Robust Indexing and Retrieval of Spoken Documents

Presenter: Suhan Yu

Reference

- Soft indexing of speech content for search in spoken documents. Ciprian Chelba et al. (2007)
- Indexing Uncertainty for Spoken Document Search. *Ciprian Chelba and Alex Acero.*(2005)
- Integration of metadata in spoken document search using position specific posterior lattice. *Jorge Silva*1*, Ciprian Chelba and Alex Acero*. (2006)
- The TREC spoken document retrieval track: A success story. John S. Garofolo, Cedric G. P. Auzanne, Ellen M. Voorhees. (2000)
- Lattice-Based Search for Spoken Utterance Retrieval. Murat Saraclar and Richard Sproat. (2004)
- Beyond ASR 1-best Using word confusion networks. Dilek Hakkani-Tu¨ r et al. (2005)

Spoken Document Retrieval

- SDR is accomplished by using a combination of automatic speech recognition and information retrieval technologies.
- A speech recognizer is applied to an audio stream and generates a time-marked transcription of the speech.
- The transcription may be phone- or word-based in either a lattice, n-best list, or more typically,
- Narrow the gap between speech and text document retrieval.

Spoken Document Retrieval

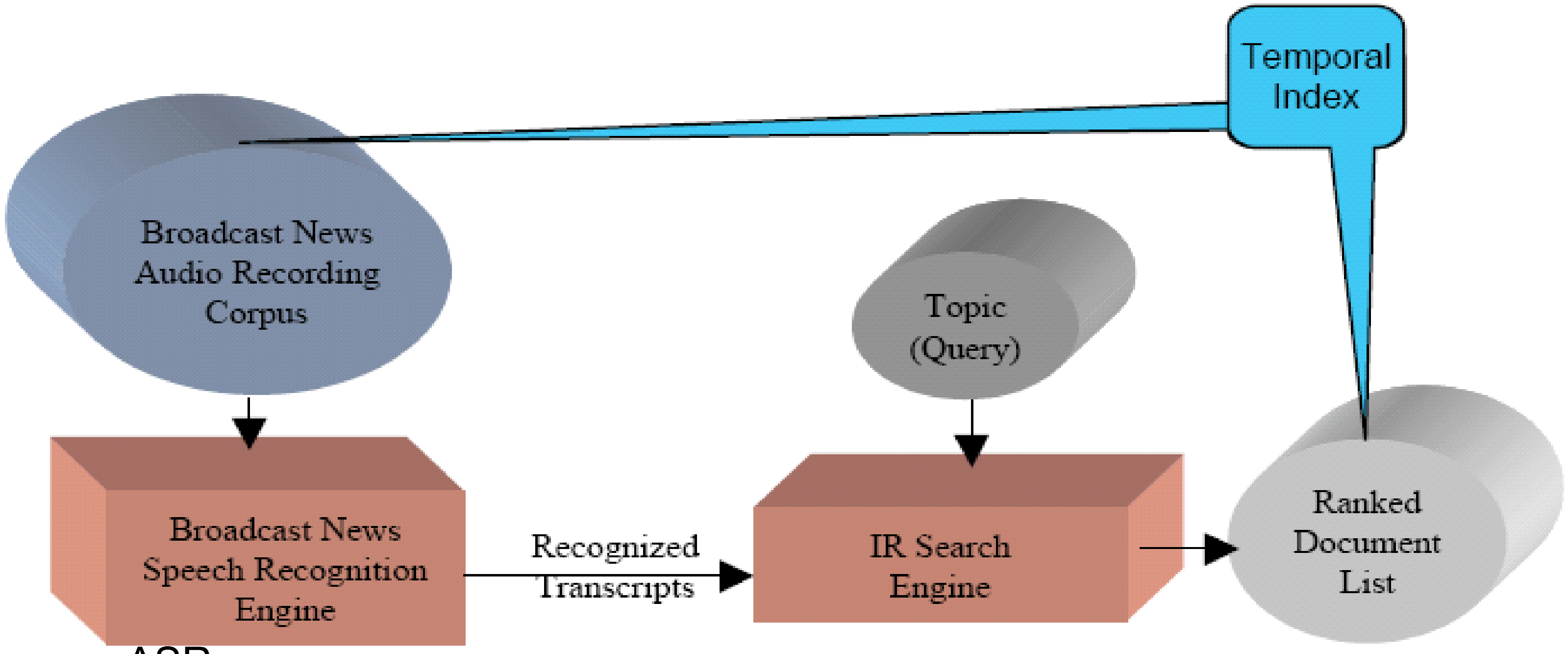
- The goal is to enable users to:
  - Search for spoken documents as easily as they search for text.
  - Accurately retrieve relevant spoken documents.
  - Efficiently browse through returned hits.
  - Quickly find segments of spoken documents they would most like to listen to or watch.

