IMPROVEMENTS IN PHONE BASED AUDIO SEARCH VIA CONSTRAINED MATCH WITH HIGH ORDER CONFUSION ESTIMATES

Upendra V. Chaudhari and Michael Picheny

IBM T.J. Watson Research Center
1101 Kitchawan Road, Rt. 134
Yorktown Heights, NY 10598
{uvc,picheny}@us.ibm.com

ABSTRACT

This paper investigates an approximate similarity measure for searching in phone based audio transcripts. The baseline method combines elements found in the literature to form an approach based on a phonetic confusion matrix that is used to determine the similarity of an audio document and a query, both of which are parsed into phone N-grams. Experimental results show comparable performance to other approaches in the literature. Extensions of the approach are developed based on a constrained form of the similarity measure that can take into consideration the system dependent errors that can occur. This is done by accounting for higher order confusions, namely of phone bi-grams and tri-grams. Results show improved performance across a variety of system configurations.

Index Terms— Phone, search, indexing, approximate

1. INTRODUCTION

The ability to search in audio is important both from a business and consumer standpoint. This paper explores a method for approximate search in audio, collected in a typical context where large volumes of data are processed through a speech recognizer, or perhaps a phone recognizer, to generate transcripts. Any resulting word based transcripts can be expanded into phone based transcripts. These are then indexed, so that information retrieval can be performed at a later date. We focus on the case where a speech recognizer is used, noting that in the broader context, certain queries may require the power of a language model, for example to disambiguate various homophones or where word boundary information is important. However, in real world scenarios, data can be collected in heterogeneous environments, at multiple times, in multiple locations, and processed with a variety of recognition systems. These systems will most likely differ in their properties, and in particular their vocabularies. That is, it should not be assumed that the same recognition system will be used at all times. When the search terms are part of the recognition vocabulary, the problem is relatively straightforward, however one still must deal with recognition errors. When this is not the case techniques must be developed to cope with the out of vocabulary (OOV) queries. An audio indexing system would likely combine these with methods of addressing in-vocabulary queries, giving more flexibility. Much work exists in the literature addressing aspects of the OOV search problem as well as search in the presence of recognition errors. In [1], a similarity measure based on a phone confusion matrix is developed and shown to be quite effective in being able to match sequences of phones approximately. In [2], errors within particular phone classes are also handled. [3] compares a variety of indexing techniques based on sub-word units, as does [1]. Sub-word lattice based techniques [4] have also performed well, as have techniques based on confusion networks [5] [6], with their ability to capture complex decoding errors. The baseline approach in this paper combines and extends some of the techniques presented in [1] and [3] to construct a method incorporating approximate query match on an N-gram document phone index. One of the goals is to balance the ability to search accurately with the cost of ingesting and indexing the audio data. Thus, the results in this paper are based on 1-best outputs of the recognizers.

First, in sections 2 and 3, the initial technique is described and evaluated in a number of situations to elucidate the performance and stability of the approach. In particular results are given on in-vocabulary and out-of-vocabulary (OOV) queries as well as with respect to varying ASR system configurations, as may be used in the audio data ingestion phase.

Then, in section 4, extensions of the method are developed to take into consideration the higher order confusions that the ASR systems could make. These take the general form of phone sequence confusions that are learned from training data. Note that to use this technique, we do not need to decode lattices in the ingestion phase. The method is developed in conjunction with a modification that replaces the initial edit distance-like computation with an alignment similarity that can account for the higher order confusions, yielding an improved approximate match.

2. BASELINE APPROACH

The baseline method incorporates elements from work published in the literature. In [1], a phonetic confusion matrix is used together with a weighted edit distance-like computation as a measure of phone sequence similarity, or approximate match. In [3], queries to be matched are parsed into N-grams which are then searched for, using an exact match, in a phoneme index. The approach used in this paper first parses the documents to be searched into N-grams and builds an inverted index to the documents. Queries are then also parsed into N-grams and the approximate search is used to match each N-gram in the query to elements in the index. The resulting approximate scores for the N-grams in the query are then combined to form the final document score. Given scores, accept and reject mechanisms can be used to determine the set of returned documents for a query. We first compute the best score possible for the query, which is a document and index independent value. This value could be referred to as the self match score. The documents with the top score (multiple documents can achieve the top score) are returned, unless the score is less than half of the best score. We avoid the need to train a threshold. While, this is a computationally intensive search procedure, the extensions and alternatives developed later in the paper will allow a fair amount of pre-computation.

