Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Computer Speech and Language 21 (2007) 458–478

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Soft indexing of speech content for search in spoken documents

Ciprian Chelba [a,*], Jorge Silva [b,c], Alex Acero [a]

[a] *Speech Research Group, Microsoft Research, One Microsoft Way, Redmond, WA 98052, United States*

[b] *Speech Analysis and Interpretation Laboratory (SAIL), Department of Electrical Engineering, Viterbi School of Engineering, University of Southern California, USA*

[c] *Electrical Engineering Department, University of Chile, Santiago, Chile*

Received 9 February 2006; received in revised form 15 June 2006; accepted 1 September 2006
Available online 2 October 2006

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. This technique explicitly takes into consideration the content uncertainty by means of using *soft-hits*. Indexing position information allows one to approximate *N*-gram expected counts and at the same time use more general proximity features in the relevance score calculation. In fact, one can easily port *any state-of-the-art text-retrieval algorithm* to the scenario of indexing ASR lattices for spoken documents, rather than using the 1-best recognition result.

Experiments performed on a collection of lecture recordings—MIT iCampus database—show that the spoken document ranking performance was improved by 17–26% relative over the commonly used baseline of indexing the 1-best output from an automatic speech recognizer (ASR).

The paper also addresses the problem of integrating speech and text content sources for the document search problem, as well as its usefulness from an ad hoc retrieval—keyword search—point of view. In this context, the PSPL formulation is naturally extended to deal with both speech and text content for a given document, where a new relevance ranking framework is proposed for integrating the different sources of information available. Experimental results on the MIT iCampus corpus show a relative improvement of 302% in Mean Average Precision (MAP) when using speech content *and* text-only metadata as opposed to just text-only metadata (which constitutes about 1% of the amount of data in the transcription of the speech content, measured in number of words). Further experiments show that even in scenarios for which the metadata size is artificially augmented such that it contains more than 10% of the spoken document transcription, the speech content still provides significant performance gains in MAP with respect to only using the text-metadata for relevance ranking.

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,

[*] Corresponding author. Currently with Google. Tel.: +1 425 739 5987.
E-mail address: ciprianchelba@google.com (C. Chelba).
URLs: http://ciprian.chelba.googlepages.com (C. Chelba), http://www-scf.usc.edu/%7Ejorgesil (J. Silva).

0885-2308/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2006.09.001

search has emerged as a key application as more and more data are 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.

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) (Chelba and Acero, 2005) 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. Due to the use of integer position bins for every word in the ASR lattice, the PSPL representation allows one to port to the speech scenario any relevance ranking function originally devised for text documents.

Another novel aspect of the current work is the evaluation methodology used: since our aim is to bridge the gap between text and speech-grade search technology, we take as reference the output of a text retrieval engine that runs each query on the manually transcribed documents, rather than the spoken ones. We then evaluate spoken document retrieval (SDR) performance by measuring Mean Average Precision (MAP) with respect to this set of relevant documents, rather than using costly manual annotations of relevant documents, as in the TREC evaluations.

Our experiments are carried out on a collection of about 200 h of speech split into roughly 170 documents made available by MIT under the iCampus project. The speech is recorded in realistic conditions using a lapel microphone and no domain-specific tuning or adaptation techniques have been employed in the results reported here, leading to a very high word error rate (WER)—approximately 50% using a state-of-the-art large vocabulary speech recognition engine trained for desktop dictation.

Experiments carried out in this setup show an encouraging 17–26% relative improvement in MAP over the baseline obtained by indexing the 1-best ASR output.

Spoken document collections usually have metadata is text information alongside the spoken documents. On one hand, the text metadata is deterministic, very limited in size, and very likely differs from the actual spoken transcription which may limit its relevance to the content of the document. On the other hand, the speech recognition output is a noisy representation of the underlying lexical content and therefore we need to deal with content document uncertainty. Consequently, an approach that optimally integrates these two sources of information is desirable.

We propose a simple method for integrating text metadata and speech content for the spoken document retrieval problem and we investigate how much performance gain is provided by the spoken document material with respect to a baseline system that uses only the text-metadata for document search.

We present experimental evidence supporting the fact that the spoken content provides significant improvement in performance—approximately 300% relative MAP—with respect to the scenario where only text metadata are considered for relevance ranking. Surprisingly, this result is obtained using an ASR system with high WER.

2. Related work

The main research effort aiming at 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 does not always evaluate performance from a document retrieval point of view—using a metric like MAP or similar—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 h 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% for video and 7.5% for 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.[1]

In order to evaluate the accuracy of retrieval systems, search queries—"topics"—along with binary relevance judgments were compiled by human assessors for each of the 21,574 retrieval units—"documents".

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.

The recognizers are heavily tuned for the domain leading to very good ASR performance. 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 work at an ASR operating point which has higher WER. In our work we have used a standard dictation ASR engine whose language model has been trained on newswire text and the acoustic model was trained on wide-band continuous speech, see Section 4.1 for details, resulting in an ASR operating point of 50% WER.[2]

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 scenario, 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 ability to deal in an effective way with OOV query words is an important issue. The most common approach 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. His approach was to index phone *n*-grams appearing in ASR N-best lists. This work also focused on Broadcast News speech, thus benefiting from unrealistically superior ASR performance. 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 and thus more desirable, 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 on the entire set of documents after which the costly detailed phonetic match is carried out on only 15% of the documents in the collection.

[1] The distribution of errors in manual transcriptions or closed captionings can be very different from the ASR errors, and may have a very different impact on retrieval performance.

[2] This is a state-of-the-art recognizer trained for desktop dictation: the language model was trained on a large amount of newswire data and the acoustic model is trained on carefully articulated, wide-band speech; the recognizer achieves a WER of about 10% on matched test data.

More recently, Saraclar and Sproat (2004) proposes an approach that builds an inverted index from ASR lattices—word or phone (sub-word) level—by storing the full connectivity information in the lattice; retrieval is performed by looking up strings of units. This approach 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. No compression of the original lattice is achieved. Their evaluation is focused on word-spotting rather than document retrieval performance.

