Fusion of Semantic and Acoustic Approaches for Spoken Document Retrieval

Beth Logan, Patrawadee Prasangsit, Pedro Moreno
Cambridge Research Laboratory
HP Laboratories Cambridge
HPL-2003-55
March 18th , 2003*

spoken document retrieval, out-of-vocabulary words, multimedia indexing, knowledge management

Most spoken document retrieval systems use the words derived from a large vocabulary speech recognizer as the internal representation for indexing the document. However, the use of recognition transcripts inherently limits the performance of the system since the size of the dictionary restricts the number of queries for which matches can be found. In this paper we present a new approach to this problem based on combining Probabilistic Latent Semantic Analysis (PLSA) with phonetic indexing. PLSA maps the words in documents and queries into a semantic space in which they can be compared even if they don't share any common words. Combining this semantic distance with acoustic scores gives an improvement of 6-11% relative for OOV queries and 4% relative for all queries on a 75 hour broadcast news indexing task.

* Internal Accession Date Only Approved for External Publication
To be presented at the ISCA Workshop on Multilingual Spoken Document Retrieval, 4-5 April 2003, Hong Kong

1 Introduction

As more spoken audio becomes available on the Web, the need for efficient ways to index this data arises. Since transcribing all this audio by hand is infeasible, the most common way to build a retrieval system is to transcribe the data using speech recognition and then apply standard algorithms for indexing the corresponding textual documents. This index can then be used to provide pointers to the relevant audio in response to a user's query. *SpeechBot* [14] (http://www.speechbot.com) is an example of such a system. It uses speech recognition technology to allow users to search archived radio broadcasts, returning a list of audio clips related to user's textual query.

Although such schemes can be effective, many challenges remain. First, the transcriptions from speech recognition are rarely perfect. Error rates can vary from 15% for clean speech to as much as 50% for extremely noisy speech or for spontaneous speech. Inserted and substituted words in the transcripts can lead to false hits for a query. If query words are deleted from the transcripts it will be impossible to find those sections of audio using the query.

Second, the dictionary and language model used by the speech recognizer are often limited to around 65,000 words. This is insufficient to cover typical spoken documents and queries. For example, our studies show the number of unique words in *The New York Times* over a two year period to be around 650,000 words. Another study shows that up to 13% of user queries may be Out-of-Vocabulary (OOV) [9]. Thus OOV words are a problem, particularly in domains where new words appear over time such as broadcast news.

In general, solutions to the OOV problem can be classified into acoustic and semantic approaches. Acoustic approaches recover portions of audio that are *acoustically similar* to the query word/s. Examples include using indexes of phonemes, sequences of phonemes or syllables, or expanding the OOV word query into acoustically similar in-vocabulary phrases. (e.g. [7], [13], [15], [10], [11]). Although such systems are useful, they may suffer from lack of scalability and all have high false positives rates for any given query.

Other approaches to the OOV problem, typically motivated by work in the text Information Retrieval (IR) community, use *semantic* information about the query to find related documents. An example is query and/or document expansion (e.g. [16]) which uses additional documents to find related in-vocabulary query words. Its performance relies heavily on the quality of these additional documents. A related technique is to change the recognizer vocabulary using documents from a parallel corpus (e.g. [8]). Other semantically motivated approaches have been investigated for text indexing (e.g. [4], [5]). They are based however on the vector-space document model which does not scale well to very large collections.

Clearly both acoustic and semantic solutions offer advantages and disadvantages. In this paper, we investigate a system which combines both techniques. Our motivation is our observation in previous work with acoustic approaches that many of the incorrect hits are semantically unrelated to the original query. Thus the use of semantic information might aid in the elimination of such hits.

Our approach uses acoustic information to quickly obtain a set of initial hits for the query ranked by a *tf.idf* metric. We then use Probabilistic Latent Semantic Analysis (PLSA) [5] to obtain a score reflecting how semantically close each hit is to the query. A linear combination of both acoustic and semantic scores yields the final ranking of the hits.

Our approach is novel for several reasons. First, we examine the use of PLSA for speech indexing as opposed to text indexing. Second, we apply it in a way that is more scalable than the original formulation. Third, we combine PLSA with acoustic information.

