Position Specific Posterior Lattices for Indexing Speech

Ciprian Chelba and **Alex Acero**
Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
{chelba, alexac}@microsoft.com

Abstract

The paper presents the Position Specific Posterior Lattice, a novel representation of automatic speech recognition lattices that naturally lends itself to efficient indexing of position information and subsequent relevance ranking of spoken documents using proximity.

In experiments performed on a collection of lecture recordings — MIT iCampus data — the spoken document ranking accuracy was improved by 20% relative over the commonly used baseline of indexing the 1-best output from an automatic speech recognizer. The Mean Average Precision (MAP) increased from 0.53 when using 1-best output to 0.62 when using the new lattice representation. The reference used for evaluation is the output of a standard retrieval engine working on the manual transcription of the speech collection.

Albeit lossy, the PSPL lattice is also much more compact than the ASR 3-gram lattice from which it is computed — which translates in reduced inverted index size as well — at virtually no degradation in word-error-rate performance. Since new paths are introduced in the lattice, the ORACLE accuracy increases over the original ASR lattice.

1 Introduction

Ever increasing computing power and connectivity bandwidth together with falling storage costs result in an overwhelming amount of data of various types being produced, exchanged, and stored. Consequently, search has emerged as a key application as more and more data is being saved (Church, 2003). Text search in particular is the most active area, with applications that range from web and intranet search to searching for private information residing on one's hard-drive.

Speech search has not received much attention due to the fact that large collections of untranscribed spoken material have not been available, mostly due to storage constraints. As storage is becoming cheaper, the availability and usefulness of large collections of spoken documents is limited strictly by the lack of adequate technology to exploit them.

Manually transcribing speech is expensive and sometimes outright impossible due to privacy concerns. This leads us to exploring an automatic approach to searching and navigating spoken document collections.

Our current work aims at extending the standard keyword search paradigm from text documents to spoken documents. In order to deal with limitations of current automatic speech recognition (ASR) technology we propose an approach that uses recognition lattices — which are considerably more accurate than the ASR 1-best output.

A novel contribution is the use of a representation of ASR lattices which retains only position information for each word. The Position Specific Posterior

Lattice (PSPL) is a lossy but compact representation of a speech recognition lattice that lends itself to the standard inverted indexing done in text search — which retains the position as well as other contextual information for each hit.

Since our aim is to bridge the gap between text and speech -grade search technology, we take as our reference the output of a text retrieval engine that runs on the manual transcription.

The rest of the paper is structured as follows: in the next section we review previous work in the area, followed by Section 3 which presents a brief overview of state-of-the-art text search technology. We then introduce the PSPL representation in Section 4 and explain its use for indexing and searching speech in the next section. Experiments evaluating ASR accuracy on iCampus, highlighting empirical aspects of PSPL lattices as well as search accuracy results are reported in Section 6. We conclude by outlining future work.

2 Previous Work

The main research effort aiming at spoken document retrieval (SDR) was centered around the SDR-TREC evaluations (Garofolo et al., 2000), although there is a large body of work in this area prior to the SDR-TREC evaluations, as well as more recent work outside this community. Most notable are the contributions of (Brown et al., 1996) and (James, 1995).

One problem encountered in work published prior or outside the SDR-TREC community is that it doesn't always evaluate performance from a document retrieval point of view — using a metric like Mean Average Precision (MAP) or similar, see `trec_eval` (NIST, www) — but rather uses word-spotting measures, which are more technology- rather than user- centric. *We believe that ultimately it is the document retrieval performance that matters and the word-spotting accuracy is just an indicator for how a SDR system might be improved.*

The TREC-SDR 8/9 evaluations — (Garofolo et al., 2000) Section 6 — focused on using Broadcast News speech from various sources: CNN, ABC, PRI, Voice of America. About 550 hrs of speech were segmented manually into 21,574 stories each comprising about 250 words on the average. The approximate manual transcriptions — closed captioning for video — used for SDR system comparison with text-only retrieval performance had fairly high WER: 14.5% video and 7.5% radio broadcasts. ASR systems tuned to the Broadcast News domain were evaluated on detailed manual transcriptions and were able to achieve 15-20% WER, not far from the accuracy of the approximate manual transcriptions. In order to evaluate the accuracy of retrieval systems, search queries —"topics" — along with binary relevance judgments were compiled by human assessors.