HIT = an occurrence of a query word in a document

Spoken language understanding

- Two key components in goal driven human–machine conversational systems
  - Utterance intent determination (call classification)
  - Corresponding argument extraction (named entity extraction)
    - For example, if the user says ‘‘I would like to get my balance for the account number 1 2 3 4 5 6’’, then the corresponding intent or semantic label (call-type) would be ‘‘Request (Balance)’’ and an argument or parameter for this call-type, i.e., the account number, would be ‘‘123456.’’

Typical SDR Process



ASR
(Automatic speech recognizer)

Figure 1: Typical SDR Process

ASR 1-best
(2002,2003)

ASR
word lattices
N-best
(1997)

ASR
word
confusion
network
(2005)

Typical SDR Process

- The system consists of three main components.
  - First, the ASR component is used to convert speech into a lattice representation, together with timing information.
  - Second, this representation is indexed for efficient retrieval.
  - Finally, when the user enters a query the index is searched and matching audio segments are returned.

Word confusion network

- WCNs are much smaller than ASR lattices and they still have comparable word accuracy using their best path and even better oracle accuracy.

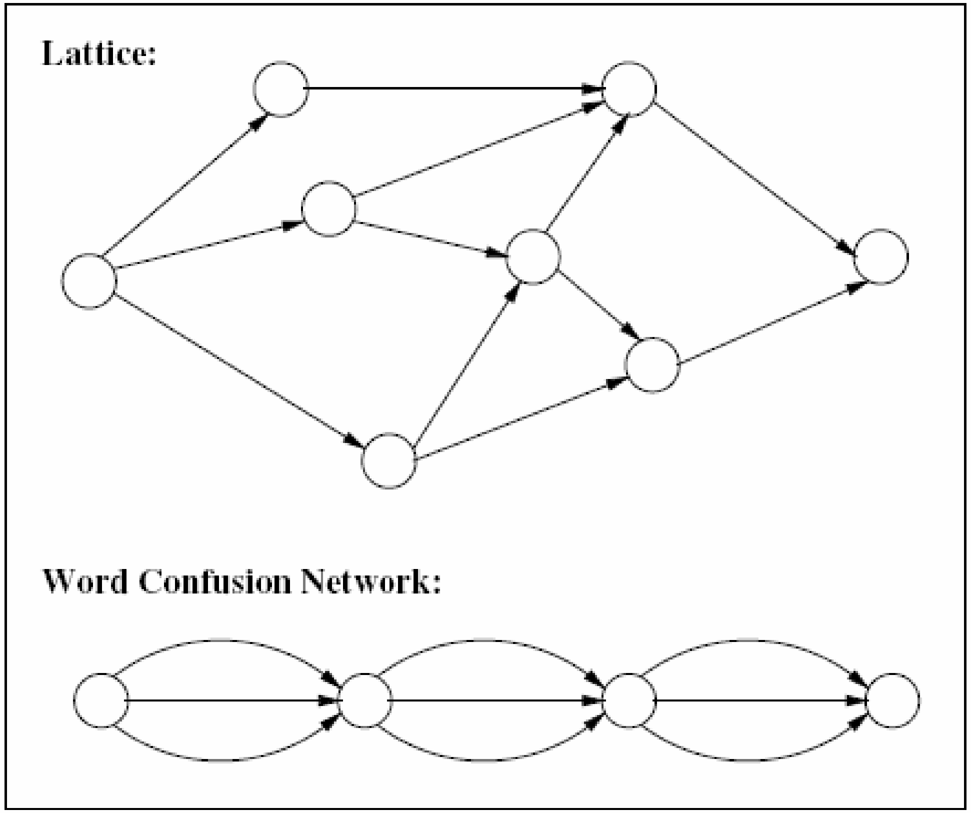


Fig. 1. Typical structures of lattices and WCNs.

Lattice Indexing

- In the case of lattices, we store a set of indices, one for each arc label (word or phone) l, that records the lattice number L[a], input-state k[a] of each arc a labeled with l in each lattice, along with the probability mass f(k[a]) leading to that state, the probability of the arc itself p(a|k[a]) and an index for the next state.

$$p(a) = \sum_{\pi \in L : a \in \pi} p(\pi) = f(k[a])p(a|k[a])$$

TREC SDR Background