Siegler (1999) and Saraclar and Sproat (2004) show that making use of more than just the 1-best information—N-best lists, and full ASR lattices, respectively—improves retrieval accuracy.

2.1. Confusion networks

Our soft-indexing approach using PSPL could easily use alternate representations of the ASR lattices such as the one developed by Mangu et al. (2000), where lattice links are approximately binned according to the time span of the link. Both approaches—ours as well as the one in Mangu et al. (2000)—result in approximate word proximity and adjacency representations of the original lattice but have the advantage of compressing it.

Preliminary experiments carried out in our experimental setup showed that simply collapsing all links with the same word, start and end time, respectively, the number of links in the resulting "lattice" (same as number of inverted index entries) was approximatively 40% lower than that of the PSPL.[3] Since this is only the initialization step in building confusion networks (Mangu et al., 2000), it is expected that a fully built confusion network may be even smaller. Pruning also reduces the sizes of both confusion networks and PSPL—an index whose size was 15% of the original one achieves very little degradation in MAP accuracy, see Section 3.5.

One possible shortcoming of using the confusion network representation for evaluating hit proximity and adjacency is the presence of the `epsilon` links which complicate the computation of expected counts for query N-grams or skip N-grams. Removing them would result in duplicating the same word across adjacent bins—a drawback of the PSPL representation as well, see Section 4.2, and Fig. 2.

The impact of pruning on retrieval accuracy, the effects of various approximations of hit proximity information—PSPL, confusion networks, or other methods—clearly deserve a more careful evaluation and comparison. The work presented in this paper does not aim at providing the final answer as to what is the best such approximation. Recent work attempts to investigate the different trade-offs involved (Zhou et al., 2006), but that is clearly an interesting area of future research.

An important aspect when attempting such a study is that the current evaluation framework is unable to take into account differences in the reference ranking—which in our case was obtained by running queries against the manually transcribed documents, see Sections 4.1.4 and 5. This shortcoming of the evaluation framework prohibits conclusive results regarding the effects of various lattice compression techniques on proximity and adjacency-based retrieval algorithms.

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

3. Soft indexing using position specific posterior probability lattices

3.1. Overview of text indexing and search

Probably the most widespread text retrieval model is the TF-IDF vector model (Baeza-Yates and Ribeiro-Neto, 1999). For a given query $\mathcal{Q} = q_1 \ldots q_i \ldots q_Q$ 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:

$$S(D_j, \mathcal{Q}) = \sum_{i=1}^{Q} w_{i,j}, \quad w_{i,j} = f_{i,j} \cdot idf_i \tag{1}$$

[3] Note that each entry in the inverted index built from such a "time-anchored posterior probability lattice" (TAPPL) needs to store both start and end time, which requires more storage than a integer position as in the case of PSPL.

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. For example, the fact that the words LANGUAGE and MODELING occur next to each other (or not) in a given document is not used for relevance scoring, although the occurrence of the bigram LANGUAGE MODELING is probably more relevant than the combined occurrences of LANGUAGE and MODELING as unigrams. Moreover, the ability to evaluate proximity of query terms on the document side becomes critical if one is willing to enhance the query language such that quoted functionality is to be allowed, e.g. find only documents that contain the phrase "LANGUAGE MODELING".

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,[4] 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.

3.1.1. Early Google approach

Aside from the use of PageRank for relevance ranking, the early Google approach also uses both *proximity* and *context* information heavily when assigning a relevance score to a given document, see 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 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.

3.1.2. 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).

3.2. Position specific posterior probability lattices

As highlighted in the previous section, position information is crucial for being able to evaluate proximity 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

[4] Text describing the hypertext link pointing to the given document/web page.

are adjacent but questions about the distance in number of links between the occurrences of two query words in the lattice are 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`)—a pair of integers identifying the document and the position of the query word in the document, respectively.

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. The `posterior probability` refers to the posterior probability of word w occuring at position `position`, as opposed to the traditional lattice arc posterior probability (regardless of position). Since it is likely that more than one path contains the same word w 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 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. (2), respectively—see Fig. 1 for notation:

$$\begin{aligned} \alpha_n[l+1] &= \sum_{i=1}^{q} \alpha_{s_i}[l + \delta(l_i, \epsilon)] \cdot P(l_i) \\ \alpha_{\text{start}}[l] &= \begin{cases} 1.0, l = 0 \\ 0.0, l \neq 0 \end{cases} \end{aligned} \tag{2}$$

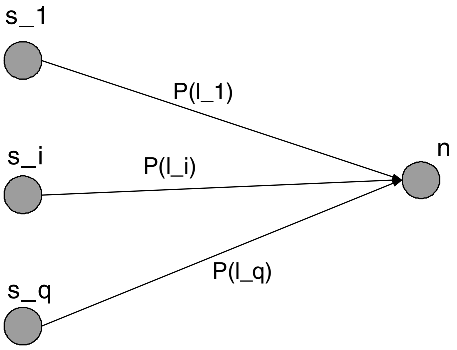

Fig. 1. State transitions.

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

$$\log P(l_i) = FLATw \cdot [1/LMw \cdot \log P_{\mathrm{AM}}(l_i) + \log P_{\mathrm{LM}}(word(l_i)) - 1/LMw \cdot \log P_{\mathrm{IP}}] \tag{3}$$

where $\log P_{\mathrm{AM}}(l_i)$ is the acoustic model score, $\log P_{\mathrm{LM}}(word(l_i))$ is the language model score, $LMw > 0$ is the language model weight, $\log P_{\mathrm{IP}} > 0$ is the "insertion penalty" and $FLATw$ is a flattening weight. In N-gram lattices where $N \geqslant 2$, all links ending at a given node n must be labeled with 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:

$$P(w, l|LAT) = \sum_{n\, s.t.\, \alpha_n[l]\cdot\beta_n > 0} \frac{\alpha_n[l] \cdot \beta_n}{\beta_{\mathrm{start}}} \cdot \delta(w, word(n)) \tag{4}$$

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

3.3. 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 hour 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:

- `position in segment` (integer value)
- `posterior probability` (floating point or quantized value)

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.

3.3.1. 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.