SDR systems indexed the ASR 1-best output and their retrieval performance — measured in terms of MAP — was found to be flat with respect to ASR WER variations in the range of 15%-30%. Simply having a common task and an evaluation-driven collaborative research effort represents a huge gain for the community. There are shortcomings however to the SDR-TREC framework.

It is well known that ASR systems are very brittle to mismatched training/test conditions and it is unrealistic to expect error rates in the range 10-15% when decoding speech mismatched with respect to the training data. It is thus very important to consider ASR operating points which have higher WER.

Also, the out-of-vocabulary (OOV) rate was very low, below 1%. Since the "topics"/queries were long and stated in plain English rather than using the keyword search paradigm, the query-side OOV (Q-OOV) was very low as well, an unrealistic situation in practice. (Woodland et al., 2000) evaluates the effect of Q-OOV rate on retrieval performance by reducing the ASR vocabulary size such that the Q-OOV rate comes closer to 15%, a much more realistic figure since search keywords are typically rare words. They show severe degradation in MAP performance — 50% relative, from 44 to 22.

The most common approach to dealing with OOV query words is to represent both the query and the spoken document using sub-word units — typically phones or phone n-grams — and then match sequences of such units. In his thesis, (Ng, 2000) shows the feasibility of sub-word SDR and advocates for tighter integration between ASR and IR technology. Similar conclusions are drawn by the excellent work in (Siegler, 1999).

As pointed out in (Logan et al., 2002), word level indexing and querying is still more accurate, were it not for the OOV problem. The authors argue in favor of a combination of word and sub-word level indexing. Another problem pointed out by the paper is the abundance of word-spotting false-positives in the sub-word retrieval case, somewhat masked by the MAP measure.

Similar approaches are taken by (Seide and Yu, 2004). One interesting feature of this work is a two-pass system whereby an approximate match is carried out at the document level after which the costly detailed phonetic match is carried out on only 15% of the documents in the collection.

More recently, (Saraclar and Sproat, 2004) shows improvement in word-spotting accuracy by using lattices instead of 1-best. An inverted index from symbols — word or phone — to links allows to evaluate adjacency of query words but more general proximity information is harder to obtain — see Section 4. Although no formal comparison has been carried out, we believe our approach should yield a more compact index.

Before discussing our architectural design decisions it is probably useful to give a brief presentation of a state-of-the-art text document retrieval engine that is using the keyword search paradigm.

3 Text Document Retrieval

Probably the most widespread text retrieval model is the TF-IDF vector model (Baeza-Yates and Ribeiro-Neto, 1999). For a given query $\mathcal{Q} = q_1 \dots q_i \dots q_Q$ and document D_j one calculates a similarity measure by accumulating the TF-IDF score $w_{i,j}$ for each query term q_i, possibly weighted by a document specific weight:

$$\begin{aligned} S(D_j, \mathcal{Q}) &= \sum_{i=1}^{Q} w_{i,j} \\ w_{i,j} &= f_{i,j} \cdot idf_i \end{aligned}$$

where $f_{i,j}$ is the normalized frequency of word q_i in document D_j and the inverse document frequency for query term q_i is $idf_i = \log \frac{N}{n_i}$ where N is the total number of documents in the collection and n_i is the number of documents containing q_i.

The main criticism to the TF-IDF relevance score is the fact that the query terms are assumed to be independent. *Proximity information* is not taken into account at all, e.g. whether the words LANGUAGE and MODELING occur next to each other or not in a document is not used for relevance scoring.

Another issue is that query terms may be encountered in different *contexts* in a given document: title, abstract, author name, font size, etc. For hypertext document collections even more context information is available: anchor text, as well as other mark-up tags designating various parts of a given document being just a few examples. The TF-IDF ranking scheme completely discards such information although it is clearly important in practice.

