Indexing Uncertainty for Spoken Document Search

Ciprian Chelba, Alex Acero

Speech Research Group, Microsoft Research
Microsoft Corp., One Microsoft Way, Redmond, WA 98052
{chelba,alexac}@microsoft.com

Abstract

The paper presents the Position Specific Posterior Lattice, a novel lossy representation of automatic speech recognition lattices that naturally lends itself to efficient indexing and subsequent relevance ranking of spoken documents. Albeit lossy, the PSPL lattice is much more compact than the ASR 3-gram lattice from which it is computed, 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.

In experiments performed on a collection of lecture recordings — MIT iCampus database — 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.

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 emerges as a key application as more and more data is being saved [1].

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 becomes 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 ubiquitous 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.

2. Previous Work

The main research effort aiming at spoken document retrieval (SDR) was centered around the SDR-TREC evaluations [2], although there had been a large body of work in this area prior to the SDR-TREC evaluations, most notable being the contributions of [3] and [4]. In the TREC-SDR 8/9 evaluations, SDR systems indexed the ASR 1-best output and their retrieval performance — measured in terms of MAP [5] — 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.

However there are shortcomings to the SDR-TREC framework. The recognizers were heavily tuned for the domain leading to very good ASR performance. It is well known that ASR is very brittle to mismatched training/test conditions and it is unrealistic to expect error rates of 10-15% when decoding speech mismatched relative to the training data. It is crucial to explore the effects of higher WER.

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 unrealistically low. [6] 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.

As pointed out in [7], word level indexing is more accurate but plagued by the OOV problem. The authors argue in favor of a combination of word and sub-word level indexing. A similar approach is taken by [8].

More recently, [9] 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.

3. Text Document Retrieval

Probably the most widespread text retrieval model is the TF-IDF vector model [10]. The main criticism to TF-IDF is that the query terms are assumed to be independent. *Proximity information* is not taken into account, e.g. whether the words LANGUAGE and MODELING occur next to each other or not in a document is not used for relevance scoring; returning only documents that contain the phrase "LANGUAGE MODELING" is impossible. Another issue is that query terms may be encountered in various *contexts* in a given document: title, abstract, author name, font size, etc. TF-IDF ranking 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, [11] also uses both *proximity* and *context* information heavily when assigning a relevance score to a given document (see [11], 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*.

For a single word query, the [11] 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 "close" and "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.

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.

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 representation introduced in the next section addresses this issue.

4.1. Position Specific Posterior Lattices

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 prob)` 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 that contain a given word at a given position. A simple dynamic programming algorithm which is a variation on the standard forward-backward algorithm can be employed for performing this computation. During the forward pass one needs to split the forward probability arriving at a given node n, α_n, according to the length of the partial paths that start at the start node of the lattice and end at node n. For details the reader is referred to [12].

The "log-probability" $\log P(\cdot)$ of a link u is "flattened" using a FLATw ≥ 1.0 weight:

$$\begin{aligned}\log P(u) = \text{FLATw } \cdot [1/\text{LMw} \cdot \log P_{AM}(u) + \\ \log P_{LM}(word(u)) - 1/\text{LMw} \cdot logP_{IP}]\end{aligned} \qquad (1)$$

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

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

In N-gram lattices, $N \geq 2$, all links ending at a given node n must contain the same word $word(n)$, so the posterior probability of word w occurring at position l can be easily calculated using:

$$\begin{aligned}&P(w, l|LAT) = \\ &\quad \textstyle\sum_{n \ s.t. \ P(n,l)>0} P(n, l|LAT) \cdot \delta(w, word(n))\end{aligned}$$

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.

5. Spoken Document Indexing and Search

Speech content can be very long. In our case the speech content of a typical spoken document was approximately 1 hour long. It is customary to segment a given speech file in shorter segments. Each *soft hit* in our index will store `position` in segment and `posterior probability`. The soft hits for a given word are stored as a vector of entries sorted by `(document id, segment id)`. The *soft index* simply lists all hits for every word in the ASR vocabulary.

5.1. Relevance Ranking Using PSPL

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.

For all query terms, a 1-gram score is calculated by summing the PSPL posterior probability across all segments s and positions k. The results are aggregated in a common value $S_{1-gram}(D, \mathcal{Q})$; similar to [11], logarithmic tapering off is used for discounting the effect of large counts in a document:

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

$$S_{1-gram}(D, \mathcal{Q}) = \sum_{i=1}^{Q} S(D, q_i) \qquad (2)$$

Our current ranking scheme takes into account proximity in the form of matching N-grams present in the query. 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:

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

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

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; in the current implementation the weights increase linearly with N.

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

6. Experiments

We have carried all our experiments on the iCampus corpus [13] prepared by MIT CSAIL. It consists of about 169 hours of lecture material recorded in the classroom:

- 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 (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 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 110k words, 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%.

6.1. PSPL lattices

We generated 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`. 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.

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 of the iCampus corpus: node and link density, 1-best and “oracle” WER, size on disk

6.2. Spoken Document Retrieval

Our aim is to narrow the gap between speech and text document retrieval. We have thus taken as 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.

One problem with this evaluation framework is that the reference TF-IDF ranking results are not using any proximity information and thus we cannot fully evaluate our ranking framework. A better baseline ranking engine is clearly desirable.