The recognized word sequences for the document D clearly belong to the ASR vocabulary $\mathcal{V}$ whereas the words in the query may be out-of-vocabulary (OOV).

Throughout this paper we ignore the out-of-vocabulary problem, and assume that the words in the query are all contained in $\mathcal{V}$; OOV query words are thus 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 $S_{1\text{-}gram}(D, \mathcal{Q})$:

$$\begin{aligned} S(D, q_i) &= \log\left[1 + \sum_s \sum_k P(w_k(s) = q_i|D)\right] \\ S_{1\text{-}gram}(D, \mathcal{Q}) &= \sum_{i=1}^{Q} S(D, q_i) \end{aligned} \tag{5}$$

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

In order to take into account proximity information, this ranking scheme attempts to match N-grams present in the query. Similar to the 1-gram case, an expected tapered-count is calculated for each N-gram

$q_i, \ldots, q_{i+N-1}$ in the query and then the results are aggregated in a common value $S_{N\text{-}gram}(D, \mathcal{Q})$. This is repeated for each order N allowed by the query length, resulting in relevance scores at different proximity types.

$$\begin{aligned} S(D, q_i \ldots q_{i+N-1}) &= \log\left[1 + \sum_s \sum_k \prod_{l=0}^{N-1} P(w_{k+l}(s) = q_{i+l}|D)\right] \\ S_{N\text{-}gram}(D, \mathcal{Q}) &= \sum_{i=1}^{Q-N+1} S(D, q_i \ldots q_{i+N-1}) \end{aligned} \tag{6}$$

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

$$S(D, \mathcal{Q}) = \sum_{N=1}^{Q} \lambda_N \cdot S_{N\text{-}gram}(D, \mathcal{Q}) \tag{7}$$

In the current implementation the weights increase linearly with the N-gram order, reflecting our intuition that higher order N-gram matches should have a higher contribution to the relevance score. No algorithm has been used for tuning the weights (or the linear slope), although this is clearly a promising research direction.

It is important to note that the transcription as well as any text-metadata can be represented as a PSPL with exactly one word per position bin and posterior probability 1.0.

Another important observation is that the PSPL representation facilitates porting of any text information retrieval algorithm to the spoken document retrieval case. For example, the relevance scores calculated according to Eqs. (5) and (6) are very similar to the ones specified in Brin and Page (1998) for text document retrieval, if they were to use the same N-gram proximity features that we are. Also, our relevance score calculation, see Eqs. (5)–(7), can easily incorporate the more general proximity features used by Brin and Page (1998).

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 query phrases, e.g. the query ``‘‘L M’’ ‘‘A S R’’`` will return only documents containing occurrences of `L M` and of `A S R`.

3.4. Spoken document ranking in the presence of text meta-data

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. Saving hit context information (type of content where the hit occurred) emerges as a natural way to enhance retrieval quality: e.g. a hit in the title is likely to deserve different treatment compared to a hit in some other part of the document.

As mentioned in the previous section, PSPL lattices can be used to represent text content as well, and consequently to naturally integrate the text metadata in a unified search framework.

As proposed in the previous section, we represent documents as collections of segments. To deal with segment types, we introduce an attribute on segments that allows for different segment categories, and which will be transferred to a hit in that particular segment as a "hit-type". As a first approximation, in our experiments we use different segment type labels for representing the `speech` content and the `text metadata` content for a given document. In other cases we may be able to split the text-metadata itself in different sub-categories (such as title, abstract, etc.) using the segment type attribute.

The following section details the use of segment types for relevance scoring.

3.4.1. Relevance ranking considering segment types

Again let's consider a given query $\mathcal{Q} = q_1 \ldots q_i \ldots q_Q$ and a spoken document D. To be more specific, the document D is a collection of segments denoted by Θ_D, where Θ_D is partitioned in different segment types, see Eq. (8):

$$D \doteq \Theta_D = \cup_{k=1}^{N_D} \Theta_D^{type\_k} \tag{8}$$

When calculating the relevance score of document D with respect to the query Q, we first calculate individual scores for the different segment types, $\forall k \in \{1,\ldots,N_D\}$, by:

$$S^{type\_k}(D,\mathscr{Q}) = \sum_{N=1}^{Q} \lambda_N \cdot S_{N\text{-}gram}^{type\_k}(D,\mathscr{Q}) \tag{9}$$

and the mixing weights are set as explained in Section 3.3.1. The N-gram scores are the generalization of Eqs. (5) and (6) for the case of having specific segment types:

$$\begin{aligned} S^{type\_k}(D,q_i\ldots q_{i+N-1}) &= \log\left[1+\sum_{s\in\Theta_D^{type\_k}}\sum_k\prod_{l=0}^{N-1}P\big(w_{k+l}(s)=q_{i+l}|D\big)\right] \\ S_{N\text{-}gram}^{type\_k}(D,\mathscr{Q}) &= \sum_{i=1}^{Q-N+1} S^{type\_k}\big(D,q_i\ldots q_{i+N-1}\big) \end{aligned} \tag{10}$$

Finally, the relevance score for document D is a linear combination of the segment-type specific ones, as shown in Eq. (11). The mixing weights in this expression, $\{\lambda_{type\_k}:k=1,\ldots,N_D\}$, allow for adjusting the contribution of each segment type to the final relevance score.

$$\hat{S}(D,\mathscr{Q}) = \sum_{k=1}^{N_D} \lambda_{type\_k} \cdot S^{type\_k}(D,\mathscr{Q}) \tag{11}$$

3.5. Pruning techniques

In this section we present the formulation of two pruning techniques for PSPL lattices (Sanchez et al., 2006), whose aim is to allow different precision–recall operation points of the spoken document retrieval system.

3.5.1. Relative pruning

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

$$w_k^* = \arg\max_{w\in\mathscr{V}} P(w_k(s)=w|D) \tag{12}$$

and then it retains the set W_k of PSPL entries at the same bin position whose log-probability is greater than the most likely one minus a predefined threshold τ_r:

$$W_k = \left\{w\in\mathscr{V} : \log\frac{P(w_k(s)=w_k^*|D)}{P(w_k(s)=w|D)} \leqslant \tau_r\right\} \tag{13}$$

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

The W_k entries are renormalized to make it a proper probability mass function and then used to calculate the expected N-gram counts, see Eqs. (5) and (6).