- In 1996, an evaluation of retrieval using the output of an optical character recognizer (OCR) was run as a “confusion” track in TREC-5 to explore the effect of OCR errors on retrieval.
- This track showed that it was possible to implement and evaluate retrieval on “corrupted” text.
- After implementing this track, NIST and members of the TREC community thought it would be interesting to implement a similar experiment using automatic speech recognition (ASR).

		TREC-4 (1995)	TREC-5 (1996)	TREC-6 (1997)	TREC-7 (1998)	TREC-8 (1999)	TREC-9 (2000)
confusion	confusion	🟨	🟨				
	Spoken Document Retrieval			🟨	🟨	🟨	🟨

TREC SDR Background

- Led by Karen Spärck Jones from the University of Cambridge.
- Discuss the possibility of applying information retrieval techniques to the output of speech recognizers.

TREC-6 SDR: Known Item Retrieval

- The first year for the SDR Track was truly one of getting the speech and IR communities together and exploring the feasibility of implementing and evaluating SDR technology.
- The goal in a known-item retrieval task is to generate a single correct document for each topic rather than a set of relevant topics as in an ad hoc task.
- Differences between broadcast news stories and document-based IR collections
  - The broadcast news stories (276 words per story) were extremely short with regard to number of words.

TREC-6 SDR: Known Item Retrieval

- The first SDR evaluation showed us that we could successfully implement an evaluation of SDR technology and that existing component technologies worked well on a known-item task with a small audio collection.

Reference retrieval using "perfect"[1] human-transcribed reference transcriptions
Baseline retrieval using "given" IBM ASR-generated transcriptions
Speech retrieval using the recordings themselves, requiring both ASR and IR components