2 Probabilistic Latent Semantic Analysis

The PLSA model is a family of probability distributions over a pair of discrete random variables. For text data, this pair consists of a document label and a word. Let d denote a document from a corpus, w a word, and z a topic. Assuming the occurrence of a document and a word are independent of each other given a topic, the joint probability of generating a particular topic, word, and document label is

$$P(d, w, z) = P(d|z)P(w|z)P(z).$$

Here, $P(w|z)$ is a unigram language model conditioned on the topic, $P(d|z)$ is a probability distribution over the training document labels and $P(z)$ is the prior distribution of the topics. The $P(d, w)$ document term matrix is easily computed as

$$P(d, w) = \sum_{i=1}^{Z} P(d|z_i)P(w|z_i)P(z_i).$$

Effectively PLSA factors a highly sparse document-term matrix into three smaller dimensional matrices by introducing a new hidden topic variable. These variables are commonly associated with natural clusters that in the case of news documents represent topics such as *politics* or *sport*.

Given a corpus of N documents and the words within those documents, the training data for a PLSA model is the set of pairs $\{(d_n, w_n^d)\}$ for each document label and each word in those documents. We can use the Expectation Maximization (EM) algorithm to learn such a model from an uncategorized corpus. Further details on the training technique are given in [5]. For our experiments we have used two years of the *New York Times* to train a PLSA model of generic news corpora. We prefer to use a training corpora which is well segmented and free from speech recognition errors as this will generally yield cleaner PLSA models.

Once this PLSA model is trained on clean textual data we apply the PLSA decomposition to our spoken document corpora. For each document in the index and each query we can compute $P(z|d)$ which constitutes the low dimensional representation or ‘signature’ (in the topic space) of the document or query. Each dimension of the reduced space reflects the likelihood of that document being about a pre-learnt topic z_i. Similarity between documents or between documents and queries is evaluated in the concept space and reflects likely semantic closeness rather than merely ‘word overlap’ as represented by the bag of words model.

3 Scalable Indexing

PLSA has been demonstrated to give good performance on text indexing tasks [6]. However, as originally published, retrieval using PLSA is an order N operation since the signature for

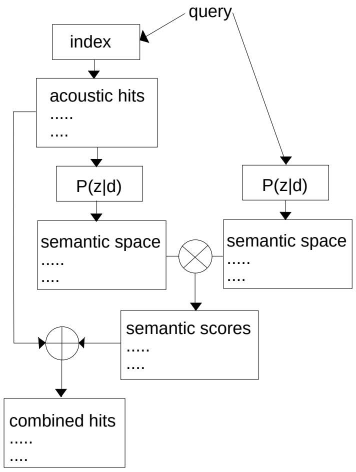

Figure 1: Our indexing scheme which combines acoustic and semantic information

the query must be compared to the signature for each document in the archive. Additionally, for Web-based applications which typically have short queries, the signature can be poorly characterized. Recognition errors can cause additional problems.

We therefore consider using PLSA merely to re-rank a set of acoustic matches returned from our indexing system rather than ranking all documents in the archive. We use the highly scalable Alta Vista index to return the initial acoustic matches.

To cope with poorly characterized signatures, we further use a combination of acoustic and semantic information to compute the final ranking. In this paper we simply use linear combination of the scores but more complex fusion schemes are possible. Our retrieval process is shown in Figure 1.

4 Experiments

We experiment on a broadcast news audio database. The details are given below.

4.1 Audio Database

We index a set of broadcast news audio for which we have ground truth transcriptions supplied by the Linguistic Data Consortium. The audio is from broadcast sources and is sampled at 16kHz. For training acoustic models we use 65 transcribed hours of the HUB4_96 training set. Our indexing experiments are performed on about 75 hours of audio composed of the HUB4_96 development and test data and the HUB4_97 training and test data.

4.2 Topic Training Data

Our *New York Times* corpora consists of 95,000 text articles from 1998, 1999 and the first 5 months of 2000. We use this corpora to train varying numbers of topic models. As discussed in Section 2, we prefer to use clean, well segmented data to train the models. The topic model dictionary is around 250,000 words with stop words removed. We use as large a dictionary as possible in order to obtain meaningful values of $P(z|\mathrm{query})$ for a wide range of query words.

