Approaches to Reduce the Effects of OOV Queries on Indexed Spoken Audio

Beth Logan, Pedro Moreno, JM Van Thong
Cambridge Research Laboratory
HP Laboratories Cambridge
HPL-2003-46
March 5th , 2003*

spoken document retrieval, speech indexing, out-of-vocabulary words, OOV words

We present several novel approaches to the OOV query problem for spoken audio: indexing based on syllable-like units called particles and query expansion according to acoustic confusability for a word index. We also examine linear and OOV-based combination of indexing schemes.

We experiment on 75 hours of broadcast news, comparing our approaches to a word index, a phoneme index and a phoneme index queried with phoneme sequences. Our results show that our approaches are superior to both a word index and a phoneme index for OOV words, and have comparable performance to the sequence of phonemes scheme. The particle system has worse performance than the acoustic query expansion scheme. The best system uses word queries for in-vocabulary words and a linear combination of the phoneme sequence scheme and acoustic query expansion for OOV words. This system improved the average precision from 0.35 for a word index to 0.40.

* Internal Accession Date Only Approved for External Publication
Portions of this work were based on papers published in Human Language Technology Conference 24-27 March 2002, San Diego, CA and in the International Conference on Spoken Language Processing, September 2002, Denver, Colorado

1 Introduction

In recent years, systems to index vast audio repositories have emerged (e.g. [18], [2], [16], [5]). Typically, speech recognition is used to transcribe the audio and then standard textual information retrieval (IR) algorithms are applied. However, this approach cannot process queries which are not in the recognizer's vocabulary. This is a problem for example in broadcast news as public figures with unseen names appear over time. A typical out of vocabulary (OOV) rate for user queries could be over 10% [10], even when a large vocabulary recognizer is used.

Much effort has been devoted to the OOV problem. A popular solution is to transcribe the audio using sub-word units such as phonemes or syllables. Word queries are then converted to the sub-word units and searched for in the hypotheses. (e.g. [7], [6], [14], [20]). Additionally, to compensate for recognition errors, phonetic confusion matrices and N-best lists may be used to expand the query and document representations (e.g. [7], [11], [15]).

Although the use of sub-word units can improve retrieval, the improvement often comes at the cost of many false alarms since syllables occur much more frequently than words. For example, in [6], a false alarm rate of the order of 0.5 per hour of audio indexed is quoted for phoneme queries of length 7-11. For an index of 1,000,000 hours, this would mean that a single query might generate 500,000 or so false alarms.

A second disadvantage of phoneme-based systems is that each new query involves a search through the multiple hypotheses. The search time increases linearly with the size of the repository. A word-based system however can use an index with a relatively constant access time regardless of size. This search problem can be alleviated by building an index of sequences phonemes or syllables (e.g. [20]).

Approaches which combine word and phoneme models have also been tried (e.g. [8], [12]). Typically, linear combinations are considered. The theoretical properties of linearly combined indexes are studied in [17]. Here it is noted that the usefulness of linear combination is limited to certain situations. The main problem is that it is not known how to optimize the combination parameters for all possible queries as this set is infinite.

Other researchers have tackled the OOV query problem using IR techniques such as query expansion and stemming [21]. Query expansion, which uses documents from a different source to find words related to the query, is reliant on the quality of these additional documents. Stemming's ability to help retrieve OOV proper names is limited.

An approach related to query expansion is to change the recognizer vocabulary using documents from a parallel corpus (e.g. [9]). This has two disadvantages. First, previously recognized documents must be reprocessed if it is desired to find the OOV words in them. Second, it may be difficult to obtain enough data to train good language models which include the new words. The first problem may be less of an issue if words are hypothesized from an intermediate representation (e.g. [19]).

In this paper, we examine several strategies to attack the OOV problem. Ideally we would like to develop systems that have the low OOV rates of sub-word based systems while maintaining the good scalability, speed of search and low false alarm rate of word-based indexing systems.

We first examine a novel indexing system based on *particles*. This is a syllable-like system with particles consisting of automatically determined within-word sequences of phonemes. Our hope

is that it can find OOV queries with less false alarms than a phoneme system since it indexes frequently occurring syllables.
We then investigate a novel indexing scheme which can handle OOV words using a word index. It expands query words into in-vocabulary phrases and searches for these phrases in the word index. For example, *taliban* may be expanded to *tell a band*. The aim is to mimic mistakes the speech recognizer makes when transcribing the audio. This technique has several advantages. First, it can expand all types of OOV words and can be applied to any word index without reprocessing the audio. Second, because we use a word index, the space and time requirements are very reasonable. Third, we do not need to make decisions about which parallel document collections to use which may bias our results.
Finally, we consider simple techniques for combining the various indexing systems.

