FAST AUDIO SEARCH USING VECTOR SPACE MODELLING

Brett Matthews, Upendra Chaudhari and Bhuvana Ramabhadran

IBM TJ Watson Research Center
1101 Kitchawan Road, Rte 134
Yorktown Heights, NY, USA 10598

brett@ece.gatech.edu, {uvc,bhuvana}@us.ibm.com

ABSTRACT

Many techniques for retrieving arbitrary content from audio have been developed to leverage the important challenge of providing fast access to very large volumes of multimedia data. We present a two-stage method for fast audio search, where a vector-space modelling approach is first used to retrieve a short list of candidate audio segments for a query. The list of candidate segments is then searched using a word-based index for known words and a phone-based index for out-of-vocabulary words. We explore various system configurations and examine trade-offs between speed and accuracy. We evaluate our audio search system according to the NIST 2006 Spoken Term Detection evaluation initiative. We find that we can obtain a 30-times speedup for the search phase of our system with a 10% relative loss in accuracy.

Index Terms— Spoken-term detection, audio search, vector-space modelling, latent semantic indexing.

1. INTRODUCTION

Instantaneous access to very large volumes of data, in many forms, is an increasingly important convenience. Clearly, the sheer volume of material available presents a challenge to fast and accurate search and retrieval of desired information. Audio data present a particularly important challenge for information retrieval since text-based search can be insufficient when applied to such media as recorded telephone conversations, conference meetings and archived video data.

Automatic speech recognition (ASR) systems have been applied extensively to the audio search task. While topologies vary for audio search methods reported in the literature, in most of these systems ASR is first applied to the audio data to produce a transcription, or other text-based representation, to which text-based search methods can be applied. Many systems use a lattice representation of audio to model confusable alternatives to recognition hypotheses and thus improve recall [1, 2, 3]. For large databases, efficient search schemes such as inverted indices are often used since searching through a large collection of lattices can be computationally intensive. In [2], lattices are indexed by an approximation of expected term frequencies of phone N-grams in a two-stage search scheme. A truncated list of documents is then provided to a second stage on which a full linear search is performed.

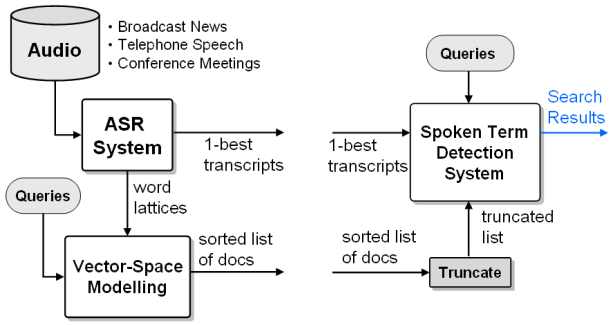
We present in this paper an efficient approach to audio search which draws upon vector-space modelling (VSM) of audio data. Vector-space modelling is common in text-based information retrieval, and has been applied to such speech recognition and retrieval tasks as language identification [4], retrieval by spoken query [5] and retrieval by spoken document [6].

An overview of our system for audio search is given in Figure 1. We extract word-based tokens from the lattice representation for each audio segment in the database and from the input query. A vector-space similarity measure is then used to retrieve a short list of likely candidate segments for each query. The list of candidate segments is then searched using word-based indexing for known words and phone-based indexing for out-of-vocabulary (OOV) entries. OOV words are scored according to a similarity measure based on phone confusion probabilities. We evaluate our audio search system according to the NIST 2006 Spoken Term Detection evaluation initiative.

The rest of the paper is organized as follows: Section 2 describes the NIST 2006 Spoken Term Detection evaluation initiative and its primary performance measure, the Actual Term-Weighted Value (ATWV) statistic. A brief overview of our system for audio search is given in section 3. A brief review of vector-space modelling and its application to audio search is provided in Section 4. Our spoken term detection system is described in section 5. We evaluate our overall system and its two major components in terms of speed and accuracy in Section 6. Finally conclusions and extensions to future work are given in Section 7.