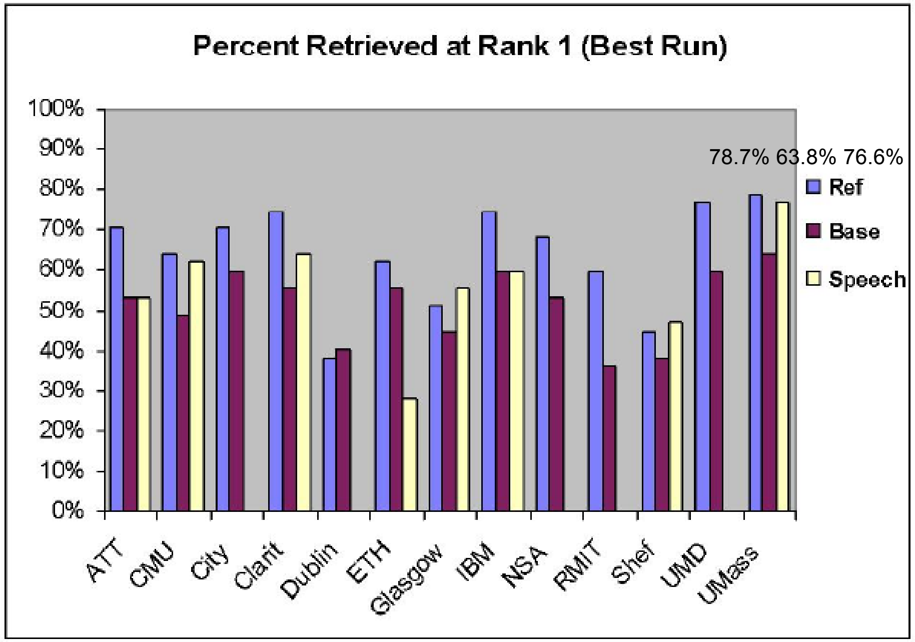


Figure 2: TREC-6 SDR Retrieval rate at rank 1 for all systems and modes (best run)

TREC-7 SDR : Ad Hoc Retrieval

- In an ad hoc retrieval test, systems are posed with topics and attempt to return a list of documents ranked by decreasing similarity to the topic.
- Following are two of the test topics they created
  - *Find reports of fatal air crashes.* (Topic 62)
  - *What economic developments have occurred in Hong Kong since its incorporation in the Chinese People’s Republic?* (Topic 63)

TREC-8 SDR : Large Audio Collection

- Linguistic Data Consortium began collecting a large radio and television corpus for the Topic Detection and Tracking (TDT) program.
- The TDT-2 corpus, collected to support the TDT program in 1998-99, contains news recordings from ABC, CNN, Public Radio International, and the Voice of America.
- Best ASR results were obtained by the University of Cambridge HTK recognizer with a 20.5% WER.
  - As with the speech recognition performance, overall retrieval performance was quite good.
  - Example:
    - Topic 105: *How and where is nuclear waste stored in New Mexico?*

    (.85 average MAP across all systems/runs, 7 relevant stories).

TREC-9 SDR Plans

- Few minor changes.
- the story boundaries unknown condition can make effective use of audio-signal information not found in the transcriptions such as speaker changes, noise changes, volume changes, music, prosody, etc., we will encourage the development of a common non-lexical information exchange format which can be used to store and share such information.

Soft indexing using position specific posterior probability lattices

- Text indexing and search
  - Tf-IDF Model:
    - $S(D_j, \vartheta) = \sum_{i=1}^{Q} w_{i,j}, \quad w_{i,j} = f_{i,j} \cdot idf_i$
    - Drawback:
      - The query terms are assumed to be independent. Proximity information is not taken into account at all.
      - Query terms may be encountered in different contexts in a given document.
- Page Rank
  - Early Google approach (Brin and Page, 1998)
  - For each given query term qi one retrieves the list of hits corresponding to qi 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.

Soft indexing using position specific posterior probability lattices

- Position specific posterior probability lattices
  - Position information is crucial for being able to evaluate proximity when assigning a relevance score to a given document.
  - 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.
  - ASR lattices do have much better WER but the position information is not readily available.