3.1 Early Google Approach

Aside from the use of PageRank for relevance ranking, (Brin and Page, 1998) also uses both *proximity* and *context* information heavily when assigning a relevance score to a given document — see Section 4.5.1 of (Brin and Page, 1998) for details.

For each given query term q_i one retrieves the list of *hits* corresponding to q_i in document D. Hits can be of various types depending on the *context* in which the hit occurred: title, anchor text, etc. Each type of hit has its own *type-weight* and the type-weights are indexed by type.

For a single word query, their ranking algorithm takes the inner-product between the type-weight vector and a vector consisting of count-weights (tapered counts such that the effect of large counts is discounted) and combines the resulting score with PageRank in a final relevance score.

For multiple word queries, terms co-occurring in a given document are considered as forming different *proximity-types* based on their proximity, from adjacent to “not even close”. Each proximity type comes with a proximity-weight and the relevance score includes the contribution of proximity information by taking the inner product over all types, including the proximity ones.

3.2 Inverted Index

Of essence to fast retrieval on static document collections of medium to large size is the use of an *inverted index*. The inverted index stores a list of hits for each word in a given vocabulary. The hits are grouped by document. For each document, the list of hits for a given query term must include position — needed to evaluate counts of proximity types —

as well as all the context information needed to calculate the relevance score of a given document using the scheme outlined previously. For details, the reader is referred to (Brin and Page, 1998), Section 4.

4 Position Specific Posterior Lattices

As highlighted in the previous section, position information is crucial for being able to evaluate proximity information when assigning a relevance score to a given document.

In the spoken document case however, we are faced with a dilemma. On one hand, using 1-best ASR output as the transcription to be indexed is suboptimal due to the high WER, which is likely to lead to low recall — query terms that were in fact spoken are wrongly recognized and thus not retrieved. On the other hand, ASR lattices do have much better WER — in our case the 1-best WER was 55% whereas the lattice WER was 30% — but the position information is not readily available: it is easy to evaluate whether two words are adjacent but questions about the distance in number of links between the occurrences of two query words in the lattice are very hard to answer.

The position information needed for recording a given word hit is not readily available in ASR lattices — for details on the format of typical ASR lattices and the information stored in such lattices the reader is referred to (Young et al., 2002). To simplify the discussion let's consider that a traditional text-document hit for given word consists of just `(document id, position)`.

The occurrence of a given word in a lattice obtained from a given spoken document is uncertain and so is the position at which the word occurs in the document.

The ASR lattices do contain the information needed to evaluate proximity information, since on a given path through the lattice we can easily assign a position index to each link/word in the normal way. Each path occurs with a given posterior probability, easily computable from the lattice, so in principle one could index *soft-hits* which specify

```
(document id, position,
              posterior probability)
```

for each word in the lattice. Since it is likely that more than one path contains the same word in the same position, one would need to sum over all possible paths in a lattice that contain a given word at a given position.

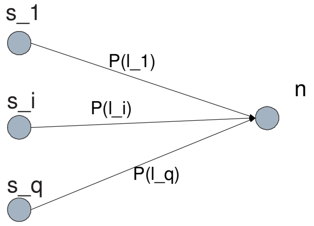

Figure 1: State Transitions

A simple dynamic programming algorithm which is a variation on the standard forward-backward algorithm can be employed for performing this computation. The computation for the backward pass stays unchanged, whereas during the forward pass one needs to split the forward probability arriving at a given node n, α_n, according to the length l — measured in number of links along the partial path that contain a word; null (ϵ) links are not counted when calculating path length — of the partial paths that start at the start node of the lattice and end at node n:

$$\alpha_n[l] \doteq \sum_{\pi:end(\pi)=n,length(\pi)=l} P(\pi)$$

The backward probability β_n has the standard definition (Rabiner, 1989).