2. SPOKEN TERM DETECTION TASK

We evaluate our audio search method according to the NIST 2006 Spoken Term Detection (STD) evaluation initiative [7],



(a) Audio Segment Retrieval (b) Spoken Term Detection

Fig. 1. Audio search system overview

where audio search is formulated as a detection task. The NIST evaluation provides a standard set of raw audio data and a list of 1107 query terms. For each query term, which may consist of one or more words, systems are required to locate all occurrences in the database and return their start and stop times. The primary evaluation measure, the Actual Term Weighted Value measure (ATWV) is a weighted average based on the number of false alarms and misses for each query, and is given by

$$ATWV = 1 - \frac{1}{N_{terms}} \sum_{t \in terms} \left(P_{miss}(t) + \beta \cdot P_{fa}(t) \right) \quad (1)$$

where $\beta \approx 1000$, $P_{miss}(t)$ and $P_{fa}(t)$ are defined as

$$P_{miss}(t) = 1 - \frac{N_{corr}(t)}{N_{true}(t)} \quad (2)$$

$$P_{fa}(t) = \frac{N_{spurious}(t)}{Total - N_{true}(t)}. \quad (3)$$

For each term t, $N_{corr}(t)$ and $N_{spurious}(t)$, are the number of correct and incorrect detections, respectively. $N_{true}(t)$ is the number of occurrences of t in the database and $Total$ is a value proportional to the total length of audio in the database.

3. SYSTEM OVERVIEW

Our system for audio search consists of an audio segment retrieval stage followed by a spoken term detection stage, as depicted in Figure 1. An ASR system, described in detail in Section 6.1, is used to produce lattices for the audio segment retrieval stage, and 1-best hypotheses for the spoken term detection stage. A vector-space representation of queries and lattices is used to score audio segments according to their relevance to each input query and to produce a sorted list of audio segments in the database. The spoken term detection system accepts the set of 1-best transcriptions as well as a truncated list of audio segments as inputs; only transcriptions corresponding to audio segments in the list are entered into its indexing system to be searched. If the list of audio segments is short and consists mostly of segments relevant to an input query, then the search system will see significant speedup with minimal loss of accuracy.

4. VECTOR-SPACE MODELLING FOR AUDIO SEGMENT RETRIEVAL

Vector-space modelling (VSM) approaches to information retrieval convert documents from their original form to a numeric vector, often called a *document vector*. Input queries can be similarly converted to *query vectors* and the relevance of any document in the set to an input query can be determined numerically, using vector-space similarity metrics or clustering techniques. The cosine distance, which gives the cosine of the angle between two vectors, is commonly used to express the similarity between two document vectors. For two document vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ the cosine similarity is given by

$$SIM_{cos}(\mathbf{x}^1, \mathbf{x}^2) = \frac{\mathbf{x}^1 \cdot \mathbf{x}^2}{|\mathbf{x}^1||\mathbf{x}^2|} \quad (4)$$

In the remainder of this section, we briefly review Latent Semantic Analysis (LSA), a popular VSM technique for information retrieval [8]. We then discuss our application of LSA to audio search.

4.1. Latent Semantic Analysis

With Latent Semantic Analysis (LSA), as with VSM in general, documents and queries are converted to numeric vectors using a so-called *bag-of-words* model in which a finite list of terms of interest is extracted from all documents without regard for their original ordering. Each document vector $\mathbf{x}^j$ then has one element for each item in the list of terms, and its values are some meaningful statistic used to represent the co-occurrence between the document and its terms. The collection of document vectors can then be expressed as a term-document co-occurrence matrix C, such that

$$C = \left[\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^D \right]. \quad (5)$$

With LSA, a document d_j is expressed as the vector $\mathbf{x}^j$ according to the following relation

$$\mathbf{x}_i^j = (1 - \epsilon(w_i)) \cdot \frac{n(w_i, d_j)}{\sum_i n(w_i, d_j)} \quad (6)$$

where $n(w_i, d_j)$ is the count of term w_i in document d_j and $\sum_i n(w_i, d_j)$ is the count of all terms in document d_j. The *normalized entropy*, $\epsilon(w_i)$, of term w_i across all documents in the database, is given by

$$\epsilon(w_i) = -\frac{1}{\ln D} \sum_{j=1}^{D} \frac{n(w_i, d_j)}{\sum_j n(w_i, d_j)} \ln \frac{n(w_i, d_j)}{\sum_j n(w_i, d_j)} \quad (7)$$

where D is the total number of documents in the database. This weighting scheme for the terms in matrix C is commonly called *TF-epsilon* [8] and is used in all experiments discussed in this paper.