Position specific posterior probability lattices

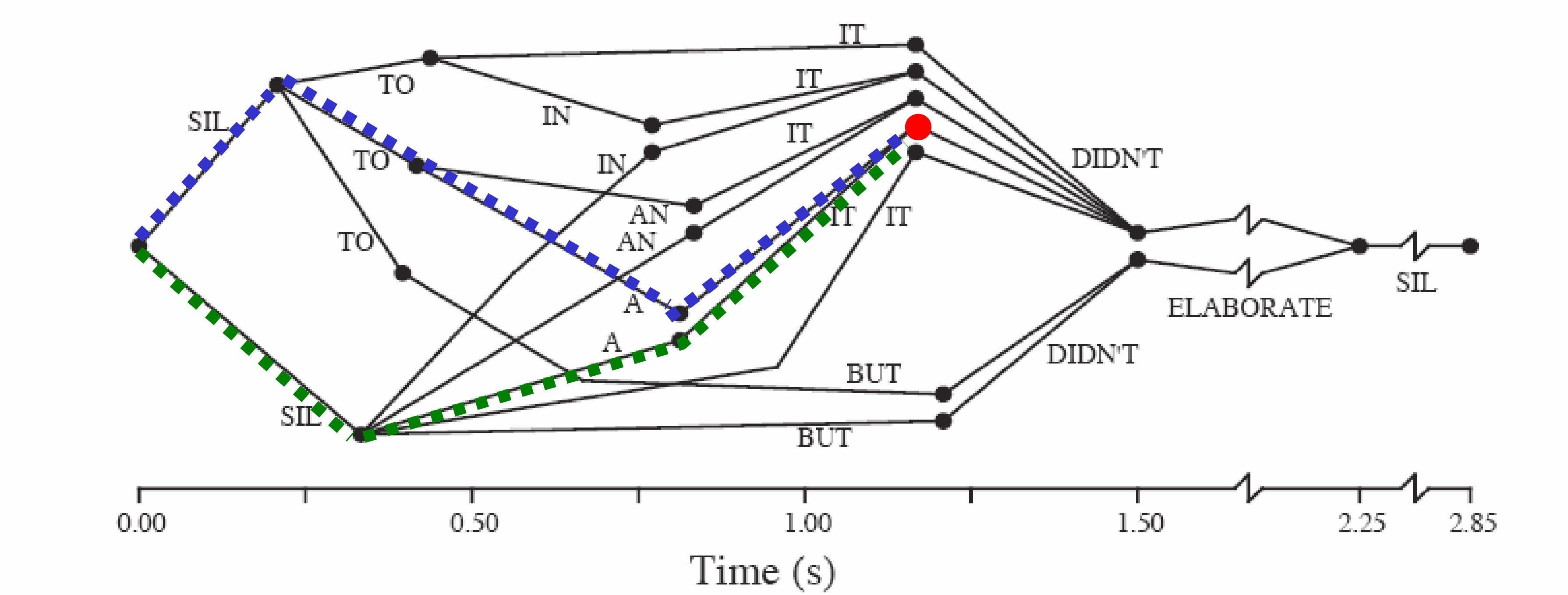
- For each word in the lattice
  - Index soft-hits

(document id, position, posterior probabilit y)

- Split forward probability based on path length

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

The different from pre-work



Position specific posterior probability lattices

- The backward probability β_n has the standard definition

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

$$w = word(n) \quad \delta = 1$$

$$w \neq word(n) \quad \delta = 0$$

Relevance ranking using PSPL representation

- This paper ignore the OOV problem
- There divided several segment in a document
- 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.

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

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

N-gram

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

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

Relative pruning

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

$$w_k^* = \arg\max_{w \in v} p(w_k(s) = w \mid D)$$

Word set $$W_k = \{w \in V : \log \frac{P(w_k(s) = w_k^* \mid D)}{P(w_k(s) = w \mid D)} \leq \tau_r\}$$

- 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.

