CONFUSION-BASED QUERY EXPANSION FOR OOV WORDS IN SPOKEN DOCUMENT RETRIEVAL

Beth Logan and JM Van Thong

Hewlett-Packard Laboratories
One Cambridge Center
Cambridge MA 02142, U.S.A.
Beth.Logan@hp.com, JM.VanThong@hp.com

ABSTRACT

We present a novel approach to the out of vocabulary (OOV) query problem for audio indexing. Our technique first builds a word index for the audio using speech recognition. It then expands query words into in-vocabulary phrases according to intrinsic acoustic confusability and language model scores. The aim is to mimic the mistakes the speech recognizer makes when transcribing the OOV words.

We present results of retrieval experiments on a broadcast news repository of 75 hours. Our results indicate that our approach is promising. Our technique is better than simply using word queries and only slightly worse than a more sophisticated scheme which expands queries into overlapping sequences of phonemes. We can also combine our technique with the phoneme indexing system to further improve performance. Finally, our approach is simple, requires only a word index be built for the audio and has little computational overhead.

1. INTRODUCTION

In recent years, systems to index vast audio repositories have emerged (e.g., [1]). 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% [2], 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., [3], [4], [5]). 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., [3], [6]).

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. A second disadvantage 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 of phonemes or syllables (e.g., [5]).

Approaches which combine word and phoneme models have also been tried (e.g., [7], [8]). Typically, linear combinations are considered. The theoretical properties of linearly combined indexes are studied in [9]. 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 [10]. 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., [11]). 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., [12]).

In this paper, we propose to expand query words into in-vocabulary phrases and to search for these phrases in a word index. For example, Talibum may be expanded to tell a band. The aim is to try 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. Fourth, our technique can be combined with other approaches.

2. CONFUSION-BASED QUERY EXPANSION

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. At present we generate only one pronunciation per query. For each 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 lan-
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 is the query’s pronunciation and the ‘acoustic’ score between it and words in the dictionary is determined using a confusion matrix. As usual, we prune paths which fall below a given threshold.

Our confusion matrix is obtained using the clean speech TIMIT corpora [14] 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 on a held out set of queries. In practice, we obtained similar retrieval performance for a wide range of parameters. Table 1 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.

### 3. EXPERIMENTS

We index audio documents which are at least half an hour long. Our current user interface does not segment these into topics but instead 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.

### 3.2. Document and Relevance Definitions

We index audio documents which are at least half an hour long. Our current user interface does not segment these into topics but instead 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.

### 3.3. Evaluation Metric

Our evaluation metric is 11-pt average precision. This is an estimate of the area under a precision recall curve. The greater this area, the better the system. An ideal system has average precision 1.0. We average our results over all queries.

### 3.4. Query Selection

In [15] 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.

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 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 for over two years 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[2].

### 3.5. Indexing

To index the audio, we transcribe it 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%.

Having obtained transcriptions for the audio, we then feed them into an index. We use a modified version of the AltaVista indexing engine [16]. 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 proximity information.

We examine three indexing systems. The first is a standard word index with word queries. The second is a standard word

---

Table 1: Typical confusable phrases generated by our algorithm

<table>
<thead>
<tr>
<th>Query</th>
<th>Expansions</th>
</tr>
</thead>
<tbody>
<tr>
<td>blackfeet</td>
<td>black feet, black feat, black wheat</td>
</tr>
<tr>
<td>looper</td>
<td>luper, looped, loop are</td>
</tr>
<tr>
<td>yassar</td>
<td>yasir, yasser, ya sir</td>
</tr>
</tbody>
</table>

---

1www.speechbot.com
index with queries formed by expanding words into in-vocabulary phrases using our algorithm.

The third system uses overlapping sequences of phonemes, similar to [5]. We build an index of phonemes, deriving these directly from the word transcripts using a dictionary. To query this index, 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 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.

3.6. Results

Table 2 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. Close examination of the results shows that it is never helpful to use query expansion for in-vocabulary words.

<table>
<thead>
<tr>
<th>Query Expansion</th>
<th>Nr. Conf.</th>
<th>11-pt Av.Prec.</th>
<th>Recall</th>
<th>False Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (words)</td>
<td>-</td>
<td>0.35</td>
<td>0.39</td>
<td>0.08</td>
</tr>
<tr>
<td>Confusions</td>
<td>1</td>
<td>0.35</td>
<td>0.40</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.37</td>
<td>0.44</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.30</td>
<td>0.47</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Table 2. Results averaged over all queries for word queries and confusion-based expanded queries; All queries expanded.