2 Particle-based Indexing

Particles are defined as within-word sequences of characters obtained from orthographic or phonetic transcriptions of words [19]. Particles are used as the recognition units in a speech recognition system which permits word-vocabulary independent speech decoding, and thus can be used to alleviate the OOV problem in spoken document retrieval applications.
Our particles are obtained from phonetic transcriptions and are learnt by decomposing words into sub-sequences of phonemes so as to maximize the leaving-one-out likelihood of a particle bigram language model. The resulting particle dictionary consists of phoneme sequences from single phonemes to full words.

Table 1: Example of particle transcription

Particles
IH_N_ W_AA_ SH_IH_NG_ T_AH_N_ T_AH_D_ EY_ AH_ K_AH_ N_G R_EH_ SH_AH_ N_AH_L_ K_AH_M_ IH_T_IY_ IH_Z_ B_AH_N_ S_T_ AH_D_IY_ IH_NG_ B_AE_D_ AO_R_ W_ER_S_ B_IH_ HH_EY_V_ Y_ER_
Word transcription
IN WASHINGTON TODAY A CONGRESSIONAL COMMITTEE HAS BEEN STUDYING BAD OR WORSE BEHAVIOR

This particle representation is quite flexible. If the dictionary of particles only contains single phoneme particles then the particle recognizer behaves like a phonetic recognizer. If the particles are as long as words then it behaves like a word recognizer. In effect a particle based recognition system behaves like a syllable based speech recognizer where the basic units are automatically learned from textual data.
Once the dictionary of particles is defined it can be used to translate a word corpus to particles (see 1). This new particle corpora can then be used to train acoustic and language models in the

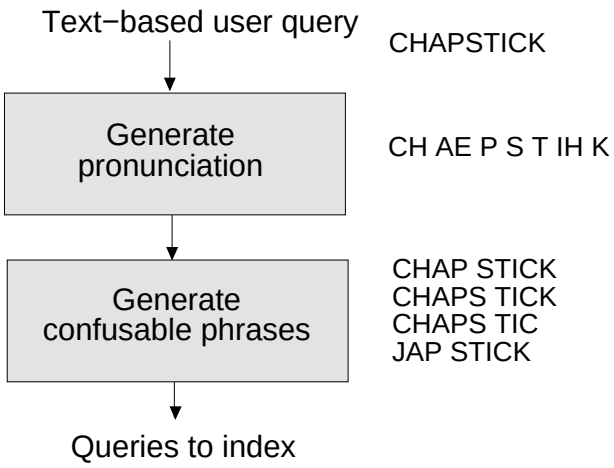

Figure 1: Query expansion algorithm

usual manner. To perform indexing, we recognize audio using these particle models and insert the particle transcripts into the index. The particle representation for word queries is found either by looking them up in the dictionary or by choosing the most likely particle decomposition of the words' pronunciation. The sequence of particles then form query terms that can be search in the index.

Our analysis of the OOV shows that we can recover about 20% of the OOV words when re-decoding the word sequences from the particle sequences transcribing the indexed documents [19]. A particle based index, with a simple look-up of the query terms in a hash table will lead to a similar accuracy improvement for the retrieval of documents containing OOV query terms, as shown in 5.

3 Confusion-based Query Expansion

Our second novel approach to the OOV problem is to expand word queries into in-vocabulary phrases according to intrinsic acoustic confusability and language model scores. We then use these phrases to query a word index. Our query expansion algorithm is shown in Figure 1. The steps are as follows.

First, given a query word or query phrase, we convert it into a sequence of phonemes. For each query word, if we can find it in a dictionary, we use the most likely pronunciation. Otherwise, we automatically generate its pronunciation using Pagel et al's algorithm [13].

Given this pronunciation we now seek confusable in-vocabulary phrases generated using the recognizer's dictionary and language model. We achieve this by first using a modified version of our existing Viterbi decoder to generate a lattice of word hypotheses for the query. We then run an A* search to generate the N-best confusable phrases from this lattice.

Normally, the decoder scores acoustic features against all combinations of words in the dictionary according to acoustic and language model scores. In our modified decoder, the input 'feature' for each query is its pronunciation and the 'acoustic' score between it and words in the

dictionary is determined using a confusion matrix. We use a language model as usual and also prune paths which have likelihood below a given threshold.
Our confusion matrix is obtained using the clean speech TIMIT corpora [1] and gives scores for the confusions between phonemes as well as the likelihood of inserting and deleting each phoneme. We experimented with confusion matrices obtained from more broadcast news-like sources but found little impact on results.
Although our search of the space of confusable phrases is not exact due to pruning, it gives believable results. We tuned the language model weights and pruning thresholds on a held out set of queries. In practice, we obtained similar retrieval performance for a wide range of parameters. Table 2 shows typical query expansions obtained using our algorithm. If we implement our program as a server with the language models permanently loaded in memory, the computational requirements to generate each set of phrases are very small.