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

3.5.2. Absolute pruning

In this case, the pruning approach considers the PSPL entries whose log-probability is higher than an absolute threshold. More precisely, for a given position k a truncated posterior "distribution"[5] $\bar{P}(w_k(s) = q|D)$ is used in the process of computing N-gram expected counts, see Eqs. (5) and (6):

$$\bar{P}(w_k(s) = q|D) = P(w_k(s) = q|D) \cdot 1_{\{\log P(w_k(s)=q|D) \geqslant \tau_{\text{abs}}\}} \tag{14}$$

where τ_{abs} represents the absolute confidence threshold taking values in $(-\infty, 0]$; when τ_{abs} is close to zero, the PSPL contains only the bin entries that have high level of confidence; also, some position bins become empty, as confirmed by our experiments.

4. Experiments

We group our experimental results in four broad sections:

(1) a first one describing the experimental setup;
(2) one outlining empirical properties of the PSPL lattices;
(3) one containing experiments on spoken document retrieval that do not use any text meta-data;
(4) a last one describing experiments that evaluate the impact of text meta-data on spoken document retrieval accuracy.

4.1. Experimental setup

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.

4.1.1. Corpus

The iCampus corpus (Glass et al., 2004) consists of about 169 h of lecture materials:

- 20 Introduction to Computer Programming Lectures (21.7 h)
- 35 Linear Algebra Lectures (27.7 h)
- 35 Electro-magnetic Physics Lectures (29.1 h)
- 79 Assorted MIT World seminars covering a wide variety of topics (89.9 h)

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 16 kHz (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 3.4. The final collection consists of 169 documents, 66,102 segments and an average document length of 391 segments.

4.1.2. Lattice generation

We have then used a standard large vocabulary continuous speech recognition 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 110k words, selected based on frequency in the training

[5] No longer a proper probability distribution.

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 collection, the WER was 54.8%, with a maximum value of 74% and a minimum value of 44%; the standard deviation around the 54.8% mean was 6.9.

4.1.3. 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[6] provided to them was the summary description in Section 4.1.1.

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 in our favor. On the other hand, 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 the same after dealing properly with the OOV query words.

4.1.4. Evaluation metrics

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 3.1. 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.

trec_eval is unable to deal with the situation where a certain query does not return any relevant spoken documents and yet the reference result on the transcription does list a set of relevant documents for that particular query. To circumvent this limitation, we choose to return all documents in the collection when a query returns no matches and yet the reference file contains it.[7]

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.

Another problem is that both MAP and R-precision ignore the ranking on the reference side: a document is considered relevant if and only if it contains all the query terms. So a better baseline engine by itself is not enough unless the scoring function is also revised to take into account the ranking of the set of relevant documents.

The PSPL lattices for each segment in the spoken document collection were indexed as explained in Section 3.1.2. 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.

4.2. Experiments on position specific posterior lattices

We have 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. 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 in size comes from the drastically smaller node density—7 times smaller, measured in nodes per word in the reference transcription. Since the PSPL representation introduces new paths compared to the original 3-gram lattice, the ORACLE WER path—least errorful path in the

[6] Arguably, more motivated users that are also more familiar with the document collection would provide a better query collection framework; the search queries are more along the lines of "exploratory search" rather than "known item search", an equally important search scenario.

[7] Discarding such queries from the test set would unfairly bias the evaluation towards the SDR results.

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

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

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.

Another interesting aspect of the PSPL lattices is the bin density—number of word entries per bin—as a function of bin position. One criticism to the PSPL representation is that a word occurrence on a given link is replicated many times in different PSPL bins since there may be paths of different lengths that arrive at the start node of the link containing a given word. This could lead to an approximately linear increase in bin density with the bin index. Fortunately this does not occur in practice: Fig. 2 plots the bin density *given that the bin at a given position exists* as well as the probability that a bin at a given position exists in some segment. The bin density increases almost linearly with the position for positions between 1 and 35 and then it stays almost constant. As it can be seen from the lower plot, the bin densities are estimated from fewer and fewer segments, so the reliability of estimates for positions beyond 60 becomes questionable.

