

PRUNING ANALYSIS FOR THE POSITION SPECIFIC POSTERIOR LATTICES FOR SPOKEN DOCUMENT SEARCH

Jorge Silva

Speech Analysis and Interpretation Laboratory
University of Southern California
jorgesil@usc.edu

Ciprian Chelba and Alex Acero

Speech Research Group
Microsoft Corporation
{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.

Two pruning techniques for generating word lattices are explored in this framework, where experiments performed on a collection of lecture recordings — MIT iCampus database — show that the spoken document ranking accuracy was improved by 20% — in the mean average precision sense — relative over the commonly used baseline of indexing the 1-best output from an automatic speech recognizer (ASR).

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.

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%. However there are shortcomings to the SDR-TREC framework: the recognizers were heavily tuned for the domain leading to very good ASR performance

— 10-15% WER, very close to that of the closed captioning text used as reference for retrieval accuracy evaluations. The effect of higher WER needs to be explored.

Position Specific Posterior Lattice (PSPL) was proposed as a way to extend the key-word search paradigm from text documents to spoken documents [6] in scenarios with high WER. The approach calculates posterior probabilities of words at a given integer position —soft indexing — as a way to model the uncertainty of the spoken content, and significantly reduce the size of the ASR lattice. The position information is used for incorporating proximity in the scoring paradigm by allowing the calculation of distance- k skip n -gram expected counts strictly based on the inverted index.

A similar approach was presented by Saraclar et al [7]: it indexes every arc in the ASR lattice (no compression), which allows for exact calculation of n -gram expected counts but more general proximity information (distance- k skip n -gram, $k > 0$) is hard to calculate. Their evaluation is focused on word-spotting rather than document retrieval performance.

In this paper, relative and absolute PSPL pruning techniques are proposed for controlling and evaluating the information transfer process from the ASR to the spoken document retrieval framework. Those techniques were contrasted in term of the precision-recall performance metrics as a function of pruning thresholds. It is shown that those techniques provide flexibility to adjust the precision-recall metrics for particular application and user needs. On the other hand, this work corroborates that the PSPL framework outperforms the 1-best approach and can be considered a better way of modeling the spoken content uncertainty, particularly relevant for high WER scenarios.

2. POSITION SPECIFIC POSTERIOR LATTICES

Of essence to fast retrieval on static text 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. This is motivated by the early Google approach [8] where context and proximity issues were shown to be important factors for doing the relevance ranking score [6].

If we want to extend this direction to spoken documents, 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 precision-recall metrics. 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 needed for recording a given word hit is not readily available in ASR lattices.

Let’s consider that a traditional text-document hit for given word consists of just (document id, position). In this context, 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 possible 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. A dynamic programming algorithm was proposed for performing this computation [6]. 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 :

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

The backward probability β_n has the standard definition, where the dynamic recursion for $\alpha_n[l]$ is formally presented in [6].

Using these forward-backward variables the posterior probability of a given word w occurring at a given position l in the lattice can be easily calculated using:

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

Finally, 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 $P(w, l|LAT)$.

3. SPOKEN DOCUMENT INDEXING AND SEARCH USING PSPL

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.

Consider a given query $\mathcal{Q} = q_1 \dots q_i \dots q_Q$ and a spoken document D represented as a PSPL. The possible word sequences in the document D clearly belong to the ASR vocabulary \mathcal{V} whereas the words in the query may be out-of-vocabulary (OOV). 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 $S_{1\text{-gram}}(D, \mathcal{Q})$:

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

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

where similar to [8], 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. 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 $S_{N\text{-gram}}(D, \mathcal{Q})$ for each order N :

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

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

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

$$S(D, \mathcal{Q}) = \sum_{N=1}^Q w_N \cdot S_{N\text{-gram}}(D, \mathcal{Q}) \quad (3)$$

Only documents containing all the terms in the query are returned. In the current implementation the weights increase linearly with the N -gram order.

4. PRUNING TECHNIQUES

The performance of the PSPL framework is evaluated applying a relative and absolute pruning techniques, respectively.