To formalize the calculation of the position-specific forward-backward pass, the initialization, and one elementary forward step in the forward pass are carried out using Eq. (1), respectively — see Figure 1 for notation:

$$\begin{aligned} \alpha_n[l+1] &= \sum_{i=1}^{q} \alpha_{s_i}[l+\delta(l_i,\epsilon)] \cdot P(l_i) \\ \alpha_{start}[l] &= \begin{cases} 1.0, l=0 \\ 0.0, l \neq 0 \end{cases} \end{aligned} \quad (1)$$

The "probability" $P(l_i)$ of a given link l_i is stored as a $\log$-probability and commonly evaluated in ASR using:

$$\log P(l_i) = FLATw \cdot [1/LMw \cdot \log P_{AM}(l_i) + \log P_{LM}(word(l_i)) - 1/LMw \cdot logP_{IP}] \quad (2)$$

where $\log P_{AM}(l_i)$ is the acoustic model score, $\log P_{LM}(word(l_i))$ is the language model score, $LMw > 0$ is the language model weight, $logP_{IP} > 0$ is the "insertion penalty" and $FLATw$ is a flattening weight. In N-gram lattices where $N \geq 2$, all links ending at a given node n must contain the same word $word(n)$, so the posterior probability of a given word w occurring at a given position l can be easily calculated using:

$$P(w,l|LAT) = \\ \sum_{n \ s.t. \ \alpha_n[l]\cdot\beta_n>0} \frac{\alpha_n[l]\cdot\beta_n}{\beta_{start}} \cdot \delta(w, word(n))$$

The Position Specific Posterior Lattice (PSPL) is a representation of the $P(w,l|LAT)$ distribution: for each position bin l store the words w along with their posterior probability $P(w,l|LAT)$.

5 Spoken Document Indexing and Search Using PSPL

Spoken documents rarely contain only speech. Often they have a title, author and creation date. There might also be a text abstract associated with the speech, video or even slides in some standard format. The idea of saving *context information* when indexing HTML documents and web pages can thus be readily used for indexing spoken documents, although the context information is of a different nature.

As for the actual *speech content* of a spoken document, the previous section showed how ASR technology and PSPL lattices can be used to automatically convert it to a format that allows the indexing of *soft hits* — a *soft index* stores posterior probability along with the position information for term occurrences in a given document.

5.1 Speech Content Indexing Using PSPL

Speech content can be very long. In our case the speech content of a typical spoken document was approximately 1 hr long; it is customary to segment a given speech file in shorter segments.

A spoken document thus consists of an ordered list of segments. For each segment we generate a corresponding PSPL lattice. Each document and each segment in a given collection are mapped to an integer value using a *collection descriptor file* which lists all documents and segments. Each *soft hit* in our index will store the PSPL position and posterior probability.

5.2 Speech Content Relevance Ranking Using PSPL Representation

Consider a given query $\mathcal{Q} = q_1 \ldots q_i \ldots q_Q$ and a spoken document D represented as a PSPL. Our ranking scheme follows the description in Section 3.1.

The words in the document D clearly belong to the ASR vocabulary $\mathcal{V}$ whereas the words in the query may be out-of-vocabulary (OOV). As argued in Section 2, the query-OOV rate is an important factor in evaluating the impact of having a finite ASR vocabulary on the retrieval accuracy. We assume that the words in the query are all contained in $\mathcal{V}$; OOV words are mapped to UNK and cannot be matched in any document D.

For all query terms, a 1-gram score is calculated by summing the PSPL posterior probability across all segments s and positions k. This is equivalent to calculating the expected count of a given query term q_i according to the PSPL probability distribution $P(w_k(s)|D)$ for each segment s of document D. The results are aggregated in a common value $S_{1-gram}(D, \mathcal{Q})$:

$$S(D, q_i) \quad = \quad \log\left[1 + \sum_s \sum_k P(w_k(s) = q_i|D)\right]$$

$$S_{1-gram}(D, \mathcal{Q}) = \sum_{i=1}^{Q} S(D, q_i) \qquad (3)$$

Similar to (Brin and Page, 1998), the logarithmic tapering off is used for discounting the effect of large counts in a given document.