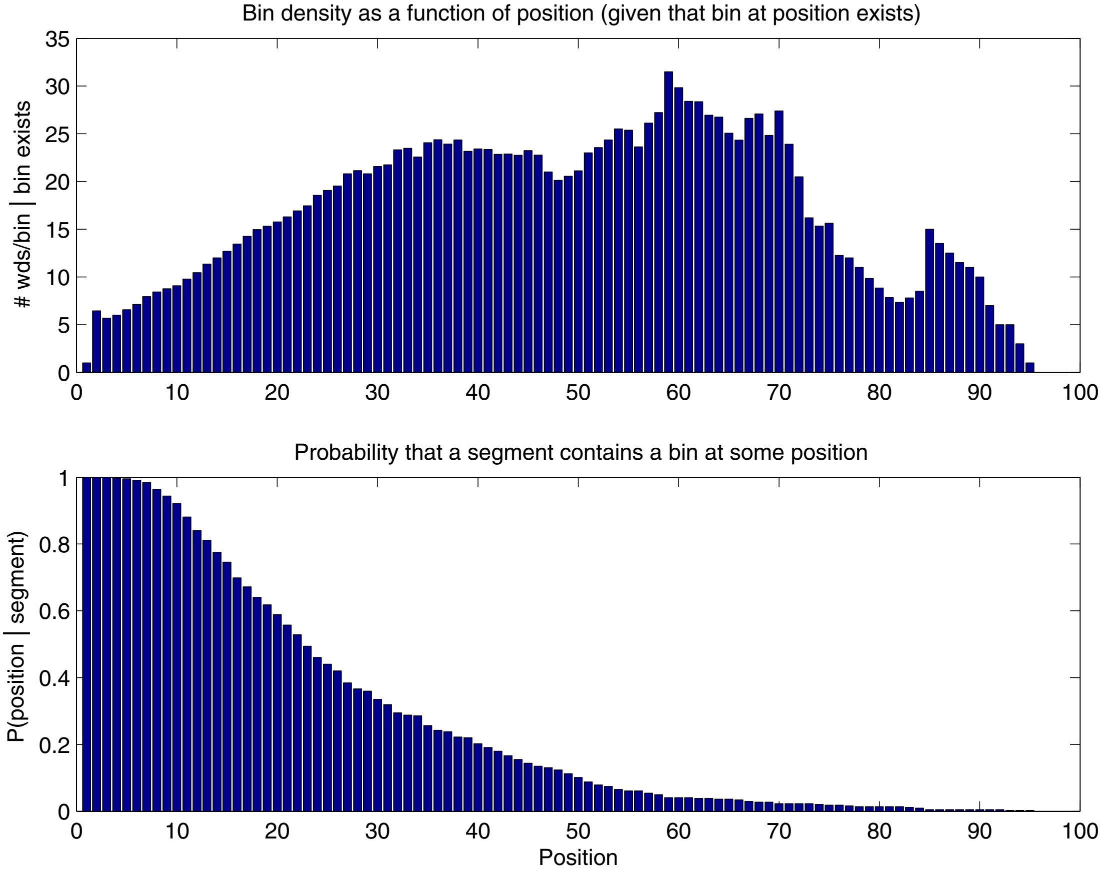

Fig. 2. PSPL bin density with position (upper plot) and probability that a bin at a given position exists in some segment (lower plot).

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

	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

The average number of bins per segment[8] is about 25.

4.3. Experiments on spoken document retrieval in the absence of text meta-data

A first batch of experiments aims at evaluating the relative merits of lattice and 1-best indexing and retrieval of speech content *in the absence of any other text meta-data*. The impact of text meta-data used in conjunction with speech content is evaluated in the experiments reported in Section 4.4.

4.3.1. MAP 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

We wish to emphasize that no tuning of retrieval weights, see Eq. (7), or link scoring weights, see Eq. (3) 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`)—17% relative improvement in mean average precision (MAP).

In order to gauge the sensitivity of the system to the accuracy of the PSPL distribution (Chelba and Acero, 2005), we have experimented with the following variations:

- `lat`: smooth PSPL, lattice link scores are flattened as explained in Eq. (3), $FLATw = 1.0$
- `raw`: lattice link scores are not flattened; $FLATw = LMw$ in Eq. (3)
- `noP`: the words in a given PSPL bin receive posterior “probability” *1.0*—resulting in a “hard-index” where more than one word occurs at the same position
- `unif`: the words in a given PSPL bin receive uniform posterior probability 1.0/#*entries*

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. (3)) has a small positive impact on the retrieval accuracy—`lat` versus `raw`.

We have also evaluated the retrieval performance when using quoted queries, such as: `‘‘OBJECT ORIENTED’’ PROGRAMMING` or `‘‘SPEECH RECOGNITION TECHNOLOGY’’`, thus performing phrase search. The quotes were assigned by hand in a plausible way, and queries containing no quotes were discarded, resulting in a set of 70 queries. Only 41 queries returned a set of relevant transcribed documents that was nonempty, and they were used as our final reference answer. Table 4 presents the results of using both PSPL and 1-best representations for the spoken document for retrieval.

[8] The bin density at position 1 is 1 since all lattices must start with SENT_START, the first word in the recognition network built by the recognizer.

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

	lat	raw	noP	unif	1-best
# docs retrieved	4971	4971	4971	4971	3206
# relevant docs	1416	1416	1416	1416	1416
# rel retrieved	1301	1301	1301	1301	1088
MAP	0.62	0.60	0.47	0.57	0.53
R-precision	0.58	0.56	0.42	0.52	0.53

Table 4
Retrieval performance on indexes built from ASR 1-best and PSPL lattices, respectively, when performing phrase search

	1-best	lat
MAP	0.58	0.73
R-precision	0.57	0.71

The relative improvement in MAP by using PSPL lattices (lat) instead of 1-best (1-best) is even higher than in the previous case, 26% relative.

The next section attempts to give an intuitive explanation why that would be the case.

4.3.2. 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 words 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[9] one could expect a dramatic increase in Recall.

Another aspect is that people enter *typical N-grams* as queries. The contents of adjacent PSPL bins are fairly random in nature, see Fig. 3, so if a typical 2-gram is found in the PSPL, chances are it was actually spoken. This translates in little degradation in Precision.

We thus believe that the PSPL (or a similar representation) could allow for a dramatic increase in Recall at little cost in Precision over the simple baseline obtained by using the 1-best ASR output, even when the ASR WER is as high as the one in this example, namely around 45%.

4.3.3. Pruning experiments

Fig. 4 presents the precision–recall graphs using the relative and absolute pruning techniques presented in Section 3.5. A wide range of threshold magnitudes were used to explore 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 Fig. 4 for both techniques. It is important to note

[9] A few factors that work against this argument are: the relative WER performance of PSPL vs. ASR 1-best is not always this good; the actual posterior probability of query terms violates the independence assumption we made, especially when considering an *N*-gram match.

```
WE SHOULD UPHOLD THE TREATIES THAT WE HAVE MADE SINCE WORLD WAR TWO
THAT ARE BEING TORN UP DAY-BY-DAY, COMPREHENSIVE TEST BAN TREATY,
NON-PROLIFERATION TREATY, ANTI-BALLISTIC MISSILE TREATY AND SO FORTH.

[30]:
BALLISTIC = -8.25249e-006
MISSILE = -11.7412
A = -15.0421
TREATY = -53.1494
ANTIBALLISTIC = -64.189
AND = -64.9143
COUNCIL = -68.6634
ON = -101.671
HIMSELF = -107.279
UNTIL = -108.239
[...]

[31]:
MISSILE = -8.25249e-006
TREATY = -11.7412
BALLISTIC = -15.0421
AND = -53.1726
COUNCIL = -56.9218
SELL = -64.9143
FOR = -68.6634
FOUR = -78.2904
SOFT = -84.1746
[...]

[32]:
TREATY = -8.25249e-006
AND = -11.7645
MISSILE = -15.0421
COUNCIL = -15.5136
ON = -48.5217
SELL = -53.1726
HIMSELF = -54.1291
UNTIL = -55.0891
FOR = -56.9218
HAS = -58.7475
[...]
```

Fig. 3. Sample utterance together with PSPL bins corresponding to ANTI-BALLISTIC MISSILE TREATY.

that the right-most point on the relative pruning curve—relative threshold equal to 0—represents the 1-best results.