The idea is to explore how the PSPL ranking performance behaves as a function of the level of uncertainty transferred from the ASR part. For doing that, the idea is to prune the PSPL probability distribution before the generation of the N-gram expected counts.

4.1. Relative Pruning

For a given PSPL position bin k , the relative pruning first finds the most likely entry given by:

$$w_k^* = \arg \max_{w \in \mathcal{V}} P(w_k(s) = w|D)$$

and then it retains the set of PSPL entries associated to the same bin position, W_k , whose log-probability is greater than the most likely minus a predefined threshold τ_r , Eq.(4).

$$W_k = \left\{ w \in \mathcal{V} : \log \frac{P(w_k(s) = w_k^*|D)}{P(w_k(s) = w|D)} \leq \tau_r \right\} \quad (4)$$

where τ_r can take values in $[0, \infty)$.

The remaining entries are renormalized to make it a proper probability mass function and used to calculate the expected N-gram counts, Eq.(1) and (2).

This approach reduces the support of the PSPL bin distribution $\{P(w_k(s) = q|D)\}_{q \in \mathcal{V}}$, concentrating the probability mass on the more likely bin entries. Note that when the threshold tends to zero we reduce to the PSPL 1-best, which is marginally different from the 1-best of the original word lattice, see Table 1.

4.2. Absolute Pruning

In this case, the PSPL framework considers the PSPL entries whose log-probability is higher than an absolute threshold. More precisely, for a given position k a truncated posterior “distribution”¹ $\bar{P}(w_k(s) = q|D)$ is used in the process of computing N-gram expected counts, Eq.(1) and (2), where $\bar{P}(w_k(s) = q|D)$ is given by:

$$\begin{aligned} \bar{P}(w_k(s) = q|D) = \\ P(w_k(s) = q|D) \cdot \mathbf{1}_{\{\log P(w_k(s)=q|D) \geq \tau_{abs}\}} \end{aligned} \quad (5)$$

τ_{abs} represents the absolute confidence threshold taking values in $(-\infty, 0]$.

The threshold is absolute and consequently when τ_{abs} is relatively close to zero, the PSPL contains only the bin entries that have high level of confidence, and some position bins become empty.

5. EXPERIMENTS

All our experiments were conducted on the iCampus corpus [9] prepared by MIT CSAIL. It consists of about 169 hours of lecture material recorded in the classroom. The corpus contains 90 Lectures (78.5 hours) and 79 Assorted MIT World

¹No longer a proper probability distribution

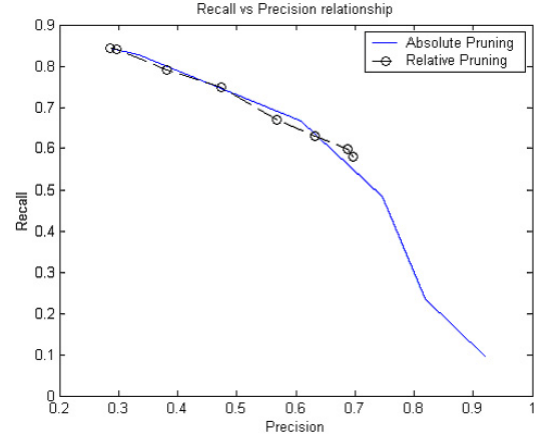


Fig. 1. Recall vs Precision for the relative and absolute threshold techniques; 1-best result is highest Precision on the relative pruning curve.

seminars (89.9 hours). 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 set, Introduction to Computer Programming Lectures (21.7 hours), the WER of the ASR system was 44.7%; the OOV rate was 3.3%. We generated 3-gram lattices and PSPL lattices using the above ASR system.

For the queries 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. 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.

Finally as a reference for evaluation we have taken the output of a standard retrieval engine working according to one of the TF-IDF flavors [10]. 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.

5.1. Pruning analysis

Figure 1 presents the precision-recall graphs using the relative and absolute pruning techniques presented in Section 4. A wide range of threshold magnitudes were used to explore

MAP	R-precision	τ_r Pruning Threshold
0.529	0.538	0
0.540	0.549	0.1
0.582	0.578	1.0
0.612	0.591	2.0
0.622	0.596	3.0
0.623	0.577	5.0
0.620	0.573	100.0

Table 1. Retrieval performance using relative threshold. Zero threshold represents the result for the 1-best approach.

a representative range of precision-recall points in these two pruning settings.