Table 2: Typical confusable phrases generated by our algorithm

Query	**Expansions**
blackfeet	black feet, black feat, black wheat
looper	luper, looped, loop are
yassar	yasir, yasser, ya sir
aerosmith	aerosmith, aero smith, arrow smith, aero smyth, arrow smyth
afghanistan	afghanistan, afghan stan, afghan austin, afghan us tan, afghan bran
bilbao	biller, bill bow, bill bough, bill bao, bil bow

We use the set of confusable phrases to query our index, searching for exact matches of each phrase.

4 Experimental Setup

We examine the operation of our algorithms 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 transcriptions supplied by the LDC [1]. The transcripts provide us with the ground truth and allow us to automatically estimate precision, recall and false alarm rates. 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 Document and Relevance Definitions

We index 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 define a document as a 10s clip and define it as relevant if the query word was spoken within it according to the transcripts.

4.3 Evaluation Metric

Our primary 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.
Because we are examining sub-word-based systems for which false alarms are a major problem, we also explicitly report the number of false alarms even though 11-pt average precision implicitly includes this quantity. The number of false alarms for a given query is defined as the number of incorrect hits divided by the total number of hits returned. We average our results over all queries.
For completeness, we also show recall, top 5 precision, and top 10 precision. These measures are also implicitly included in 11-pt average precision since it is an overall figure of merit. For all metrics we average over all queries.

4.4 Query Selection

In [3] 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 unambiguous 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 hits for each query similar to what would appear on a Web page of hits.

For our database, comparison of the ground truth to the word recognition dictionary yields 23 suitable OOV queries (*i.e.* proper names with at least 10 hits). We choose the remaining 27 queries as the most frequent in-vocabulary queries to the *SpeechBot*[1] public site which have at least 10 hits and are proper names. The result is a query set with 47 single word queries and 3 two word queries.
The *SpeechBot* site has been in operation since December 1999 and is therefore a good source of real-world queries. According to its user logs, almost 80% of user queries are two words or less. Note that our query OOV rate of around 50% is much higher than the 13% rate observed on the site[10].

[1] www.speechbot.com

4.5 Indexing Systems

We build three indexes: a word index, a particle index and a phoneme index. These are constructed as described below.

To construct the word index, we first transcribe the audio 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 64,000 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%.

For the particle system, we transcribe the audio using our particle recognizer. This is trained on the same audio and text corpora as the word recognizer. In our implementation we use a dictionary of about 7,000 particles. We have found that this dictionary size with particles of length from one to three phonemes yields optimal results.

Finally, our third system indexes phoneme sequences. We do not run a phoneme recognizer. Instead, we use a dictionary to automatically convert the transcripts from the word recognizer in our first system to phonemes. Small-scale tests indicated that this gives better results than running a phoneme recognizer.

Having obtained three sets of time-marked transcriptions for the audio, we then build three indexes. We use a modified version of the AltaVista indexing engine [4]. 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* IR metric augmented by information about the proximity of query terms.

In our experiments we examine five basic indexing systems plus combinations of these. The first three systems are as described above: a word index with word queries, a particle index with particle queries and a phoneme index with phoneme queries.

We additionally examine two techniques of acoustic query expansion. First we study our word expansion technique in which a word index is queried with confusable phrases derived by our algorithm as described in Section 3.

We also study querying the phoneme index with queries expanded into overlapping sequences of phonemes, similar to [20][2]. Here, word queries are converted to phonemes either by looking up a dictionary or by using spelling to pronunciation rules [13]. Each query is further expanded into sequences of 5 phonemes overlapped by 4 phonemes. For example, the sequence *jh uw p ah t er* is expanded as *jh uw p ah t* and *uw p ah t er*. We then search for exact matches of these sequences in the phoneme index. Since many expansions give hits in the same document, these results are merged into one hit and the scores added. This system is meant to serve as an example of a good existing approach to the query OOV problem so our choice of expansion and overlap length is tuned to give the best results on our database.

For clarity, Table 3 enumerates the five basic indexing schemes studied. As described in the

[2] In addition to querying with phoneme sequences, Witbrock additionally indexed phoneme sequences but this is an implementation detail.

results, we also examine combinations of some of these schemes.

Table 3: Characteristics of the five basic indexing schemes studied.