Experiments

- iCampus corpus
  - 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)
- Generate lattice
  - 3-gram ASR lattices
  - PSPL lattices

Experiments

- Query collection and retrieval setup
  - Query out-of-vocabulary rate (Q-OOV) was 5.2%
  - The average query length was 1.97 words.
  - Removed the queries which contained OOV words
- Evaluation Metrics
  - trec_eval (NIST) package requires reference annotations for documents with binary relevance judgments for each query
  - Standard Precision/Recall and Precision@N documents
  - Mean Average Precision (MAP)
  - R-precision (R=number of relevant documents for the query)

Experiments Result

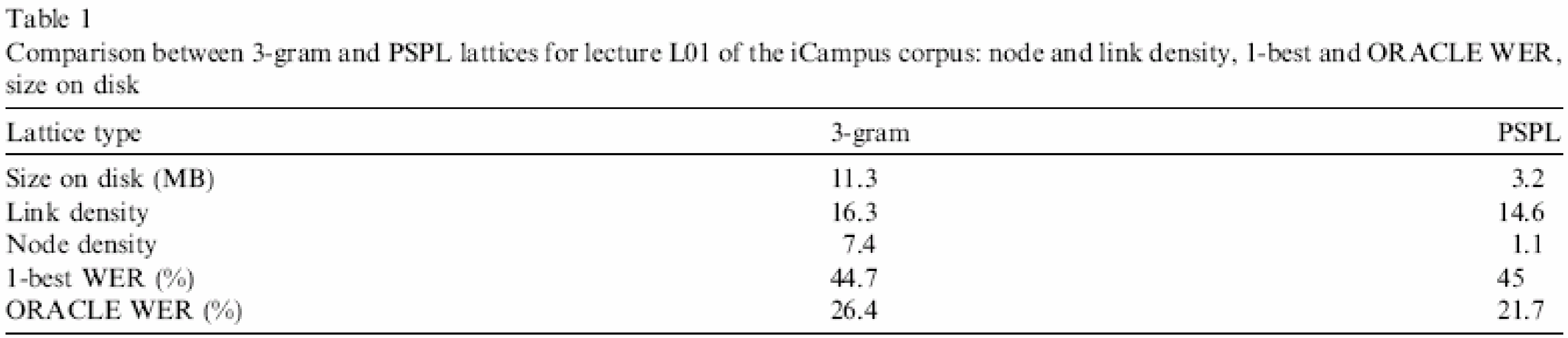


Table 1
Comparison between 3-gram and PSPL lattices for lecture L01 of the iCampus corpus: node and link density, 1-best and ORACLE WER, size on disk

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

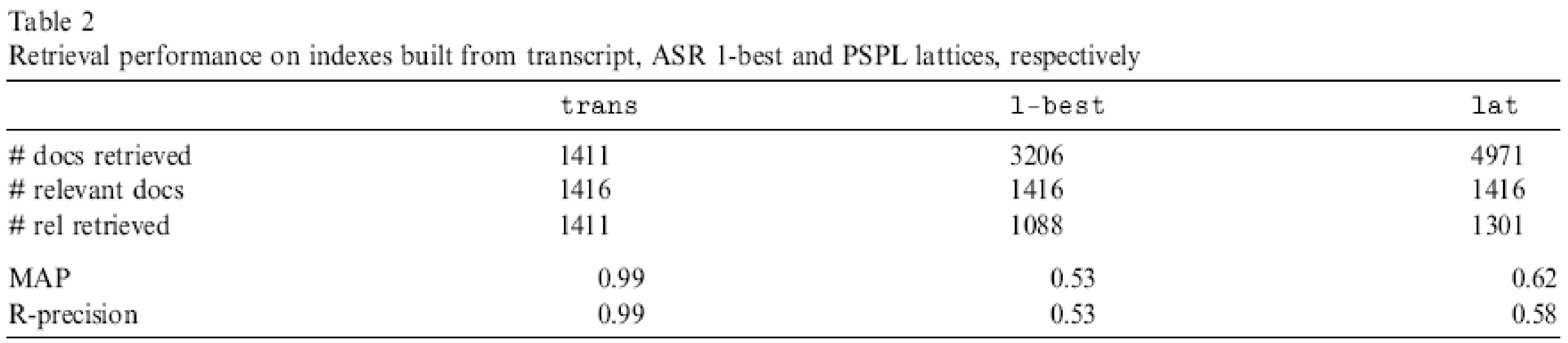
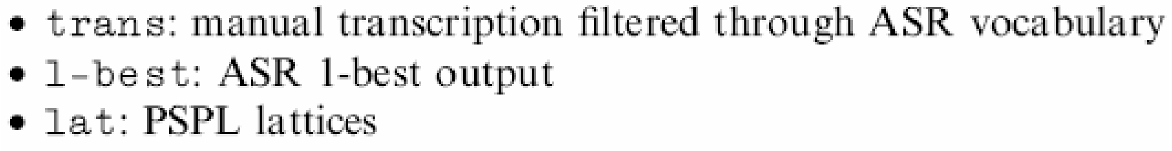