Our current ranking scheme takes into account proximity in the form of matching N-grams present in the query. Similar to the 1-gram case, we calculate an expected tapered-count for each N-gram $q_i \ldots q_{i+N-1}$ in the query and then aggregate the results in a common value $S_{N-gram}(D, \mathcal{Q})$ for each order N:

$$S(D, q_i \ldots q_{i+N-1}) = \qquad (4) \\ \log\left[1 + \sum_s \sum_k \prod_{l=0}^{N-1} P(w_{k+l}(s) = q_{i+l}|D)\right]$$

$$S_{N-gram}(D, \mathcal{Q}) = \sum_{i=1}^{Q-N+1} S(D, q_i \ldots q_{i+N-1})$$

The different proximity types, one for each N-gram order allowed by the query length, are combined by taking the inner product with a vector of weights.

$$S(D, \mathcal{Q}) = \sum_{N=1}^{Q} w_N \cdot S_{N-gram}(D, \mathcal{Q}) \quad (5)$$

Only documents containing all the terms in the query are returned. In the current implementation the weights increase linearly with the N-gram order. Clearly, better weight assignments must exist, and as the hit types are enriched beyond using just N-grams, the weights will have to be determined using machine learning techniques.

It is worth noting that the transcription for any given segment can also be represented as a PSPL with exactly one word per position bin. It is easy to see that in this case the relevance scores calculated according to Eq. (3-4) are the ones specified by 3.1.

6 Experiments

We have carried all our experiments on the iCampus corpus prepared by MIT CSAIL. The main advantages of the corpus are: realistic speech recording conditions — all lectures are recorded using a lapel microphone — and the availability of accurate manual transcriptions — which enables the evaluation of a SDR system against its text counterpart.

6.1 iCampus Corpus

The iCampus corpus (Glass et al., 2004) consists of about 169 hours of lecture materials: 20 Introduction to Computer Programming Lectures (21.7 hours), 35 Linear Algebra Lectures (27.7 hours), 35 Electro-magnetic Physics Lectures (29.1 hours), 79 Assorted MIT World seminars covering a wide variety of topics (89.9 hours). Each lecture comes with a word-level manual transcription that segments the text into semantic units that could be thought of as sentences; word-level time-alignments between the transcription and the speech are also provided. The speech style is in between planned and spontaneous. The speech is recorded at a sampling rate of 16kHz (wide-band) using a lapel microphone.

The speech was segmented at the sentence level based on the time alignments; each lecture is considered to be a spoken document consisting of a set of one-sentence long segments determined this way — see Section 5.1. The final collection consists of 169 documents, 66,102 segments and an average document length of 391 segments.

We have then used a standard large vocabulary ASR system for generating 3-gram ASR lattices and PSPL lattices. The 3-gram language model used for decoding is trained on a large amount of text data, primarily newswire text. The vocabulary of the ASR system consisted of 110kwds, selected based on frequency in the training data. The acoustic model is trained on a variety of wide-band speech and it is a standard clustered tri-phone, 3-states-per-phone model. Neither model has been tuned in any way to the iCampus scenario.

On the first lecture `L01` of the Introduction to Computer Programming Lectures the WER of the ASR system was 44.7%; the OOV rate was 3.3%. For the entire set of lectures in the Introduction to Computer Programming Lectures, the WER was 54.8%, with a maximum value of 74% and a minimum value of 44%.