6.2.1. *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. The only information provided to them was the same as the summary description in Section 6.

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. The results on both the 1-best and the lattice indexes are equally favored by this, so the relative performance of one over the other is likely to be same after dealing properly with the OOV query words — see Section 6.2.3.

6.2.2. *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; the flattening weight FLATw in Eq. (1) was set to 1.0 resulting in a smoother PSPL distribution. Table 2 presents the results. As a sanity check, the retrieval results on transcription — `trans` — match almost perfectly the reference[1]. 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

In order to gauge the sensitivity of the system to the accuracy of the PSPL distribution we have experimented with the

[1] The small difference comes from stemming rules that the baseline engine is using for query enhancement which are not replicated in our retrieval engine

following variations:

- `lat`: FLATw $= 1.0$ smooth PSPL
- `raw`: FLATw $=$ LMw $= 13$ skewed PSPL
- `noP`: the words in a given PSPL bin receive posterior "probability" 1.0 — resulting in a "hard-index"
- `unif`: the words in a given PSPL bin receive uniform posterior probability $1.0/\#entries$.

	lat	raw	noP	unif
# docs retrieved	4971	4971	4971	4971
# relevant docs	1416	1416	1416	1416
# rel retrieved	1301	1301	1301	1301
MAP	0.62	0.60	0.47	0.57
R-precision	0.58	0.56	0.42	0.52

Table 3: Retrieval performance on indexes built from PSPL lattices under various PSPL probability assignments

Table 3 presents the results. The retrieval results are very sensitive to large variations in the PSPL distribution. In particular, ignoring the PSPL probability distribution altogether (`noP`) leads to worse results than using the 1-best. Also, flattening the ASR lattice scores (see Eq. 1) by using FLATw $= 1.0$ has a small positive impact on the retrieval accuracy.

6.2.3. Out-of-Vocabulary Query Words

In order to gauge the effect of using a truncated vocabulary we used our soft-indexing method to index the *manual transcription* using two vocabularies[2]:

- CLOSED-INDEX: closed vocabulary derived from the manual transcriptions and the full query set (*before removing the queries containing OOV words*, see Section 6.2.1)
- OPEN-INDEX: ASR vocabulary, OOV words are mapped to `<unk>`

The query set used for testing did not discard the queries containing OOV words. The effect of using the finite ASR vocabulary is to lump all OOV words into one type, and thus reduce the retrieval accuracy.

When scoring the output on OPEN-INDEX against the output on the CLOSED-INDEX reference, MAP is 0.92 and R-precision is 0.91. A substantial hit in performance is to be expected when using a vocabulary of finite size, unless we have a good way of spotting and scoring OOV words.

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. 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 techniques developed here can be applied to indexing documents in the presence of uncertainty over their contents: handwriting and optical character recognition are examples of such situations.

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.

9. References

[1] K. W. Church, "Speech and language processing: Where have we been and where are we going?" in *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[2] J. Garofolo, G. Auzanne, and E. Voorhees, "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 2000. [Online]. Available: citeseer.ist.psu.edu/garofolo00trec.html

[3] M. G. Brown, J. T. Foote, G. J. F. Jones, K. S. Jones, and S. J. Young, "Open-vocabulary speech indexing for voice and video mail retrieval," in *Proc. ACM Multimedia 96*, Boston, November 1996, pp. 307–316.

[4] D. A. James, "The application of classical information retrieval techniques to spoken documents," Ph.D. dissertation, University of Cambridge, Downing College, 1995.

[5] NIST, "The TREC evaluation package (see README file)." [Online]. Available: www-nlpir.nist.gov/projects/trecvid/trecvid.tools/trec_eval

[6] P. C. Woodland, S. E. Johnson, P. Jourlin, and K. S. Jones, "Effects of out of vocabulary words in spoken document retrieval," in *Proceedings of SIGIR*, Athens, Greece, 2000, pp. 372–374.

[7] B. Logan, P. Moreno, and O. Deshmukh, "Word and sub-word indexing approaches for reducing the effects of OOV queries on spoken audio," in *Proc. HLT*, 2002. [Online]. Available: citeseer.nj.nec.com/585562.html

[8] F. Seide and P. Yu, "A hybrid word/phoneme-based approach for improved vocabulary-independent search in spontaneous speech," in *Proceedings of ICSLP*, Jeju, Korea, 2004.

[9] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *HLT-NAACL 2004: Main Proceedings*, Boston, Massachusetts, USA, May 2 - May 7 2004, pp. 129–136.

[10] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: Addison Wesley, 1999, ch. 2, pp. 27–30.

[11] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998. [Online]. Available: citeseer.ist.psu.edu/brin98anatomy.html

[12] C. Chelba and A. Acero, "Position specific posterior lattices for indexing speech," in *Proceedings of ACL*, June 2005.

[13] J. Glass, T. J. Hazen, L. Hetherington, and C. Wang, "Analysis and processing of lecture audio data: Preliminary investigations," in *HLT-NAACL 2004 Workshop: Interdisciplinary Approaches to Speech Indexing and Retrieval*, Boston, Massachusetts, USA, May 6 2004, pp. 9–12.

[2] It can be easily checked that the PSPL representation for a lattice containing exactly one path is identical to the original lattice, where posterior probabilities for each word are 1.0; the "soft-index" is thus identical to the regular index used for text documents.