Unlike the 1-best approach, both pruning techniques provide an extra degree of freedom allowing for different recall–precision operation 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. Finally, Tables 5 and 6 show the mean average precision (MAP) and R-precision as a function of different threshold magnitudes. By overlaying the MAP results on Fig. 4 one can easily see that MAP is a recall-biased metric—the highest MAP value is obtained for the least pruning, which corresponds to the highest recall point on both P/R curves.

Table 5 also shows the index size for various pruning thresholds applied to the `lat` PSPL. 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.

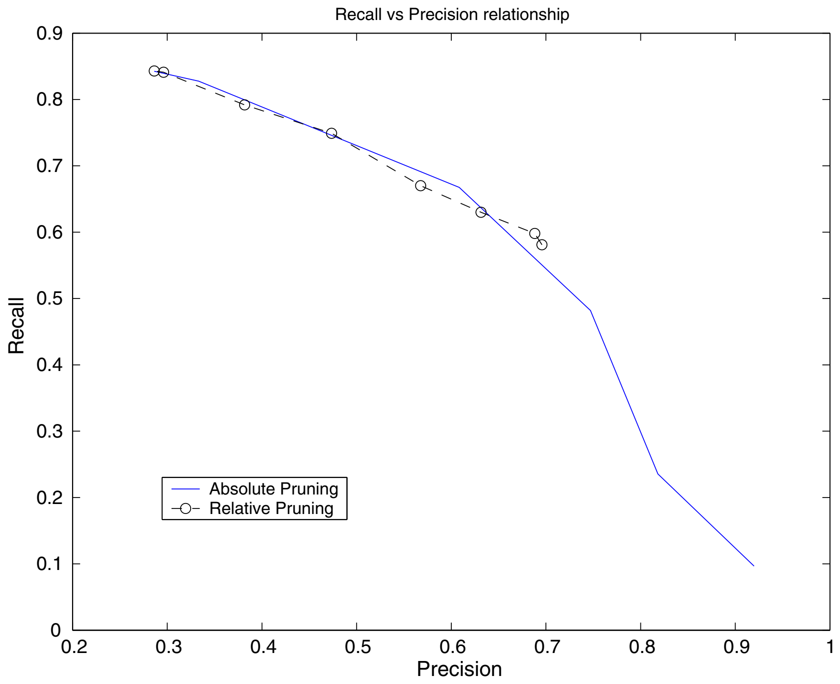

Fig. 4. Recall vs. precision for the relative and absolute threshold techniques; 1-best result is highest Precision on the relative pruning curve.

Table 5
Retrieval performance on indexes built from pruned PSPL lattices using the relative thresholding technique, along with index size; 0 threshold represents the result for the 1-best approach

τ_r Pruning threshold	MAP	R-precision	Index size (MB)
0.0 (1-best)	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	<u>0.61</u>	0.59	<u>110</u>
5.0	0.62	0.57	300
10.0	0.62	0.57	460
1000000	0.62	0.57	540

Table 6
Retrieval performance using absolute threshold pruning of hits at query run-time

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

4.3.4. WER robustness

We believe that the performance gap between the *soft-hit* (PSPL) and *hard-hit* (1-best) approaches is proportional to the uncertainty of the document content (in this case associated with the ASR performance), and they tend to be the same when the content is close to deterministic.

We artificially generated spoken document instances by swapping PSPL lattices with the transcription of the same segment. We do the same for the documents containing the 1-best transcription (by swapping the 1-best segment representation with the actual transcription of the given segment). Conceptually, we decrease the average uncertainty of the word sequence posterior distribution for documents/segments, by making some of those distributions deterministic—a delta function in a particular word sequence when the content is represented by their respective transcription; of course, in the process we decrease the "oracle" error rate (OER) and WER.

We use a "swap probability" to control the number of instances where the segment PSPL lattice is swapped with the actual transcription. In this process we generate different spoken document instances by varying the swap probability in the range $\{0.1, 0.2, 0.3, \ldots, 0.9\}$. Those instances were used to evaluate performance differences between the PSPL and 1-best retrieval approaches.

In the case of PSPL, we explore the evolution of the precision–recall curves, using the absolute pruning technique as a function of the quality of the spoken content, see Fig. 5. As expected, the area under the P–R curve increases monotonically with the value of the "swap probability".

The performance of the approach based on 1-best also improves when swapping in correct transcriptions, as can be seen in Table 7 and Fig. 5. In addition, Table 7 presents the performance gap between the PSPL (soft-hits) and the 1-best as a function of the swap probability. For the PSPL we consider the absolute pruning equal to -5.0, which is one of the best scenarios across all the spoken quality content evaluated. These results clearly show that the performance gap between PSPL and 1-best decreases as a function of the quality of the spoken document content; in all cases PSPL outperforms 1-best retrieval.

4.4. Experiments on spoken document retrieval in the presence of text meta-data

This section reports experiments whose aim is to evaluate the impact of text metadata on the retrieval performance when using the PSPL framework presented in Section 3.4.