6.2 PSPL lattices

We have then proceeded to generate 3-gram lattices and PSPL lattices using the above ASR system. Table 1 compares the accuracy/size of the 3-gram lattices and the resulting PSPL lattices for the first lecture `L01`. As it can be seen the PSPL representation is much more compact than the original 3-gram lattices at a very small loss in accuracy: the 1-best path through the PSPL lattice is only 0.3% absolute worse than the one through the original 3-gram lattice. As expected, the main reduction comes from the drastically smaller node density — 7 times smaller, measured in nodes per word in the reference transcription. Since the PSPL representation

Lattice Type	3-gram	PSPL
Size on disk	11.3MB	3.2MB
Link density	16.3	14.6
Node density	7.4	1.1
1-best WER	44.7%	45%
ORACLE WER	26.4%	21.7%

Table 1: Comparison between 3-gram and PSPL lattices for lecture L01 (iCampus corpus): node and link density, 1-best and ORACLE WER, size on disk

introduces new paths compared to the original 3-gram lattice, the ORACLE WER path — least errorful path in the lattice — is also about 20% relative better than in the original 3-gram lattice — 5% absolute. Also to be noted is the much better WER in both PSPL/3-gram lattices versus 1-best.

6.3 Spoken Document Retrieval

Our aim is to narrow the gap between speech and text document retrieval. We have thus taken as our reference the output of a standard retrieval engine working according to one of the TF-IDF flavors, see Section 3. The engine indexes the manual transcription using an unlimited vocabulary. All retrieval results presented in this section have used the standard `trec_eval` package used by the TREC evaluations.

The PSPL lattices for each segment in the spoken document collection were indexed as explained in 5.1. In addition, we generated the PSPL representation of the manual transcript and of the 1-best ASR output and indexed those as well. This allows us to compare our retrieval results against the results obtained using the reference engine when working on the same text document collection.

6.3.1 Query Collection and Retrieval Setup

The missing ingredient for performing retrieval experiments are the queries. We have asked a few colleagues to issue queries against a demo shell using the index built from the manual transcription. The only information[1] provided to them was the same as the summary description in Section 6.1.

We have collected 116 queries in this manner. The query out-of-vocabulary rate (Q-OOV) was 5.2% and the average query length was 1.97 words. Since our approach so far does not index sub-word units, we cannot deal with OOV query words. We have thus removed the queries which contained OOV words — resulting in a set of 96 queries — which clearly biases the evaluation. On the other hand, the results on both the 1-best and the lattice indexes are equally favored by this.

6.3.2 Retrieval Experiments

We have carried out retrieval experiments in the above setup. Indexes have been built from:

- `trans`: manual transcription filtered through ASR vocabulary
- `1-best`: ASR 1-best output
- `lat`: PSPL lattices.

No tuning of retrieval weights, see Eq. (5), or link scoring weights, see Eq. (2) has been performed. Table 2 presents the results. As a sanity check, the retrieval results on transcription — `trans` — match almost perfectly the reference. The small difference comes from stemming rules that the baseline engine is using for query enhancement which are not replicated in our retrieval engine. The results on lattices (`lat`) improve significantly on (`1-best`) — 20% relative improvement in mean average precision (MAP).

	`trans`	`1-best`	`lat`
# docs retrieved	1411	3206	4971
# relevant docs	1416	1416	1416
# rel retrieved	1411	1088	1301
MAP	0.99	0.53	0.62
R-precision	0.99	0.53	0.58

Table 2: Retrieval performance on indexes built from transcript, ASR 1-best and PSPL lattices, respectively

6.3.3 Why Would This Work?

A legitimate question at this point is: *why would anyone expect this to work when the 1-best ASR accuracy is so poor?*

In favor of our approach, the ASR lattice WER is much lower than the 1-best WER, and PSPL have even lower WER than the ASR lattices. As reported in Table 1, the PSPL WER for `L01` was 22% whereas the 1-best WER was 45%. Consider matching a 2-gram in the PSPL —the average query length is indeed 2 wds so this is a representative situation. A simple calculation reveals that it is twice — $(1 - 0.22)^2/(1 - 0.45)^2 = 2$ — more likely to find a query match in the PSPL than in the 1-best — if the query 2-gram was indeed spoken at that position. According to this heuristic argument one could expect a dramatic increase in Recall. Another aspect

[1] Arguably, more motivated users that are also more familiar with the document collection would provide a better query collection framework

is that people enter *typical N-grams* as queries. The contents of adjacent PSPL bins are fairly random in nature so if a typical 2-gram is found in the PSPL, chances are it was actually spoken. This translates in little degradation in Precision.