A singular value decomposition (SVD) is then applied to the matrix C, which is typically very sparse, such that $C = T\ S\ D' \approx \tilde{T}\ \tilde{S}\ \tilde{D}'$. The matrix $\tilde{S}$ is a reduced version of S in which only the R largest singular values are retained. Similarly $\tilde{T}$ is a reduced version of T retaining the leftmost R columns. A document vector $\mathbf{x}^j$ is said to be projected onto the *latent semantic space* according to the following relation

$$\tilde{\mathbf{x}}^j = \tilde{T}' \cdot \mathbf{x}^j \tag{8}$$

where $\tilde{\mathbf{x}}^j$ is typically of lower dimension than $\mathbf{x}^j$. A query vector $\mathbf{q}$ can be similarly projected such that $\tilde{\mathbf{q}} = \tilde{T}' \cdot \mathbf{q}$. The cosine metric

$$SIM_{cos}(\tilde{\mathbf{q}}, \tilde{\mathbf{x}}^j) = \frac{\tilde{\mathbf{q}} \cdot \tilde{\mathbf{x}}^j}{|\tilde{\mathbf{q}}||\tilde{\mathbf{x}}^j|} \tag{9}$$

can then be used to judge the similarity between a query q and a document d_j.

In this paper we use LSA to sort a list of all of the audio segments in the database, according to the criterion in (9) for each of the 1107 queries provided by the NIST Spoken Term Detection evaluation initiative. The sorted lists are then truncated and searched in the second stage of our spoken term detection system.

In the following section, we discuss our methods for creating document vectors from a lattice representation of audio.

4.2. Lattice-Based Indexing for VSM

As in previously reported systems for audio search [1, 2, 3, 9, 10] we use a lattice representation for each spoken document in our database. For vector-space modelling, it is necessary to extract an unordered list of terms of interest, along with their counts, from each document in the dataset. We accomplish this for lattices by extracting expected counts of each term. The expected term count $ETC(w_i, d_j)$ of term w_i in document d_j is the expected number of occurrences of w_i over all paths in the lattice representation for document d_j, and is given by

$$ETC(w_i, d_j) = \sum_{l \in L_j} P_{L_j}(l|\mathbf{O}) \cdot C_l(w_i) \tag{10}$$

where L_j is the complete set of paths in the lattice, $C_l(w_i)$ is the count of term w_i in path l and $P_{L_j}(l|\mathbf{O})$ is the posterior probability of path l given an observation sequence $\mathbf{O}$. To create a document vector $\mathbf{x}^j$ we substitute $ETC(w_i, d_j)$ for $n(w_i, d_j)$ in (6) and (7).

Training Data

The matrix $\tilde{T}$ is constructed off-line from data independent of the testing set. We use reference transcripts, instead of lattices or the 1-best output of a recognizer, to build the off-line, term-document co-occurrence matrix for training. We build an unordered list of terms from the most frequently occurring 1-gram tokens in the training set, which is used to build test set document vectors as well. Since we use word-based lattices and transcripts, the vector-space modelling front end to our system does not account for out-of-vocabulary (OOV) terms in a query. The need for vocabulary independent search is handled in the back end spoken term detection stage, and is discussed in Section 5.

5. SPOKEN-TERM DETECTION SYSTEM

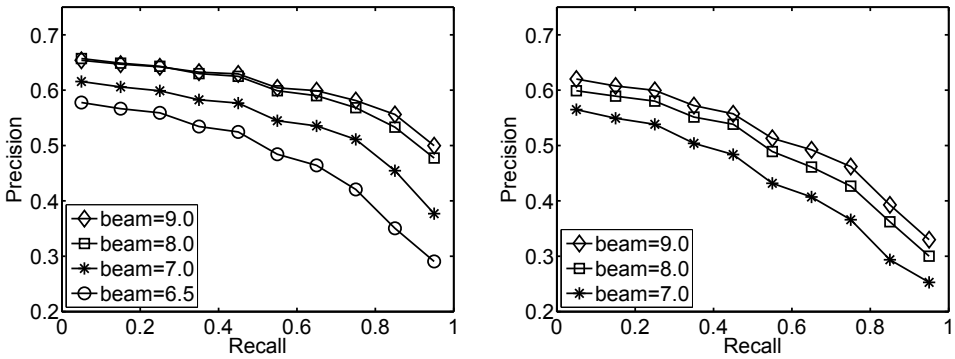
The detection system used employs a combination of a word based and phone based indexing scheme. The documents to be searched are first passed through an ASR system to generate word based transcripts, and these are later expanded into phone based transcripts using the recognizer's lexicon. Each item in the word based index points to the documents in which it occurs along with the beginning and end times. Exact match is used on the terms being searched. The word boundary information is used to ensure that terms in multi-word queries co-occur.

