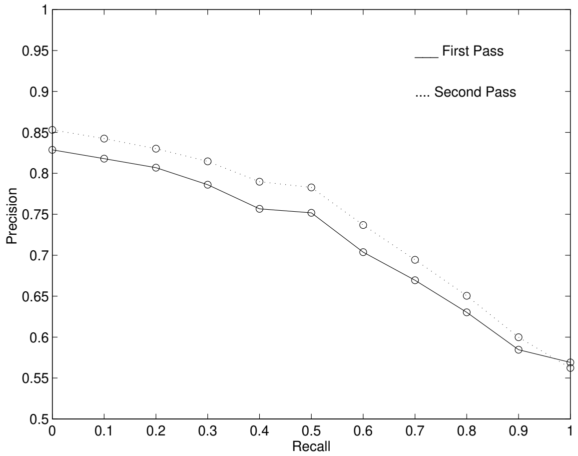

Figure 1: Precision vs Recall rate after the first and second pass.

7. Towards an Open Vocabulary System

One limitation with the current approach to audio-indexing is the finite coverage of the vocabulary used in the speech recognizer – words such as proper nouns and abbreviations that are important from a information retrieval standpoint are often found missing in the vocabulary and hence in the recognized transcripts. One method to overcome this limitation is to complement the speech recognizer with a wordspotter for the out of vocabulary (OOV) words. For this approach to be practical, however, one has to have the ability to detect spoken

[2] The mean rank was also greatly influenced by the assignment of a random rank to the relevant document when several documents shared the same score as the relevant document.

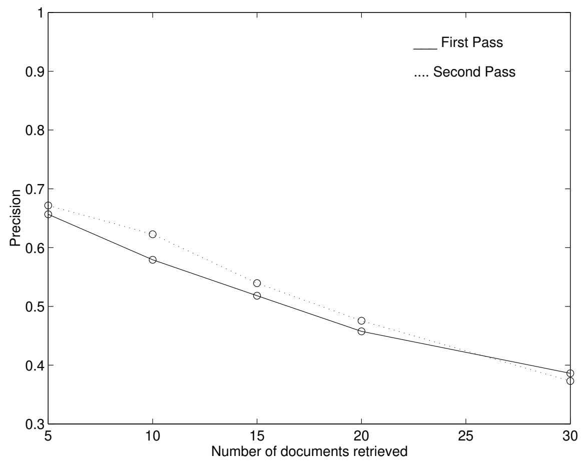

Figure 2: Precision vs number of retrieved documents after the first and second pass.

words in large amounts of speech at speeds many times faster than real-time.

We have developed a novel algorithm that gives us both speed of retrieval and the flexibility of being able to search for any word. We accomplish this by adopting a three-step procedure – a preprocessing step and a two stage search strategy. In the preprocessing step we convert the speech waveform into a representation consisting of a table of phone-ngrams with the times at which it occurs with a high likelihood. This representation allows us to search through the speech very efficiently. The two stage search consists of first a phone-ngram lookup to narrow down the time intervals where the word was likely to have been uttered and then a detailed acoustic match at these time intervals to finally decide more accurately whether the word was actually uttered in that time interval. The algorithm was tested on 10 hours of HUB4 test data using a system trained on the first 50

Rank (R)	% queries with at least one relevant document in top R ranks
5	86.79%
10	96.25%
15	98.11%
20	98.11%
30	100 %

Table 7: Rank (R) vs percentage queries with at least one relevant document in the top R ranks after the first pass

hours of the HUB4 corpus. On an average, (averaged over 12 OOV words) the detection rate was 48.8% at a false-alarm level of 10 and the average reduction in search was about 80-fold. A more detailed performance analysis is currently being conducted. Details of this algorithm and its performance can be found in [5]. Since there were only 5 OOV words in all topics (queries) of the SDR97 evaluation we did not use our wordspotting component in this evaluation.

8. Conclusions and Future work

We presented an overview of our Audio-Indexing System and reported the performance of our system on an audio-indexing task. Our system has an average precision of about 72% with 96% of the queries having a relevant document in the top 10 ranked list. The mean reciprocal rank in a known item retrieval task was 0.69 when the WER was about 50%. We are currently exploring new information retrieval methods that are better adapted to the errorful conditions created by the speech recognizer. Current work is also in progress to augment our system with the new scheme for detecting words that are out of the vocabulary of speech recognizer, yielding a open-vocabulary audio-indexing system.

References

1. L.R. Bahl and P.V. deSouza and P.S. Gopalakrishnan and D. Nahamoo and M.A. Picheny, "Robust methods for context-dependent features and models in a continuous speech recognizer," in *Proc., Intl Conf. on Acoust., Speech, and Sig. Proc.*, 1994.
2. P.S. Gopalakrishnan and L.R. Bahl and R. Mercer, "A tree search strategy for large vocabulary continuous speech recognition," in *Proc., Intl Conf. on Acoust., Speech, and Sig. Proc.*, 1995.
3. L. R. Bahl et al ., "Performance of the IBM large vocabulary continuous speech recognition system on the ARPA wall street journal task," in *Proc., Intl Conf. on Acoust., Speech, and Sig. Proc.*, pp. 41–44, 1995.
4. S.E. Robertson and S. Walker and K. Sparck-Jones and M.M Hancock-Beaulieu and M. Gatford, "Okapi at TREC-3," in *Proc., Third Text Retrieval Conference (NIST special publication)*, 1995.
5. S. Dharanipragada and S. Roukos, "New Word Detection in Audio-Indexing," to appear in *Proc., IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, CA, 1997.
6. M. Franz and S. Roukos, "TREC-6 Ad-Hoc Retrieval", to appear *Proc., Sixth Text Retrieval Conference (NIST special publication)*