# SPEECH OGLE: Indexing Uncertainty for Spoken Document Search

**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 (PSPL), a novel lossy representation of automatic speech recognition lattices that naturally lends itself to efficient indexing and subsequent relevance ranking of spoken documents.

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 inverted index built from PSPL lattices is compact — about 20% of the size of 3-gram ASR lattices and 3% of the size of the uncompressed speech — and it allows for extremely fast retrieval. Furthermore, little degradation in performance is observed when pruning PSPL lattices, resulting in even smaller indexes — 5% of the size of 3-gram ASR lattices.

## 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 private network 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 (Chelba and Acero, 2005).

## 2 Text Document Retrieval in the Early Google Approach

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

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



*Proceedings of the ACL Interactive Poster and Demonstration Sessions*,
pages 41–44, Ann Arbor, June 2005. 

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 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 a 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.

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. However, 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.

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 probability $\beta_n$ stays unchanged (Rabiner, 1989) whereas during the forward pass one needs to split the forward probability arriving at a given node $n$, $\alpha_n$, according to the length of the partial paths that start at the start node of the lattice and end at node $n$:

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

The posterior probability that a given node $n$ occurs at position $l$ is thus calculated using:

$$P(n, l|LAT) = \frac{\alpha_n[l] \cdot \beta_n}{norm(LAT)}$$

The posterior probability of a given word $w$ occurring at a given position $l$ can be easily calculated using:

$$P(w, l|LAT) = \sum_{n \ s.t. \ P(n,l)>0} P(n, l|LAT) \cdot \delta(w, word(n))$$

The Position Specific Posterior Lattice (PSPL) is nothing but a representation of the $P(w, l|LAT)$ distribution. For details on the algorithm and properties of PSPL please see (Chelba and Acero, 2005).

## 4 Spoken Document Indexing and Search 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.

The soft hits for a given word are stored as a vector of entries sorted by `(document id, segment id)`. Document and segment boundaries in this array, respectively, are stored separately in a map for convenience of use and memory efficiency. The *soft index* simply lists all hits for every word in the ASR vocabulary; each word entry can be stored in a separate file if we wish to augment the index easily as new documents are added to the collection.

### 4.1 Speech Content Relevance Ranking Using PSPL Representation

Consider a given query $\mathcal{Q} = q_1 \dots q_i \dots q_Q$ and a spoken document $D$ represented as a PSPL. Our ranking scheme follows the description in Section 2.

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 $S_{1-gram}(D, \mathcal{Q})$:

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

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

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 $q_i \dots q_{i+N-1}$ in the query and then aggregate the results in a common value $S_{N-gram}(D, \mathcal{Q})$ for each order $N$:

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

$$S_{N-gram}(D, \mathcal{Q}) = \sum_{i=1}^{Q-N+1} S(D, q_i \dots q_{i+N-1}) \quad (2)$$

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.

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

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. (1-2) are the ones specified by 2.

Only documents containing all the terms in the query are returned. We have also enriched the query language with the "quoted functionality" that allows us to retrieve only documents that contain exact PSPL matches for the quoted phrases, e.g. the query `''L M'' tools` will return only documents containing occurrences of `L M` and of `tools`.

## 5 Experiments

We have carried all our experiments on the iCampus corpus (Glass et al., 2004) 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.

The corpus consists of about 169 hours of lecture materials. 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 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. The final collection consists of 169 documents, 66,102 segments and an average document length of 391 segments.

### 5.1 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. 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. In terms of relative size on disk, the uncompressed speech for the first 20 lectures uses 2.5GB, the ASR 3-gram lattices use 322MB, and the corresponding index derived from the PSPL lattices uses 61MB.

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.

#### 5.1.1 Query Collection and Retrieval Setup

We have asked a few colleagues to issue queries against a demo shell using the index built from the manual transcription.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.

#### 5.1.2 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. Table 1 presents the results. As a sanity check,

| | `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 1: Retrieval performance on indexes built from transcript, ASR 1-best and PSPL lattices

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). Table 2 shows the retrieval accuracy results as well as the index size for various pruning thresholds applied to the `lat` PSPL. MAP performance increases with PSPL depth, as expected. A good compromise between accuracy and index size is obtained for a pruning threshold of 2.0: at very little loss in MAP one could use an index that is only 20% of the full index.

## 6 Conclusions and Future work

We have developed a new representation for ASR lattices — the Position Specific Posterior Lattice — that lends itself to indexing speech content. The retrieval results obtained by indexing the PSPL are 20% better than when using the ASR 1-best output.

| pruning threshold | MAP | R-precision | Index Size (MB) |
|---|---|---|---|
| 0.0 | 0.53 | 0.54 | 16 |
| 0.1 | 0.54 | 0.55 | 21 |
| 0.2 | 0.55 | 0.56 | 26 |
| 0.5 | 0.56 | 0.57 | 40 |
| 1.0 | 0.58 | 0.58 | 62 |
| 2.0 | 0.61 | 0.59 | 110 |
| 5.0 | 0.62 | 0.57 | 300 |
| 10.0 | 0.62 | 0.57 | 460 |
| 1000000 | 0.62 | 0.57 | 540 |

Table 2: Retrieval performance on indexes built from pruned PSPL lattices, along with index size

The techniques presented can be applied to indexing contents of documents when uncertainty is present: optical character recognition, handwriting recognition are examples of such situations.

## 7 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 on our work.

## References

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.

Ciprian Chelba and Alex Acero. 2005. Position specific posterior lattices for indexing speech. In *Proceedings of ACL*, Ann Arbor, Michigan, June.

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

James Glass, Timothy 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, USA, May 6.

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.