5.1. Phone Sequence Match

The phone based approach [11] is used when the search terms are not found in the word based index. In this case, a phone sequence match is used to judge similarity. If we define the phone set as $\mathcal{P} = \{\mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3, ...\}$, then the expanded transcripts will be sequences with elements in $\mathcal{P}$. The query terms that are not found in the word based transcripts are expanded into phone sequences using an automatic baseform generator. Then, a sequence similarity measure is used to determine where they are present. This measure is based on estimates of phone confusion $P(\mathrm{p}_i|\mathrm{p}_j)$ which is the probability that p_i is the true phone when p_j is observed. These estimates are derived from heldout data which is decoded in parallel with the decoded and reference transcripts to produce parallel phone level alignments. The two results are used to compute the phone confusion matrix.

5.2. Index Construction for OOV Terms

To make the sequence match efficient, N-grams are extracted from the decoded phone based transcripts and an inverted index is created with each N-gram in the index pointing to a list of documents and the positions within each where the N-grams occur. The search index contains all of the unique N-grams $\{\mathbf{h}_i\}$ that occurred in the database of documents. At test time, the query terms Q are also used to generate N-grams $Q \mapsto \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{N_Q}\}$.



(a) System 1: 250K, SI+SA decode

(b) System 2: 30K, SI decode only

Fig. 2. Average precision vs recall for audio segment retrieval on Broadcast News.

5.2.1. Generating scores

The score for each document D is

$$\frac{1}{N_Q} \sum_{\mathbf{q}_i \in Q} \mathcal{S}(\mathbf{q}_i, \mathbf{h}^*(D, \mathbf{q}_i)),$$

where $\mathbf{h}^*(D, \mathbf{q}_i)$ is the best matching index element corresponding to the document and a scoring procedure is used to compute the constrained match *similarity* $\mathcal{S}(\mathbf{q}, \mathbf{h})$, with parameters given by $P(\mathrm{p}_i|\mathrm{p}_j)$ (see [11] for details).

The time taken for search depends linearly on the size of the index, so that any reduction in the number of elements to consider directly affects the search time. Each document corresponds to a subset of the index with elements that occur in that document. Thus, if prior to the search, a filtering has determined that only a restricted set of documents need be considered, then only the index elements in the union of the corresponding index subsets need to be tested.

6. EXPERIMENTS

Our audio search system uses vector-space modelling, to retrieve a list of relevant audio segments followed by a spoken term detection system. In this section we evaluate the performance of these two stages independently and in concert. The audio segment retrieval system is evaluated according to the trade-off between precision and recall, where *precision* indicates the fraction of documents in a truncated list, relevant to a given query. *Recall*, in contrast, is the fraction of documents in a truncated list taken with respect to all relevant documents in the database. The spoken term detection system is evaluated according to the ATWV measure discussed in Section 2. Finally, we evaluate the combined system by examining the effect on the ATWV score when a truncated list is used for audio search. We also report the execution time for generating lattices with respect to the complete length of audio in the database.

6.1. Data

Broadcast News

For the broadcast news audio search task, the NIST evaluation initiative provides 2.79 hours of broadcast news data. 1107 query terms including, single-word and multi-word terms are also provided with the evaluation. Examples of single-word and multi-word queries in the evaluation set are "too" and "six point eight," respectively.

Raw audio in the database is first segmented by speaker into short audio segments. We then create a lattice representation for each of 1408 audio segments in the database by using a speaker-adapted ASR system developed for English broadcast news transcription. Although generating lattices for audio segments in a database could be considered part of an offline indexing phase, we consider the efficiency of this stage in the system to be an important design criterion since a practical system could need to search very large volumes of audio. We compare two ASR systems, differing significantly in their complexity, for generating lattices to examine the trade-off between speed and accuracy of retrieval. The two systems are briefly described in the remainder of this section.