7 Conclusions and Future work

We have developed a new representation for ASR lattices — the Position Specific Posterior Lattice (**PSPL**) — that lends itself naturally to indexing speech content and integrating state-of-the-art IR techniques that make use of *proximity and context* information. In addition, the PSPL representation is also much more compact at no loss in WER — both 1-best and ORACLE.

The retrieval results obtained by indexing the PSPL and performing adequate relevance ranking are 20% better than when using the ASR 1-best output, although still far from the performance achieved on text data.

The experiments presented in this paper are truly a first step. We plan to gather a much larger number of queries. The binary relevance judgments — a given document is deemed either relevant or irrelevant to a given query in the reference "ranking" — assumed by the standard `trec_eval` tool are also a serious shortcoming; a distance measure between *rankings* of documents needs to be used. Finally, using a baseline engine that in fact makes use of proximity and context information is a priority if such information is to be used in our algorithms.

8 Acknowledgments

We would like to thank Jim Glass and T J Hazen at MIT for providing the iCampus data. We would also like to thank Frank Seide for offering valuable suggestions and our colleagues for providing queries.

References

Ricardo Baeza-Yates and Berthier Ribeiro-Neto, 1999. *Modern Information Retrieval*, chapter 2, pages 27–30. Addison Wesley, New York.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

M. G. Brown, J. T. Foote, G. J. F. Jones, K. Spärck Jones, and S. J. Young. 1996. Open-vocabulary speech indexing for voice and video mail retrieval. In *Proc. ACM Multimedia 96*, pages 307–316, Boston, November.

Kenneth Ward Church. 2003. Speech and language processing: Where have we been and where are we going? In *Proceedings of Eurospeech*, Geneva, Switzerland.

J. Garofolo, G. Auzanne, and E. Voorhees. 2000. The TREC spoken document retrieval track: A success story. In *Proceedings of the Recherche d'Informations Assiste par Ordinateur: ContentBased Multimedia Information Access Conference*, April.

James Glass, T. J. Hazen, Lee Hetherington, and Chao Wang. 2004. Analysis and processing of lecture audio data: Preliminary investigations. In *HLT-NAACL 2004 Workshop: Interdisciplinary Approaches to Speech Indexing and Retrieval*, pages 9–12, Boston, Massachusetts, May.

David Anthony James. 1995. *The Application of Classical Information Retrieval Techniques to Spoken Documents*. Ph.D. thesis, University of Cambridge, Downing College.

B. Logan, P. Moreno, and O. Deshmukh. 2002. Word and sub-word indexing approaches for reducing the effects of OOV queries on spoken audio. In *Proc. HLT*.

Kenney Ng. 2000. *Subword-Based Approaches for Spoken Document Retrieval*. Ph.D. thesis, Massachusetts Institute of Technology.

NIST. www. The TREC evaluation package. In *www-nlpir.nist.gov/projects/trecvid/trecvid.tools/trec_eval*.

L. R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings IEEE*, volume 77(2), pages 257–285.

Murat Saraclar and Richard Sproat. 2004. Lattice-based search for spoken utterance retrieval. In *HLT-NAACL 2004*, pages 129–136, Boston, Massachusetts, May.

F. Seide and P. Yu. 2004. Vocabulary-independent search in spontaneous speech. In *Proceedings of ICASSP*, Montreal, Canada.

Matthew A. Siegler. 1999. *Integration of Continuous Speech Recognition and Information Retrieval for Mutually Optimal Performance*. Ph.D. thesis, Carnegie Mellon University.

P. C. Woodland, S. E. Johnson, P. Jourlin, and K. Spärck Jones. 2000. Effects of out of vocabulary words in spoken document retrieval. In *Proceedings of SIGIR*, pages 372–374, Athens, Greece.

Steve Young, Gunnar Evermann, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dan Povey Dave Ollason, Valtcho Valtchev, and Phil Woodland. 2002. *The HTK Book*. Cambridge University Engineering Department, Cambridge, England, December.