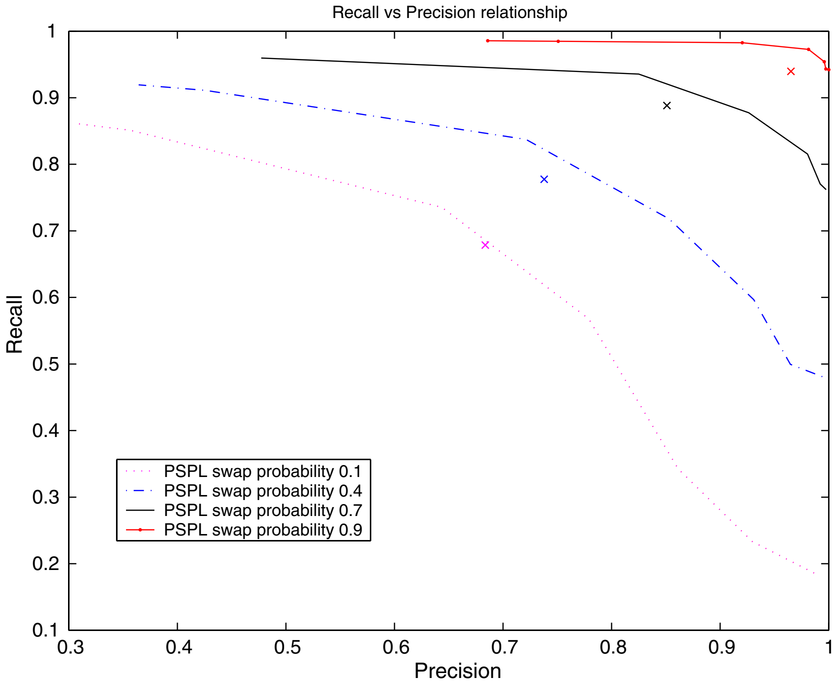

Fig. 5. Recall vs. precision curves for the absolute threshold techniques under different spoken document quality conditions; 1-best results are represented by an '×' point below their respective recall–precision curves.

Table 7
Absolute and relative MAP performance of PSPL and 1-best relevance ranking as a function of the swap probability

Swap probability	WER (%)	1-best MAP	PSPL MAP	Performances gap (%)
0.0	44.7	0.529	0.6264	18.41
0.1	40.2	0.6105	0.6601	8.12
0.2	35.8	0.6414	0.6941	8.22
0.3	31.3	0.6916	0.7456	7.81
0.4	26.8	0.7111	0.7654	7.64
0.5	22.4	0.7631	0.8049	5.48
0.6	17.9	0.8113	0.8527	5.10
0.7	13.4	0.8532	0.8747	2.52
0.8	8.94	0.9049	0.907	0.23

4.4.1. Metadata corpus

The iCampus corpus provides titles, abstracts and speaker biographies for each of the Assorted MIT World seminars documents (89.9 h). *The relative size of the metadata with respect to the spoken content is less than 1%, in number of transcribed words.* Other corpora may contain more text metadata which is also more descriptive, so our exact results may not generalize to different setups. Nevertheless, the general trend should be the same.

4.4.2. MAP experiments

In this analysis we consider two categories of segment types for every document:

- segments of type speech, PSPL lattices generated from the ASR word lattices as presented in Section 3.2;
- segments of type metadata, PSPL lattices generated directly from the text information in which we incorporate all the metadata available for the documents.

For this experimental setting we just consider the section of the corpus that has text-metadata available, namely the MIT World seminars documents (89.9 h). *Our choice is thus biased in favor of the metadata-only scenario—many documents do not contain any metadata and thus the speech content is the only hope for being able to retrieve them; we do not include those in our evaluation.*

The purpose of this set of experiments is to analyze performance changes as a function of the speech–metadata relative weight in the scoring framework, see Eq. (11); we explore different weight combinations under the following condition: $\lambda_{type\_speech} + \lambda_{type\_metadata} = 1.0$; note that this allows one to evaluate the limit cases of using only the metadata, $\lambda_{type\_metadata} = 1.0$, or only the speech content, $\lambda_{type\_metadata} = 0.0$, respectively.

Table 8 presents MAP and R-precision values for different weight combinations. As expected, in the process of increasing the relative weight of the metadata segment type with respect to using only the speech segments, there is an improvement in performance. Relative performance increases monotonically with the size of the metadata weight from 1.62% to 2.4%. This can be explained because the metadata content is much more reliable than the speech information and highly related to the content of the associated spoken document. Consequently, giving higher ranking to documents obtained from metadata than from speech improves the ranking performance. Supporting this point, Table 9 shows that precision for the metadata and speech content is 1.0 and 0.32, respectively.

Table 8
Retrieval performance as a function of the weight placed on metadata and speechcontent

Metadata weight	MAP	R-precision
0.0 (speech only)	0.6449	0.5905
0.1	0.6554	0.5999
0.3	0.6583	0.6022
0.5	0.6599	0.604
0.7	0.6606	0.6048
1.0 (metadata only)	0.1642	0.1408

Table 9
Precision and recall for metadata, speech and speech–metadata scenarios, this last one with $\lambda_{type\_metadata}$ equal to 0.8

Scenario	Precision	Recall
Metadata	1	0.056
Speech	0.319	0.815
Meta-speech	0.323	0.826

Table 10
Relative MAP performance gain of using speech and metadata for document indexing under different metadata representation quality conditions

Sampl. Prob.	Meta (MAP)	Meta-speech (MAP)	Relative gain (%)
0.01	0.106	0.647	510.1
0.04	0.131	0.647	394.6
0.08	0.182	0.665	265.2
0.10	0.206	0.670	225.0

When placing all the weight on metadata segments there is a significant drop in MAP performance. Looking at it the other way, the performance gain obtained by adding the speech content with respect to only considering the metadata is 302% relative. Consequently, adding spoken document information provides a dramatic gain in performance, which should be no surprise given the fact that the metadata constitutes only about 1% of the amount of words in the transcription of speech content.

4.4.3. Enriching meta-data

In order to explore the interaction between metadata and speech content and its impact on the MAP performance gain, we enrich the original text metadata by adding the actual transcription of the spoken documents at different sampling rates.

In this case, we consider the speech segment transcription as the atomic sampling unit. More precisely, for every document *D* we enrich its metadata content by adding metadata segments which correspond to the transcription of some of the speech segments associated with *D*. We thus generate 4 different metadata sets:

- metadata + 1% transcription
- metadata + 4% transcription
- metadata + 8% transcription
- metadata + 10% transcription

Given that we use the actual transcription as a source for the enriched metadata, the entire iCampus corpus can be used for evaluation since we now have metadata for every spoken document in the corpus; we consider the same relative weight between text metadata and speech across all these experiments: $\lambda_{type\_metadata} = 0.8$ and $\lambda_{type\_speech} = 0.2$.