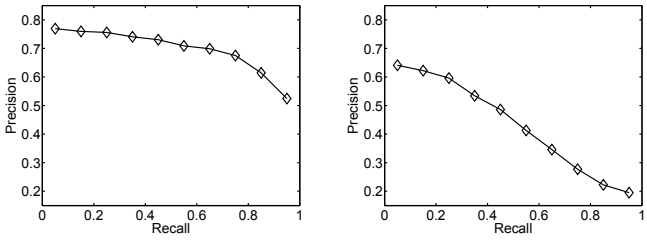
ASR System 1: 250K, SI+SA decode

We use the speaker-adapted ASR system, described in [12] to create lattices. Its speaker-independent and speaker-adapted models share a common alphabet of 6000 quinphone context-dependent states and 250K Gaussian mixtures. 40-dimensional recognition features computed via a subspace projection of 9 frames of 19-dimensional PLP features are used for both the speaker-independent (SI) and speaker-adapted (SA) training phases. Acoustic models for the speaker-independent phase are trained to optimize the MMI criterion. Features for speaker-adapted training are normalized with vocal-tract length normalization as well as speaker-wise mean subtraction and variance normalization.

ASR System 2: 30K, SI only decode

Our speaker-independent system for producing lattices is significantly less complex. It uses speaker-independent models

with 30K Gaussian mixtures and 3000 context-dependent triphone states. No speaker-adapted models were trained for this system. Both of these systems use a 4-gram language model, built from a 54M n-gram corpus.



(a) Conversational Telephone Speech (b) Conference Meetings

Fig. 3. Average precision vs. recall for Conversational Telephone Speech (CTS) and Conference Meetings (CNFMTG) audio.

6.2. Audio Segment Retrieval

Plots of average precision vs. recall are given in Figure 2 for Broadcast News data. The pruning beam width for generating lattices has a direct effect on speed and ASR decoding accuracy. In Figure 2 we illustrate the trade-off between speed and *retrieval* accuracy by varying the beam width for lattice generation. We also compare ASR System 1, with System 2, in Figures 2 (a) and (b), respectively. Precision for the two ASR systems is comparable for low values of recall, but System 1 retains high precision even when recall is high (i.e. when most of the relevant documents have been retrieved). This implies that, on average, truncated lists containing a comparable number of documents relevant to a query, are shorter for System 1.

System 1: 250K SI+SA decode		System 2: 30K SI decode only	
beam	RTF	beam	RTF
6.5	0.43	7.0	0.09
7.0	0.43	8.0	0.11
8.0	0.48	9.0	0.13
9.0	0.55	---	---

Table 1. Execution times for generating lattices in terms of real-time factor (RTF).

While the speaker-adapted System 1 gives better retrieval accuracy, its complexity causes it to be significantly slower than System 2. Table 1 reports execution times for generating lattices for System 1 and System 2 as a fraction of the length of audio in the database, often called the Real Time Factor (RTF). The slowest execution time for System 2, 0.13xRT, is significantly faster than the fastest time for System 1 at 0.43xRT.

Telephone Speech and Conference Meetings

The NIST evaluation also provides about 3 hours of conversational telephone speech and 2 hours of recorded conference meetings. Lattices were generated for telephone speech and conference meeting data using speech transcription systems similar to ASR system 1, but trained on audio data appropriate to each source type. The same set of queries is used for all source types in the NIST Spoken Term Detection evaluation. Plots of precision versus recall are given for retrieving audio segments from telephone speech and conference meetings data in Figure 3. For the best configurations of our speech transcription system, we achieve a word error rate (WER) of 12.5% on the development set for Broadcast News data, and 19.6% and 47% for telephone speech and conference meetings, respectively. As Figure 3 (a) shows, precision for audio segment retrieval or conversational telephone speech is above 0.65 even when 80% of the relevant documents have been retrieved, on average, and is the best for all source types.

6.3. Spoken Term Detection

Performance results for our back-end spoken term detection system for broadcast news data, given in terms of the Actual Term Weighted Value measure in (1), are plotted in Figure 4. Searching through the full (unfiltered) list of documents, our back end spoken term detection system achieves an ATWV score of 0.78. This is indicated in Figure 4 with a single flat dashed line.

The effect of the audio segment retrieval front end is also illustrated in Figure 4. The ATWV score achieved when sorted lists provided by the front-end system are truncated to various lengths is plotted for ASR Systems 1 and 2 in Figure 4 for various pruning beam widths. While using ASR System 1 generally results in higher search accuracy, the best configuration for ASR System 2 is generally within 10 points, absolute, of the best configuration for ASR System 2.

Since search time for our spoken term detection system varies linearly with the size of the list of audio segments, Figure 4 depicts a trade-off between speed and accuracy for our search system. When the list of audio segments is truncated to 3% of its full length, our audio search system achieves a speedup of about 30 times. Our best configuration for ASR system 1, with a pruning beam width of 9.0, achieves an ATWV score of 0.703, a relative loss of about 10% in search accuracy. The best performance for System 2, at a beam width of 9.0, is 0.633, a further 10% reduction in search performance.