We therefore consider only using query expansion for OOV words and a simple word query otherwise. Table 3 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 2 and 3 which show precision vs recall curves for this scheme. Figures 2 shows the results for all queries while Figure 3 shows results only for OOV queries.

<table>
<thead>
<tr>
<th>Query Expansion</th>
<th>Nr. Conf.</th>
<th>11-pt Av.Prec.</th>
<th>Recall</th>
<th>False Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (words)</td>
<td>-</td>
<td>0.35</td>
<td>0.39</td>
<td>0.08</td>
</tr>
<tr>
<td>Confusions</td>
<td>1</td>
<td>0.37</td>
<td>0.42</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.38</td>
<td>0.43</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.37</td>
<td>0.46</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Table 3. Results averaged over all queries for word queries and confusion-based expanded queries; Only OOV queries expanded.

We now compare the performance of our best system to a system in which the OOV queries are expanded into phoneme sequences and used to search a phoneme index as described in Section 3.5. Table 4 and Figures 4 and 5 show these results. We see that our technique has slightly worse performance than the phoneme expansion scheme. Also shown in these figures and table is the result of combining our best system with the phoneme expansion system. We use simple linear combination where for OOV words, we add the scores for the documents returned by each scheme. Here the average precision is slightly better than either scheme, indicating the two approaches are somewhat additive.

<table>
<thead>
<tr>
<th>Query Expansion</th>
<th>Nr. Conf.</th>
<th>11-pt Av.Prec.</th>
<th>Recall</th>
<th>False Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confusions</td>
<td>10</td>
<td>0.39</td>
<td>0.43</td>
<td>0.26</td>
</tr>
<tr>
<td>Phoneme sequences</td>
<td>-</td>
<td>0.39</td>
<td>0.46</td>
<td>0.34</td>
</tr>
<tr>
<td>Combination</td>
<td>10</td>
<td>0.40</td>
<td>0.47</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Table 4. Results averaged over all queries for confusion-based expanded queries, phoneme sequence expanded queries and their combination; Only OOV queries expanded.

4. DISCUSSION

While it appears that our query expansion scheme is promising, it clearly does not solve the OOV problem. Close examination of the successful and unsuccessful proposed confusable queries highlighted that a bad initial pronunciation could be fatal. For example, the OOV query liderman appears in the speech recognition transcripts as liederman, that is L I Y D ER M AH N. However, the automatically generated pronunciation for liderman is L A Y D ER M HA N, generating confusable queries leberman, leiter mun and so on. If a full Viterbi search without pruning were conducted or the initial pronunciation were better, liderman would more likely
appears as one of the proposed queries. This problem is particularly acute for foreign names which are prone to have poor pronunciations.

The phoneme indexing scheme is more robust to this type of error since it expands the whole word as overlapping sequences of phonemes. If part of the word pronunciation is correct there is more chance of it matching the speech recognition transcription. This robustness comes at some cost however. The phoneme indexing scheme requires that a phoneme index be stored in addition to a word index. If a vector-space retrieval model is used, an efficient implementation would require each sequence of 5 phonemes to be indexed. Since these overlap by 4 phonemes, the final index would be 5 times larger than the original index. Also, because of the large amount of overlap in the phoneme expansion, many queries to the index are redundant, returning the same document.

5. CONCLUSIONS AND FUTURE WORK

We have presented a novel approach to the OOV query problem for audio indexing. Our technique expands query words into in-vocabulary words or phrases according to intrinsic acoustic confusability and language models. These phrases are then used to query a word index built from the audio using speech recognition. Our aim is to mimic the mistakes the recognizer makes when transcribing the audio. We presented results on a broadcast news repository of 75 hours. Our results indicate that while our approach is better than simply using word queries, it performs slightly worse than a more sophisticated scheme which expands queries into overlapping sequences of phonemes. The linear combination of our system and this phoneme system results in improved performance however. Our technique is also simple, requires no more storage than the original word index for the audio and has little computational overhead.

Future work will focus on improving the quality of pronunciations generated for the queries since our system relies heavily on these. We will also investigate combining our technique with other indexing schemes and explore its performance on more challenging databases.

6. ACKNOWLEDGMENTS

Thanks are due to Pedro Moreno for helpful discussions.

7. REFERENCES