Scheme	Index	Queries	Example Query
Word	Word	Word	*"peekskill"*
Particles	Particles	Particles	*"p_iy_", "k_s_", "k_ih_l_"*
Phonemes	Phonemes	Phonemes	*"p iy k s k ih l "*
Confusions	Word	Confusable phrases	*"peeks kill", "pig skill", ...*
Phonemes (5/4)	Phonemes	Phoneme sequences	*"p iy k s k", "y k s k ih", "k s k ih l"*

5 Results

In this section, we describe the results of our experiments.

5.1 Particle Index

We first study the performance of the particle index and compare it to that of the word and phonemes indexes. Figure 2 and the first three lines of Table 4 show the performance of the word, particle and phoneme indexing systems averaged over all queries. We see that the word-based system has the best performance overall. However, as Figures 3 and 4 and Tables 5 and 6 demonstrate, the performance of the particle and phoneme systems are better than the word system for OOV queries and worse for in-vocabulary queries. The particle system performs slightly better than the phoneme system on OOV queries.

Table 4: Results averaged over all queries for word, particle, phoneme and phoneme (5/4) indexing systems.

System	11-pt Avg. Precision	Recall	Top 5 Precision	Top 10 Precision	False Alarms
Word	0.35	0.39	0.50	0.48	0.08
Particles	0.33	0.39	0.51	0.47	0.21
Phonemes	0.32	0.42	0.48	0.44	0.27
Phonemes (5/4)	0.35	0.48	0.48	0.45	0.57

Although the particle index has better performance than a phoneme index for OOV words, a more fair comparison is to the phoneme sequence query expansion scheme described in Section 4.5 which expands queries to syllable-like units. The preceding Tables and Figures also show results for such a system. It is denoted 'Phonemes (5/4)' since we study expansions to sequences of 5 phonemes overlapped by 4 phonemes. From these results we see that using sequences of

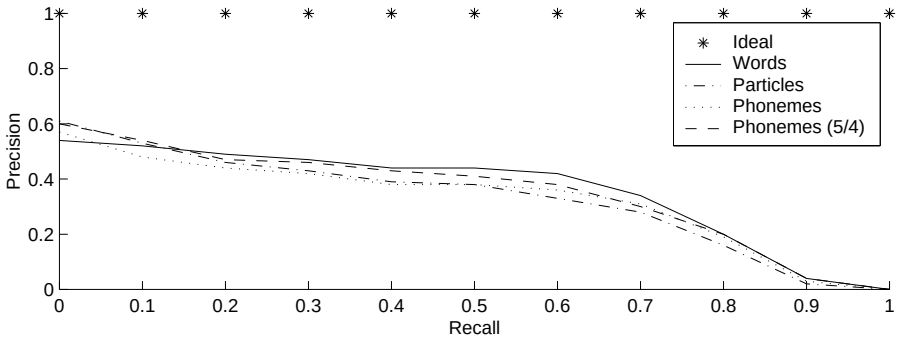

Figure 2: Precision-Recall curves averaged over all queries for the word, particle, phoneme and phoneme (5/4) indexing systems and the ideal system.

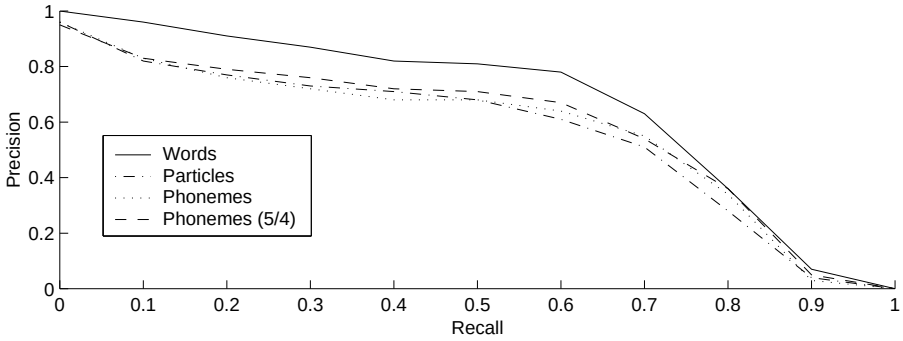

Figure 3: Precision-Recall curves for the in-dictionary queries for the word, particle, phoneme and phoneme (5/4) indexing systems.