4.3 Document and Relevance Definitions

We study two different definitions of *document*. Our *SpeechBot* system indexes audio documents which are at least half an hour long. Our current user interface plays 10s audio *clips* in response to user queries. We therefore examine one definition of a document as a 10s clip.

We have found however that users prefer to hear the audio from the start of the current story which gives more context to the occurrence of the query word/s. Therefore, we also examine retrieval using documents created using our in-house topic segmenter [1].

We use unambiguous proper name queries in order that we can score relevance automatically. A document is defined as relevant if it contains the query or its plural or possessive form. Thus documents containing *clinton, clinton's* or *clintons* would be relevant for the query *clinton*.

4.4 Evaluation Metric

Our evaluation metric is 11-pt average precision. This is an estimate of the area under a precision *vs* recall curve and is an overall measure of the quality of a retrieval system. The greater this area, the better the system. An ideal system has average precision 1.0. We average our results over all queries and thus report mean average precision.

4.5 Query Selection

In [2] it is recommended that at least 25 and preferably 50 queries are used for an evaluation for which average precision is the metric. We therefore use 50 queries. Our aims in query selection are:

- to use proper name queries for which relevance can be determined automatically;
- to have a high proportion of OOV queries;
- to use 'real-world' queries;
- to have at least 10 results for each query similar to a Web page of hits.

Finally, all queries, including OOV queries, should be in-vocabulary with respect to the dictionary used for the topic models. Otherwise, as discussed in Section 4.2 we are unable to determine meaningful values of $P(z|\mathrm{query})$. When training the topic models, we therefore use a dictionary about 3 times larger than the speech recognizer's dictionary to cover as many queries as possible. Comparison of the ground truth to the word recognition and topic dictionaries yields 56 one to four word phrases that have at least 10 hits, contain proper names and are

OOV with respect to the recognizer dictionary but are in-vocabulary with respect to the topic models' dictionary. Eliminating duplicates and spelling errors yields 15 suitable OOV queries.

For in-vocabulary queries, we searched the *SpeechBot* logs for the most frequent proper name queries which had at least 10 hits in the ground truth and which were in-vocabulary. *SpeechBot* has been live on the Web since December 1999 and is therefore a good source of real-world queries.

The resulting query set therefore has an OOV rate of 30%. 62% of the queries are single words, 34% two words and 4% three words. Note that our query OOV rate is much higher than the 13% rate observed on the site[9]. We show our query set in Table 4 (at the end of the paper).

As an aside, we note that such a query set is relatively unaffected by stemming.

4.6 Indexes

We build a word index and a phoneme index where the phonemes for the phoneme index are derived from the first-best word transcription of the audio. This transcription is obtained using our in-house large vocabulary speech recognizer. This is a standard speech recognition system based on hidden Markov model (HMM) technology. We model 6,000 tied states using Gaussian mixture models. We use a standard trigram language model with a vocabulary of 65,000 words. The acoustic and language models are trained on the 65 hour HUB4_96 training set (disjoint from the indexed audio). Some additional text sources are also used to train the language models. The word error rate for the indexed audio is 34%.

To obtain phoneme sequences, we use a dictionary to automatically convert the first-best transcripts from the word recognizer to phonemes. Small-scale tests indicated that this gives much better results than running a phoneme recognizer directly.

We then feed the time-marked transcripts into our indexes. We use a modified version of the AltaVista indexing engine [3]. The original version was designed to index text documents so for a given query it returned the list of documents. Our version can return multiple hits per document so as to find each location of the query words in long audio files. The indexer ranks documents using a standard *tf.idf* metric augmented by proximity information.

5 Results

Table 1 shows baseline mean average precision results for the word index and various phoneme indexing systems. The first result is for the word index. Because we are using a definition of relevance which includes plurals and possessives, we query this index with each query's plural and possessive form in addition to the word query (e.g. *clinton* generates queries *clinton*, *clinton's* and *clintons*). Note that this scheme actually results in non-zero average precision for OOV queries since one of the query words (*spore*) exists in the dictionary in its plural form (*spores*).