7. CONCLUSIONS AND FUTURE WORK

We have presented a system for spoken term detection using vector-space modelling to retrieve a list of promising audio

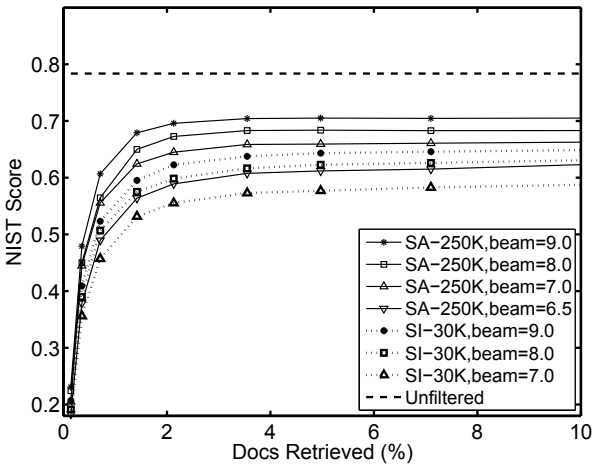


Fig. 4. ATWV vs. fraction of audio segments retrieved for broadcast news data.

segments. We use a lattice representation for audio segments in the database and apply vector-space similarity measures to identify segments relevant to each query. A vocabulary-independent, audio search system is then applied to the reduced set to achieve gains in search speed. The trade-off between speed and accuracy of both the indexing and search phases is illustrated with various pruning beam widths for generating lattices, two transcription systems of differing complexity, and truncating the list of audio segments to various lengths. We find that a 30-times speed up can be achieved for the search system with a 10% relative loss in search accuracy as measured by the NIST actual term weighted value measure. With a further relative loss of 10% accuracy, 4.23 times speedup in indexing time can also be achieved.

Possible extensions to this work include the use of higher order n-grams for vector-space modelling as well as other VSM based techniques such as probabilistic Latent Semantic Analysis (pLSA) [13]. We will also explore the use of the vector-space similarity measure for clustering audio segments and as a fast confidence score for search.

8. ACKNOWLEDGMENTS

The authors would like to thank Prof. Chin-Hui Lee of the Georgia Institute of Technology for many valuable discussions in the development of this work.

9. REFERENCES

[1] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *HLT-NAACL*, 2004.

[2] Peng Yu and F. Seide, "Fast two-stage vocabulary-independent search in spontaneous speech," in *ICASSP-2005*, vol. 1, pp. 481–484.

[3] Olivier Siohan and Michiel Bacchiani, "Fast vocabulary-independent audio search using path-based graph indexing," in *INTERSPEECH-2005*, pp. 53–56.

[4] Haizhou Li, Bin Ma, and Chin-Hui Lee, "A vector space modeling approach to spoken language identification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 271–284, January 2007.

[5] A. Moreno-Daniel, S. Parthasarathy, B. H. Juang, and J. G. Wilpon, "Spoken query processing for information retrieval," in *ICASSP-2007*, vol. 4, pp. 121–124.

[6] K. Thambiratnam and F. Seide and P. Yu, "Discriminatively trained spoken document similarity models and their application to probabilistic latent semantic analysis," .

[7] NIST, "The spoken term detection (std) 2006 evaluation plan," http://www/nist/gov/speech/tests/std/docs/std06-evalplan-v10.pdf, 2006.

[8] J.R. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1279–1296, August 2000.

[9] J.S. Olsson, J. Wintrode, and M. Lee, "Fast unconstrained audio search in numerous human languages," in *ICASSP-2007*, vol. 4, pp. 77–80.

[10] K. Thambiratnam and S. Sridharan, "Dynamic match phone-lattice searches for very fast and accurate unrestricted vocabulary keyword spotting," in *ICASSP-2005*, pp. 465–468.

[11] U. V. Chaudhari and M. Picheny, "Improvements in phone-based audio search via high order confusion estimates," in *Submitted to ASRU-2007*.

[12] Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan, "Vocabulary independent spoken term detection," in *The 30th Annual International ACM SIGIR Conference*, July 2007.

[13] Thomas Hofmann, "Probabilistic latent semantic analysis," in *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.

[14] Ciprian Chelba and Alex Acero, "Position specific posterior lattices for indexing speech," in *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, 2005, pp. 443–450, Association for Computational Linguistics.