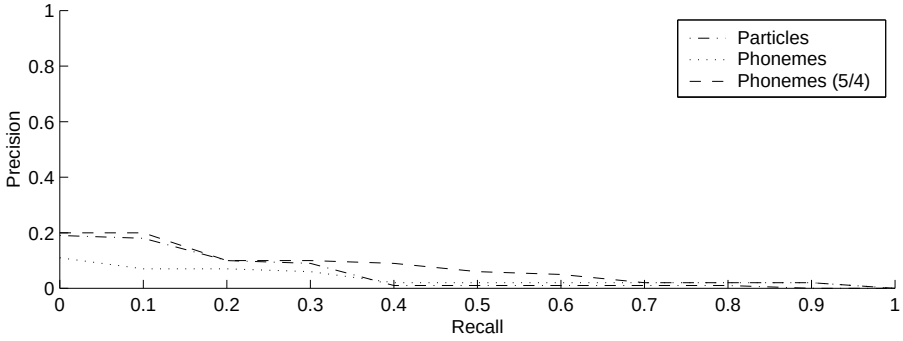

Figure 4: Precision-Recall curves for the OOV queries for the word, particle, phoneme and phoneme (5/4) indexing systems.

Table 5: Results averaged over all in-dictionary queries for the word, particle, phoneme and phoneme (5/4) indexing systems.

System	11-pt Avg. Precision	Recall	Top 5 Precision	Top 10 Precision	False Alarms
Word	0.66	0.73	0.92	0.89	0.14
Particles	0.55	0.65	0.82	0.79	0.24
Phonemes	0.56	0.71	0.84	0.77	0.29
Phonemes (5/4)	0.58	0.76	0.80	0.76	0.56

Table 6: Results averaged over all OOV queries for the word, particle, phoneme and phoneme (5/4) indexing systems.

System	11-pt Avg. Precision	Recall	Top 5 Precision	Top 10 Precision	False Alarms
Word	0.00	0.00	0.00	0.00	0.00
Particles	0.06	0.09	0.15	0.10	0.17
Phonemes	0.04	0.08	0.07	0.05	0.24
Phonemes (5/4)	0.08	0.14	0.10	0.09	0.58

phonemes can improve the average precision. The system is at least as good as using particles for OOV words and equivalent to words overall.

However, although both the word index and phoneme sequence system have an average precision of 0.35, they operate at different recall and false alarm levels. From Table 4, we see that using phoneme sequences rather than words improves the recall from 0.39 to 0.48. However, this comes at a cost of increasing the number of false alarms from 0.08 to 0.57. In some applications this increase in false alarms could be crippling. In others it might be justified by the increase in recall.

Similarly, for OOV words only, although the average precision for the phoneme (5/4) scheme (0.08) is slightly better than that of the particle index (0.06), there is a recall-false alarm trade-off. The phoneme (5/4) system has recall 0.14 with 58% false alarms whereas the particle system has recall of only 0.09 but only 17% false alarms. For some applications, the particle system may be more useful.

5.2 Confusion-based Query Expansion

We now examine our second approach to the OOV problem, namely expanding word queries into in-vocabulary phrases and querying a word index as described in Section 3. Table 7 shows the 11-pt average precision, recall and false alarms averaged over all queries for standard word queries and queries expanded to various depths using our algorithm. We see that our query expansion scheme results in improved performance for 10 confusions. For 100 confusions, however, the performance is worse than simply using word queries due to excessive false alarms.

Table 7: Results averaged over all queries for word queries and confusion-based expanded queries to the word index; All queries expanded.

Query Expansion	Nr. Conf.	11-pt Av.Prec.	Recall	Top 5 Precision	Top 10 Precision	False Alarms
None (words)	-	0.35	0.39	0.50	0.48	0.08
Confusions	1	0.35	0.40	0.50	0.48	0.15
	10	0.37	0.44	0.49	0.46	0.29
	100	0.30	0.47	0.37	0.34	0.53

Examination of the results reveal that it is never helpful to use query expansion for in-vocabulary words. We therefore consider only using query expansion for OOV words and a simple word query otherwise. Table 8 shows results for such a scheme. Note that these results are averaged over all queries but only OOV queries have been expanded. Here we see that our technique provides a definite improvement. This is also evident in Figures 5 and 6 which show precision *vs* recall curves for this scheme. Figures 5 shows the results for all queries while Figure 6 shows results only for OOV queries.

Table 8: Results averaged over all queries for word queries and confusion-based expanded queries to the word index; Only OOV queries expanded.

Query Expansion	Nr. Conf.	11-pt Av.Prec.	Recall	Top 5 Precision	Top 10 Precision	False Alarms
None (words)	-	0.35	0.39	0.50	0.48	0.08
Confusions	1	0.37	0.42	0.52	0.49	0.15
	10	0.38	0.43	0.55	0.51	0.26
	100	0.37	0.46	0.52	0.49	0.41

5.3 Index Combination

In the previous section, we noted that the best performance was obtained by the use of a word index with word queries for in-vocabulary words, our confusable phrase query expansion scheme otherwise. This can be thought of as a form of index combination in which query hits from different indexing schemes are selected depending on whether the query words are in-vocabulary or not. We now compare this type of scheme to more standard linear index combination where the scores of the hits from two indexing schemes are weighted and added together to give a new set of hits.