The rest of the results in Table 1 are for systems based on querying the phoneme index. The first system simply queries the phoneme index with phoneme sequences. Here, word queries are converted to phoneme sequences either by looking up a dictionary or by using spelling to pronunciation rules [12]. For the rest of the phoneme systems, we expand the query phoneme sequences into overlapping sub-sequences, similar to [15]. The sub-sequences are of length N phonemes overlapped by M phonemes. For example, for $N = 5$ and $M = 4$ the query *jupiter* with phoneme representation *jh uw p ah t er* is expanded as *jh uw p ah t* and *uw p ah t er*. We search for exact matches of these sequences in the phoneme index and return the hits. Since many expansions give hits in the same document, these results are merged into one hit and the scores added. We have experimented extensively with this system in previous work and found it to be a good existing approach to the OOV problem.

Table 1: Baseline results averaged over all queries for word and various phoneme indexing systems for various definitions of document.

System	**Doc**	**11-pt Avg. Prec**		
(M/N)	**Defn**	**All**	**InVoc.**	**OOV**
Words	10 sec	0.46	0.65	0.02
	Topics	0.53	0.74	0.03
Phonemes	10 sec	0.45	0.63	0.03
	Topics	0.51	0.71	0.04
Phonemes (3/2)	10 sec	0.46	0.60	0.16
	Topics	0.22	0.28	0.08
Phonemes (5/4)	10 sec	0.51	0.66	0.15
	Topics	0.56	0.72	0.17
Phonemes (7/5)	10 sec	0.47	0.62	0.11
	Topics	0.55	0.73	0.12

From Table 1 we see that the Phoneme (5/4) system produces the best results for this query set. This is in agreement with our observations on other query sets [10] and partly a function of our high OOV rate. Note also that for the most part, a less strict definition of document results in improved performance except for the Phoneme (3/2) system. In this case, too many false positives degrade the performance.

For the remainder of our experiments, we will use the Phoneme (5/4) system. We will continue however to report results for both definitions of document.

Table 2 shows results for the Phoneme (5/4) system when the list of hits for each query are resorted according to semantic distance. Although many formulations are possible, we use the L1 distance between the topic vectors for the query and each hit. We show results for varying numbers of topics T. In each case we see that sorting by semantic distance is less effective than sorting by acoustic score.

Note however that the results are affected by the number of topics used to calculate the semantic distance. This represents a trade-off between not having enough topic resolution to distinguish semantic information (too few topics) and not having enough training data to train good topic models (too many topics).

We now consider linearly combining the acoustic and semantic information. Since the semantic distance measure is a dissimilarity measure, we use the following formula:

$$\mathrm{Score} = (\mathrm{AcousticScore}) - K * (\mathrm{SemanticScore}).$$

Table 2: Results for Phoneme (5/4) system averaged over all queries. Hits are sorted by acoustic score or semantic (L1) distance for varying numbers of topics T and various definitions of document

Sort	T	**Doc Defn**	**11-pt Avg. Prec**		
			All	**InVoc.**	**OOV**
Acoustic	-	10 sec	0.51	0.66	0.15
	-	Topics	0.56	0.72	0.17
Semantic	16	10 sec	0.42	0.56	0.09
	128		0.45	0.60	0.07
	256		0.44	0.60	0.09
	16	Topics	0.46	0.60	0.14
	128		0.50	0.65	0.15
	256		0.48	0.62	0.15

Table 3 shows results using such a scheme for varying values of K. For brevity, we only show results for the semantic distance calculated using 128 topics. The results are similar however for 16 and 256 topics.

Table 3: Results for Phoneme (5/4) system averaged over all queries. Hits are sorted by a linear combination of the acoustic score and semantic score. Results shown for varying combination coefficient (K) and various definitions of document

K	**Doc Defn**	**11-pt Avg. Prec**		
		All	**InVoc.**	**OOV**
0	10 sec	0.51	0.66	0.15
1		0.53	0.68	0.16
10		0.53	0.68	0.16
100		0.53	0.69	0.16
1000		0.47	0.63	0.08
0	Topics	0.56	0.72	0.17
1		0.57	0.74	0.18
10		0.57	0.74	0.18
100		0.58	0.75	0.19
1000		0.55	0.71	0.18

The results in Table 3 show that our scheme has promise. For a wide range of combination coefficients and for several definitions of document, a small improvement over the baseline is seen. This improvement applies both to in-vocabulary and OOV queries.