Table 10 presents the relative improvement in MAP for different metadata conditions. It can be easily seen that in all metadata scenarios there is a significant gain in performance by adding the spoken content. In particular, even when the metadata segments account for more than 10% of the transcriptions, the spoken content still provides a relative improvement of more than 200% in MAP. Surprisingly, these results are obtained using an ASR system with high WER (44.7%, Table 1).

5. Conclusions, discussion and future work

The PSPL framework provides an alternative way of dealing with the document content uncertainty for spoken document 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 features in the relevance score calculation.

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 relevance ranking score. This property can be used to adjust precision–recall performance metrics to the particular optimality criterion of a given application or user needs.

The PSPL framework provides better retrieval performance than the 1-best in scenarios with relative high WER. In such cases the 1-best document representation has in fact high variance and by taking into account the confidence of the ASR system into its output, the PSPL better represents the document content. As WER decreases and ASR confidence in 1-best output increases, the PSPL representation converges gracefully towards the 1-best one.

Regarding the incorporation of text metadata, the PSPL approach provides the flexibility to represent both deterministic and stochastic document content. Leveraging this property, we propose a new relevance ranking framework that takes into account the different nature of the information sources available in the retrieval problem—deterministic and stochastic.

Moreover, experimental evidence supports the idea that exploiting the content of the spoken document does indeed provide a significant improvement in performance with respect to a scenario in which only the metadata are used for retrieval.

As for future work items, we would like to develop a scoring framework that uses the ranking on the reference side as well, and not just a binary relevance reference judgment—see Section 4.1.4. Once that is accomplished, the natural next step is to use a reference-ranking algorithm that uses proximity and context features, and not just the current TF-IDF based one.

We also believe that more desirable ways of integrating different types of text metadata and speech content must exist and our naive approach at combining these content streams could be significantly improved; in particular, the interaction between the two pruning techniques outlined in Section 3.5 and the use of text-metadata for spoken document retrieval deserves careful experimentation.

Tackling the OOV problem in an appropriate way is also a must if one aims at deploying such a search engine in the real world.

Acknowledgments

We thank Jim Glass and T.J. Hazen at MIT for providing the iCampus data.

References

Baeza-Yates, R., Ribeiro-Neto, B., 1999. Modern Information Retrieval. Addison Wesley, New York, pp. 27–30 (Chapter 2).

Brin, S., Page, L., 1998. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems 30 (1–7), 107–117. Available from: <www.citeseer.ist.psu.edu/brin98anatomy.html>.

Brown, M.G., Foote, J.T., Jones, G.J.F., Jones, K.S., Young, S.J., 1996. Open-vocabulary speech indexing for voice and video mail retrieval. In: Proceedings of the ACM Multimedia 96, Boston, 1996, pp. 307–316.

Chelba, C., Acero, A., 2005. Position specific posterior lattices for indexing speech. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 443–450. Available from: <http://www.aclweb.org/anthology/P/P05/P05-1055>.

Chelba, C., Acero, A., 2005. Indexing uncertainty for spoken document search. In: Proceedings of Eurospeech, ISCA, Lisbon, Portugal, pp. 61–64. Available from: <http://research.microsoft.com/srg/papers/2005-chelba-eurospeech.pdf>.

Church, K.W., 2003. Speech and language processing: where have we been and where are we going? In: Proceedings of Eurospeech, Geneva, Switzerland.

Garofolo, J., Auzanne, G., Voorhees, E., 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, 2000. Available from: <www.citeseer.ist.psu.edu/garofolo00trec.html>.

Glass, J., Hazen, T.J., Hetherington, L., Wang, C., 2004. Analysis and processing of lecture audio data: preliminary investigations. In: HLT-NAACL 2004 Workshop: Interdisciplinary Approaches to Speech Indexing and Retrieval, Boston, Massachusetts, pp. 9–12.

James, D.A., 1995. The application of classical information retrieval techniques to spoken documents, Ph.D. Thesis, University of Cambridge, Downing College.

Logan, B., Moreno, P., Deshmukh, O., 2002. Word and sub-word indexing approaches for reducing the effects of OOV queries on spoken audio. In: Proceedings of the HLT, 2002. Available from: <www.citeseer.nj.nec.com/585562.html>.

Mangu, L., Brill, E., Stolcke, A., 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. Computer Speech and Language 14, 373. Available from: <http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:cs/0010012>.

Ng, K., 2000. Subword-based approaches for spoken document retrieval. Ph.D. Thesis, Massachusetts Institute of Technology.

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

Sanchez, J.S., Chelba, C., Acero, A., 2006. Pruning analysis of the position specific posterior lattices for spoken document search. In: ICASSP'06 Proceedings, IEEE, Toulouse, France, pp. 945–948.

Saraclar, M., Sproat, R., 2004. Lattice-based search for spoken utterance retrieval. In: HLT-NAACL 2004, Boston, Massachusetts, pp. 129–136.

Seide, F., Yu, P., 2004. A hybrid word/phoneme-based approach for improved vocabulary-independent search in spontaneous speech. In: Proceedings of ICSLP, Jeju, Korea.

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

Siegler, M.A., 1999. Integration of continuous speech recognition and information retrieval for mutually optimal performance, Ph.D. Thesis, Carnegie Mellon University.

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

Young, S., Evermann, G., Hain, T., Kershaw, D., Moore, G., Odell, J., Dave Ollason, D.P., Valtchev, V., Woodland, P., 2002. The HTK Book. Cambridge University Engineering Department, Cambridge, England. Available from: <http://htk.eng.cam.ac.uk/docs/docs.shtml>.

Zhou, Z.-Y., Yu, P., Chelba, C., Seide, F., 2006. Towards spoken-document retrieval for the internet: lattice indexing for large-scale web-search architectures. In: Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, Association for Computational Linguistics, New York City, USA, pp. 415–422. Available from: <http://www.aclweb.org/anthology/N/N06/N06-1053>.