Table 9 summarizes the results of combination experiments. The first line repeats the results for the word index system with word queries. The second line shows results for the best linear combination technique in which a word index is linearly combined with the phoneme (5/4) system. These results are from an exhaustive search of the space of all possible combination

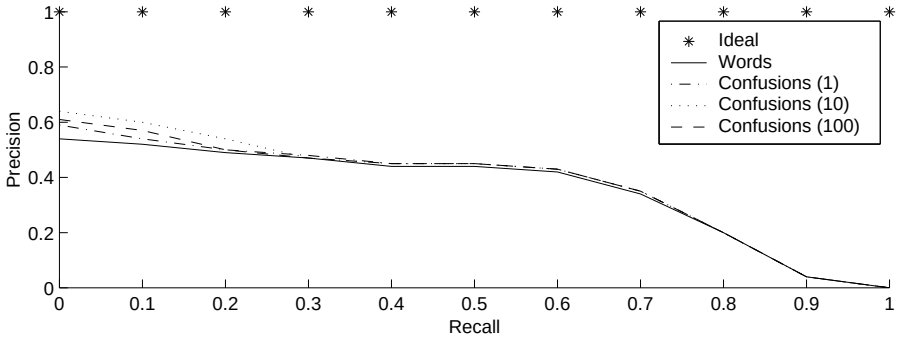

Figure 5: Precision-Recall curves averaged over all queries for word queries and confusion-based expanded queries to the word index; Only OOV queries expanded.

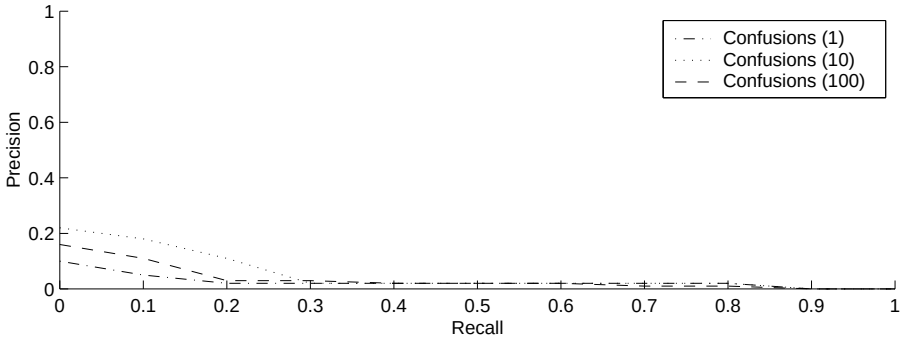

Figure 6: Precision-Recall curves averaged over all OOV queries for word queries and confusion-based expanded queries to the word index; Only OOV queries expanded.

Table 9: Results averaged over all queries for the baseline word index system and various combinations of systems.

Combination	Query Expansion	11-pt Avg. Precision	Recall	Top 5 Precision	Top 10 Precision	False Alarms
None	None	0.35	0.39	0.50	0.48	0.08
Linear	Phonemes (5/4)	0.39	0.48	0.54	0.51	0.57
OOV-based	Word confusions	0.38	0.43	0.55	0.51	0.26
OOV-based	Phonemes (5/4)	0.39	0.46	0.56	0.53	0.34
OOV-based	Phonemes (5/4) + Word confusions	0.40	0.47	0.60	0.54	0.38

coefficients. This result is therefore an upper bound, obtainable only if the coefficients could be optimized on a development query set. We see that a marked improvement over using solely the word index or the phoneme (5/4) scheme (line 4 of Table 4) is possible using linear combination.
The next two lines of Table 9 show the results of combining a word index with the phoneme (5/4) system or with our confusable word expansion scheme based on whether the word in in-vocabulary or OOV. We denote this type of combination 'OOV-based'. For in-vocabulary words, we query the word index, otherwise we use phoneme or confusable word expansions. We show results for the best word confusions scheme which was 10 confusions.
These results are similar to the best linear combination technique and have the added advantage that they do not rely on the use of a development set and could therefore be recommended for all query types. The phoneme (5/4) expansion scheme is slightly better than the confusable word expansion scheme when only used on OOV words.
Finally, the last line of Table 9 shows the result of combining the word system and the two query expansion schemes. For in-vocabulary words, we again query the word index. For OOV words, we linearly combine the hits from the phoneme (5/4) and word confusion schemes. Specifically, we add the scores for the documents returned by each scheme. The resulting average precision of 0.40 is slightly better than either scheme, indicating the two approaches are somewhat additive. It is also the best average precision obtained overall and a marked improvement over the baseline of 0.35 obtained using simply a word index with word queries.
Figure 7 shows the precision-recall curves for the various combination schemes. Figure 8 shows curves for OOV queries only.