Table 2
Retrieval performance on indexes built from transcript, ASR 1-best and PSPL lattices, respectively

	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



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

Experiments Result

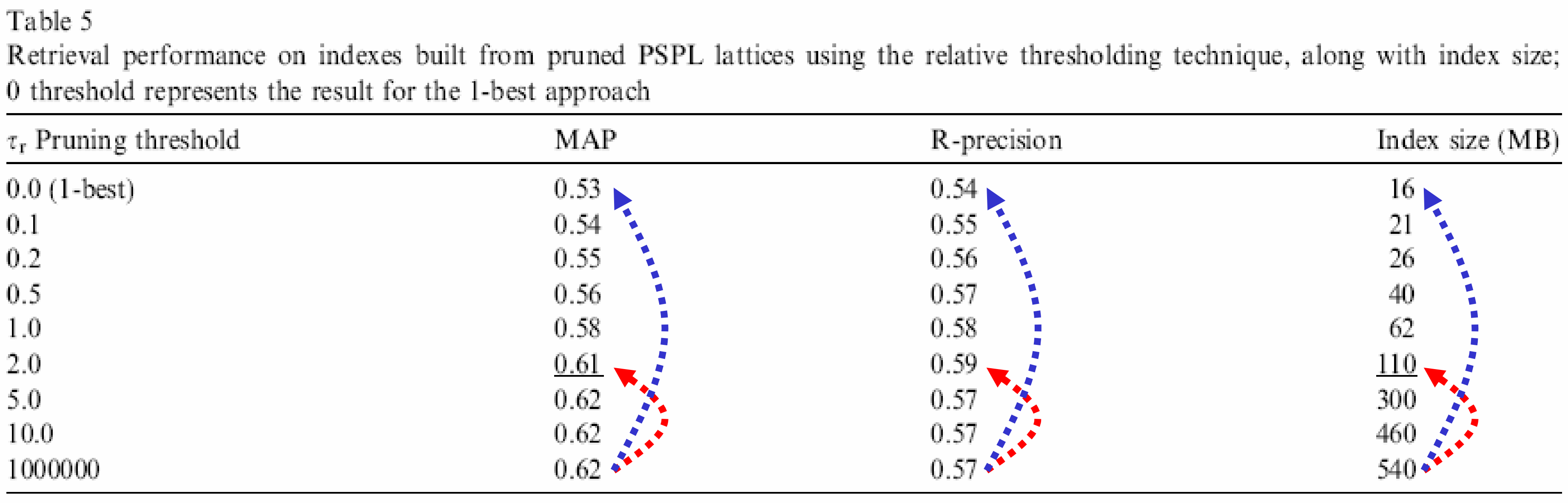


Table 5
Retrieval performance on indexes built from pruned PSPL lattices using the relative thresholding technique, along with index size; 0 threshold represents the result for the 1-best approach

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	0.61	0.59	110
5.0	0.62	0.57	300
10.0	0.62	0.57	460
1000000	0.62	0.57	540

Conclusion

- The PSPL framework provides better retrieval performance than the 1-best in scenarios with relative high WER.
- 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.
- 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.