Although we are encouraged by these results note, that we have not yet solved the problem of how to learn the parameters of our system for the set of all possible queries. However, converting queries and documents to the semantic space is appealing for learning approaches and may provide insight into how best to fuse semantic and acoustic information.

6 Conclusions and Future Work

We have presented a novel approach to the OOV problem which combines semantic and acoustic indexing to achieve improved performance. Given a set of acoustic matches for a query, we use PLSA to score the semantic closeness of each hit to the query. Linearly combining both scores gives improved retrieval performance.

Future work will concentrate on how best to formulate the fusion problem such that we can learn parameters which will generalize to new query sets. We will also investigate the use of hierarchical topic models for semantic representations of queries and documents.

References

[1] D. Blei and P. J. Moreno. Topic segmentation with an aspect hidden markov model. In *SIGIR2001*, 2001.

[2] Chris Buckley and Ellen Voorhees. Evaluating evaluation measure stability. In *SIGIR2000*, 2000.

[3] M. Burrows. *Method for Indexing Information of a Database*. U.S. Patent 5,745,899, 1998.

[4] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *American Society for Information Science*, 6(41):391–407, 1990.

[5] T. Hofmann. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, 1999.

[6] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR1999*, 1999.

[7] D. A. James. A system for unrestricted topic retrieval from radio news broadcasts. In *Proc. ICASSP*, 1994.

[8] T. Kemp and A. Waibel. Reducing the OOV rate in broadcast news speech recognition. In *Proc. ICSLP*, 1998.

[9] B. Logan, P. Moreno, JM. Van Thong, and E. Whittaker. An experimental study of an audio indexing system for the Web. In *Proc. ICSLP*, 2000.

[10] B. Logan, Pedro Moreno, and Om Deshmukh. Word and sub-word indexing approaches for retuding the effects of OOV queries on spoken audio. In *HLT*, 2002.

[11] B. Logan and JM. Van Thong. Confusion- based query expansion for oov words in spoken document retrieval. In *Proc. ICSLP*, 2002.

[12] V. Pagel, K. Lenzo, and A. Black. Letter to sound rules for accented lexicon compression. *Proc. ICSLP*, November 1998.

[13] P. Schaeuble and M. Wechsler. First experiences with a system for content based retrieval of information from speech recordings. In *IJCAI-95*, 1995.

[14] JM. Van Thong, D. Goddeau, A. Litvinova, B. Logan, P. Moreno, and M. Swain. Speechbot: a speech recognition based audio indexing system for the web. In *Proc. International Conference on Computer-Assisted Information Retrieval (RIAO)*, 2000.

[15] M. Witbrock and A. G. Hauptmann. Using words and phonetic strings for efficient information retrieval from imperfectly transcribed spoken documents. In *Second ACM International Conference on Digital Libraries*, 1997.

[16] P. C. Woodland, S. E. Johnson, P. Jourlin, and K. Spark Jones. Effects of out of vocabulary words in spoken document retrieval. In *SIGIR2000*, July 2000.

Table 4: Queries used in experiments

In Vocabulary	**OOV**
clinton	andrew cunanan
president clinton	dodi al fayed
lewinsky	dr steve salvatore
mir	eddie mair
starr	christina zorich
kaczynski	jonbenet
gore	kyoko altman
medicare	marina boughton
christmas	montserrat
mars	newseum
nichols	peekskill
princess diana	plavsic
supreme court	spektr
space station	spore
nasa	tamraz
saddam	
tyson	
castro	
byron	
bill clinton	
netanyahu	
nato	
gingrich	
karen maginnis	
albright	
kennedy	
de tocqueville	
pentagon	
royal family	
death penalty	
reagan	
microsoft	
dow jones	
lisa mullins	
democratic party	