5.4 Discussion

In the previous sections we have seen that our proposed approaches to the OOV problem while certainly providing better results than simply using a word index, are at best comparable to the phoneme sequence (phoneme (5/4)) scheme. We have also seen that the best result is obtained by combining systems. This is key and can be partly explained as follows.
Close examination of the hits for queries expanded into confusable phrases using our technique highlighted that a bad initial pronunciation could be fatal. For example, the OOV query *lider-*

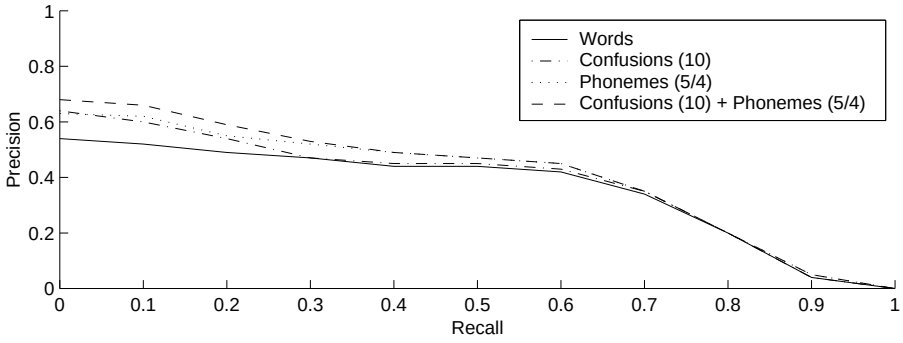

Figure 7: Precision-Recall curves averaged over all queries for word queries, confusion-based expanded queries, phoneme (5/4) expanded queries and their combination; Only OOV queries expanded.

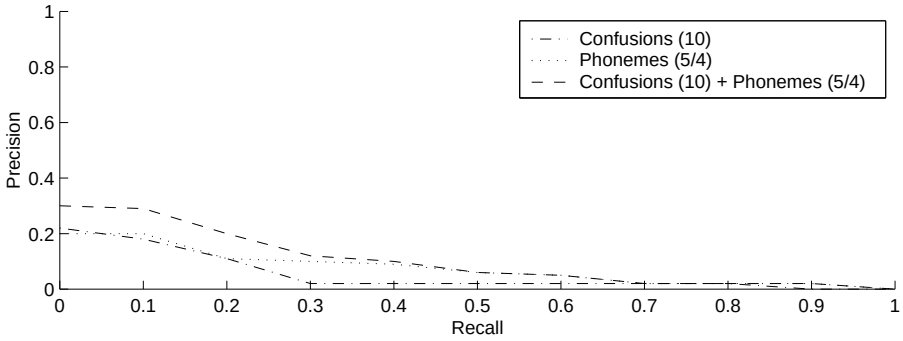

Figure 8: Precision-Recall curves averaged over all OOV queries for confusion-based expanded queries, phoneme (5/4) expanded queries; Only OOV queries expanded.

man appears in the speech recognition transcripts as *liederman*, that is *l iy d er m ah n*. However, the automatically generated pronunciation for *liderman* is *l ay d er m ah n*, generating confusable queries *leiberman*, *leiter mun* and so on. If a full Viterbi search without pruning were conducted or the initial pronunciation were better, *liederman* would more likely appear as one of the proposed queries. This problem is particularly acute for foreign names and other unfamiliar words which are prone to have poor pronunciations unless accounted for when learning spelling to pronunciation rules.

The phoneme (5/4) system, however, expands the pronunciation for an OOV word into overlapping sequences of phonemes as described in Section 4.5. We then search for exact matches of these sequences in the phoneme index, summing the scores of hits which occur in the same document. This means that the resulting score is a good reflection of how likely the phonemes in the OOV query are to be found in the indexed documents. Even if only part of the pronunciation is correct, there is a chance that some of the sequences will be found in the index. This increased recall comes at the cost of more false positives however. By combining systems, we combine some of the advantages of both schemes.

6 Conclusions and Future Work

We have presented several novel approaches to the OOV query problem for audio indexing: indexing based on syllable-like units called particles and query expansion according to acoustic confusability for a word index. We examined the performance of these schemes on 75 hours of broadcast news, comparing their performance to a standard word-based index, a phoneme index and a phoneme index queried with overlapping phoneme sequences. We also examined linear and OOV-based combination of indexing schemes.