In both scenarios the PSPL bin density increases with the absolute magnitude of the threshold, inducing the following trade-off in precision and recall: on one hand, we have more chances that the unknown document transcription is part of the set of PSPL bin entries for that document, which has a positive impact in recall. On the other hand, we increase the number of non-valid PSPL entries for a document (entries whose word index is not part of the document transcription), which in average has a negative effect in precision. This behavior explains in part the precision - recall evolution presented in Figure 1 for both techniques. It is important to note that the right-most point on the relative pruning curve —relative threshold equal to 0— represents the 1-best results.

As expected, both pruning techniques provide an extra degree of freedom allowing, relative to the 1-best approach, to explore different recall-precision performances points. Their performance is similar in the range of [0.3, 0.7] -precision. However, the absolute pruning provides a wider range of precision - recall trade-offs. The absolute pruning allows the ranking framework to be based on PSPL entries that have arbitrary high level of confidence. In that process, this approach provides the flexibility to reach very high precision, independent of the ASR performance. When using the 1-best PSPL entries — low relative pruning threshold values— those entries may not have the level of confidence necessary to reach high precision values, as can be seen in Figure 1. It is important to mention that this extra flexibility comes at no cost in performance at higher recall values.

Finally, Tables 1 and 2 show the mean average precision (MAP) and R-precision as a function of different threshold magnitudes. The PSPL ranking using both threshold settings shows scenarios with 20% relative improvement with respect to the 1-best approach — obtained for a 0 value of the relative pruning threshold, Table 1. In this context, there is no significant difference between the best possible scenario of either technique.

6. CONCLUSION

The PSPL framework provides an alternative way of dealing with the document’s content uncertainty for spoken document

MAP	R-precision	τ_{abs} Confidence Threshold
0.119	0.109	-0.1
0.241	0.240	-0.5
0.454	0.467	-1.0
0.596	0.598	-2.0
0.626	0.582	-5.0
0.620	0.572	-1000.0

Table 2. Retrieval performance using absolute threshold

retrieval. This technique explicitly takes into consideration the content uncertainty by means of using N-gram expected counts (*soft-hits*) and at the same time introducing proximity issues in the score formulation.

The proposed pruning techniques provide ways of controlling the number of possible entries in the PSPL, based on their log-probability. In particular, the absolute pruning provides a way of deciding the level of confidence that the ASR information needs to have for generating the relevant ranking score. This is useful to adjust precision-recall performance metrics to the user needs.

7. REFERENCES

- [1] Kenneth Ward 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 Assistée par Ordinateur: ContentBased Multimedia Information Access Conference*, April 2000.
- [3] M. G. Brown, J. T. Foote, G. J. F. Jones, K. Spärck 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] David Anthony James, *The Application of Classical Information Retrieval Techniques to Spoken Documents*, Ph.D. thesis, University of Cambridge, Downing College, 1995.
- [5] NIST, “The TREC evaluation package,” in [www-nlp.nist.gov/projects/trecvid/trecvid.tools/trec_eval](http://www.nlp.nist.gov/projects/trecvid/trecvid.tools/trec_eval), www.
- [6] Ciprian Chelba and Alex Acero, “Position specific posterior lattices for indexing speech,” in *Proceedings of ACL*, Ann Arbor, Michigan, June 2005.
- [7] Murat Saraclar and Richard Sproat, “Lattice-based search for spoken utterance retrieval,” in *HLT-NAACL 2004*, Boston, Massachusetts, May 2004, pp. 129–136.
- [8] Sergey Brin and Lawrence 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.
- [9] T. Hazen A. Park and J. Glass, “Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling,” in *Proc. ICASSP*, Philadelphia, March 2005.
- [10] Ricardo Baeza-Yates and Berthier Ribeiro-Neto, *Modern Information Retrieval*, chapter 2, pp. 27–30, Addison Wesley, New York, 1999.