For our query set, which has an OOV rate of around 50%, we found that both the particle index and our acoustic query expansion scheme were superior to both a word index and a phoneme index, and had comparable performance to the overlapping sequences of phonemes system. The particle system had worse performance than the acoustic query expansion scheme, but operated at a lower false alarm rate which could be important for some applications.

When combining systems, we found that detecting the query word as OOV and using the phonetic, acoustic expansion or particle system for that query works as well as using an optimal linear weighting scheme. The best system overall was was a combination system which used word queries for in-vocabulary words and a linear combination of the phoneme sequence scheme and acoustic query expansion for OOV words. This scheme improved the average precision from 0.35 for a simple word index to 0.40.

Many directions are possible for future work. First, we have not deeply explored the use of the particle system introduced in this paper. We regard our current study quite preliminary and intend to investigate this approach more in the future. Second, our experiments have highlighted that the confusion expansion scheme would benefit from improved pronunciations of OOV queries. We will also further investigate index combination, exploring more sophisticated techniques based on data fusion and Bayesian mixing of classifiers.

Finally, our attempts to compensate for OOV words have thus far been based on acoustic information. We feel that a more robust solution would additionally incorporate semantic information

about the query and intend to explore this direction in the future.

7 Acknowledgments

We acknowledge the help of our intern Om Deshmukh, now a PhD student at the University of Maryland, who helped run some of the experiments with the phoneme index.

References

[1] Linguistic data consortium. http://www.ldc.upenn.edu.

[2] D. Abberley, G. Cook, S. Renals, and T. Robinson. Retrieval of broadcast news documents with the THISL system. In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*, 1999.

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

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

[5] S-F Chang, Q. Huang, T. Huang, A. Puri, and B. Shahraray. Multimedia search and retrieval. In *Multimedia Systems, Standards and Networks*, 2000.

[6] M. Clements, P. S. Cardillo, and M. S. Miller. Phonetic searching vs. LVCSR: How to find what you really want in audio archives. In *20th Annual AVIOS Conference*, 2001.

[7] D. A. James. A system for unrestricted topic retrieval from radio news broadcasts. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1994.

[8] G. J. F. Jones, J. T. Foote, K. Spark Jones, and S. J. Young. Retrieving spoken documents by combining multiple index sources. In *SIGIR1996*, 1996.

[9] T. Kemp and A. Waibel. Reducing the OOV rate in broadcast news speech recognition. In *International Conference on Spoken Language Processing*, 1998.

[10] B. Logan, P. Moreno, JM. Van Thong, and E. Whittaker. An experimental study of an audio indexing system for the Web. In *International Conference on Spoken Language Processing*, 2000.

[11] K. Ng and V. Zue. Towards robust methods for spoken document retrieval. In *International Conference on Spoken Language Processing*, 1998.

[12] K. Ng and V. Zue. Information fusion for spoken document retrieval. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000.

[13] V. Pagel, K. Lenzo, and A. Black. Letter to sound rules for accented lexicon compression. *International Conference on Spoken Language Processing*, November 1998.

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

[15] S. Srinivasan and D. Petkovic. Phonetic confusion matrix based spoken document retrieval. In *SIGIR2000*, 2000.

[16] 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.

[17] C. C. Vogt and G. W. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.

[18] H. D. Wactlar, A. G. Hauptmann, and M. J. Witbrock. Informedia: News-on-demand experiments in speech recognition. In *DARPA Speech Recognition Workshop*, 1996.

[19] E. W. D. Whittaker, JM. Van Thong, and P. J. Moreno. Vocabulary independent speech recognition using particles. In *2001 Automatic Speech Recognition and Understanding Workshop*, Trento, Italy, 2001.

[20] 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.

[21] 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.

A List of Queries

Table 10: Queries to the system

In dictionary	Count	Out of Dictionary	Count
bill clinton	56	cunanan	70
al gore	31	mair	57
clinton	626	fayed	52
microsoft	40	dodi	37
israel	104	tamraz	26
egypt	15	peekskill	23
montreal	23	sankara	18
china	226	plavsic	18
nasdaq	53	reineck	13
paris	101	rutan	16
christmas	97	fenphen	16
jesus	11	lia	13
kennedy	48	mcaleese	14
france	62	bilbao	13
england	86	reesjones	13
germany	37	cortisol	10
switzerland	13	onondaga	10
india	39	hightech	12
nasa	73	zorich	12
australia	25	liderman	12
mexico	121	montserrat	11
cuba	141	boughton	10
florida	198	pazuto	10
canada	106		
iran	66		
texas	151		
stock market	41		