2.1. Confusion matrix

The phone set is $\mathcal{P} = \{\mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3, ...\}$. The phone confusion is represented as $P(\mathrm{p}_i|\mathrm{p}_j)$ which is the probability that p_i is the true phone when p_j is observed. It may be necessary in some cases to map from one phone set into another. For example if the index was built using a particular set, but new data is from a recognizer that outputs a different phone set. In this case the recognizer's phone set should be mapped to that of the index. This also implies that the phone set chosen for the index should be as general as possible. The distributions $P(\mathrm{p}_i|\mathrm{p}_j)$ are derived from confusion matrices. To estimate the parameters, first, held out data is decoded with a speaker independent ASR system (described in the experiments section) to produce a phone level alignment from the decoding. A forced alignment to the reference transcripts is also carried out. The two results are then used to compute the phone confusion matrix. While the audio to be searched in this paper consists of broadcast news data, it was observed that either broadcast news or telephony data could be used to estimate the confusion matrix without much effect on the results. Further, reasonable results were obtained with about 1 hour of data.

2.2. Sequence match

Let m = the hypothesis (phone) vector size and n = the reference or query (phone) vector size. The query vector is

$$\mathbf{q} = \{\mathrm{q}_0, \mathrm{q}_1, \mathrm{q}_2, \ldots, \mathrm{q}_{n-1}\} \tag{1}$$

and the hypothesis vector is

$$\mathbf{h} = \{\mathrm{h}_0, \mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_{m-1}\} \tag{2}$$

where each q_i and h_i are elements of $\mathcal{P}$. The following dynamic programming procedure, commonly used to compute the weighted edit distance, will produce a result that can be used as a similarity measure. M is an $m+1 \times n+1$ matrix holding the scores of the paths which is initialized by setting $M[0][0] = 0.0$. The initial sequence of deletions is handled by the first row of M where each query symbol is in turn deleted from the output hypothesis without affecting the hypothesis

$$M[0][i] = M[0][i-1] + f(P(0|\mathrm{q}_{i-1})), i = 1 \ldots n \tag{3}$$

The function $f()$ is included for the sake of generality, and is often taken to be $-log$. Here, it is taken to be the identity function. The initial sequence of insertions is handled by the first column of M where each hypothesis symbol is in turn inserted into the hypothesis without a matching query symbol

$$M[i][0] = M[i-1][0] + f(P(\mathrm{h}_{i-1}|0)) \tag{4}$$

then the following recursion is used to fill in the matrix:

$$\begin{aligned} M[i][j] \quad = \quad & max\{ && (5)\\ & M[i-1][j-1] + f(P(\mathrm{h}_{i-1}|\mathrm{q}_{j-1})), && (6)\\ & M[i][j-1] + f(P(0|\mathrm{q}_{j-1})), && (7)\\ & M[i-1][j] + f(P(\mathrm{h}_{i-1}|0))\} && (8) \end{aligned}$$

for $i = 1 \ldots m, j = 1 \ldots n$. After the dynamic programming algorithm is run, the score of the best match (path) is given by

$$\mathcal{M}(\mathbf{q}, \mathbf{h}) = M[m][n], \tag{9}$$

which is a measure of the weighted edit distance (or similarity).

2.3. Building an index

The format is an inverted index. Each audio segment is converted into a sequence of phones by first passing the data through a speech recognizer and then using the baseforms in the dictionary to expand the word based transcript into phones. The length of the document is the number of resulting phones and the position of each phone is the granularity of the index. Initially, an N is chosen as the length of the phone string in the index. For each document D_k, the N-gram at each position is extracted and associated with the document and position pair. At the end of the document, where full

N-grams are not available, the sub-N-grams are stored. The same N-gram can appear multiple times in the same document as well as in multiple documents. Thus the inverted index contains for each N-gram in the index, a list of document and position pairs specifying all of the locations at which it occurs. The structure of the index is:

- $\mathbf{h}_1 = \{\mathrm{p}_{1,1}, \mathrm{p}_{1,2}, \mathrm{p}_{1,3}, \mathrm{p}_{1,4}, \mathrm{p}_{1,5}\}$ at $(d_{1,1}, n_{1,1}), \ldots$
- $\mathbf{h}_2 = \{\mathrm{p}_{2,1}, \mathrm{p}_{2,2}, \mathrm{p}_{2,3}, \mathrm{p}_{2,4}, \mathrm{p}_{2,5}\}$ at $(d_{2,1}, n_{2,1}), \ldots$
- $\mathbf{h}_3 = \{\mathrm{p}_{3,1}, \mathrm{p}_{3,2}, \mathrm{p}_{3,3}, \mathrm{p}_{3,4}, \mathrm{p}_{3,5}\}$ at $(d_{3,1}, n_{3,1}), \ldots$
- etc.

The index [the N-gram part] $\mathcal{I}$ contains all of the unique N-grams that occurred in the database of documents. Note that **h** is used to indicate elements of the index, anticipating that these will be used as the hypotheses when conducting a search. The query will be the reference, or the exact sequence of phones that is sought.

2.3.1. Generating scores

The query, a list of words, is first expanded into phones by the use of an automatic baseform generator [7], yielding a vector of phones with some length n, $Q = \{\mathrm{q}_0, \mathrm{q}_1, \mathrm{q}_2, \ldots, \mathrm{q}_{n-1}\}$. This is necessary in order to handle OOV terms for which baseforms may not be readily available. The hypotheses are the parts of the documents represented in the index via the N-grams. To determine document scores for the query, it is first expanded into N-grams, extracted as above (resulting in a sequence of overlapping N-grams offset by 1 phone).

$$Q \mapsto \{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \ldots\}. \tag{10}$$

Given a document D and a query N-gram $\mathbf{q}$, the best matching index element corresponding to the document is

$$\mathbf{h}^*(D, \mathbf{q}) = \arg \max_{\mathbf{h} \in \mathcal{I}/D} \mathcal{M}(\mathbf{q}, \mathbf{h}).$$

The notation $\mathcal{I}/D$ is used to indicate the elements of $\mathcal{I}$ that occur within the document D. Let N_Q = the number of N-grams extracted from the query. The score for document D is

$$\frac{1}{N_Q} \sum_{\mathbf{q}_i \in Q} \mathcal{M}(\mathbf{q}_i, \mathbf{h}^*(D, \mathbf{q}_i)), \tag{11}$$

which is the average of the scores for each query N-gram against its best matching index entry in the document. An important characteristic of the search used in this paper is that it employs an *approximate* match on an N-gram phone index.

3. BASELINE EXPERIMENTAL RESULTS

3.1. Data

The datasets are chosen to highlight the differences in performance on in vocabulary and out of vocabulary queries as well as to give comparative results to those in the literature.

Hub4: This dataset is taken from the English Hub4 broadcast news Eval97 and Eval98 data [4]. It consists of about 6 hours of audio, segmented into 1484 audio documents which on average contained 165 phones. The queries are all of the words in the reference transcripts, minus the stop words, giving a total of 6116 single word queries. All but a few of the queries were in vocabulary and they averaged between 6 and 7 phones.

oov: This is 9 hours of data from the 1996 and 1997 Hub4 and TDT-4 Corpora. There are 4162 audio segments. The average number of phones per segment was approximately 96. A total of 185 single word out of vocabulary (OOV) queries were chosen, which on average had between 6 and 7 phones. Note that this dataset has no in-vocabulary queries with respect to the ASR systems used.

In both cases, each audio segment is considered to be a separate document that could be retrieved by the search. Each query is contained in approximately 4 segments (documents) on average. The ASR systems used were trained on roughly 430 hours of English broadcast news data. A quinphone acoustic model with approximately 6K context dependent states and 250K Gaussians was used. As indicated in the experimental results both speaker independent and speaker dependent systems were used. Also, the same recognition parameters, for pruning beam, etc., were used throughout.

The first set of experiments establish a baseline, with results comparable to those in the literature. Average precision (p) and recall (r) per query are reported (in %) in table 1. In [4] the same data as dataset *Hub4* was used, and the results obtained are comparable to those in table 1. Note that the techniques are quite different though, as [4] uses a fragment based language model during recognition and indexes paths in the resulting lattices in order to allow vocabulary independent search. Here, an approximate match on N-grams derived from the 1-best decoding is used. The recognition systems used here and in [4] differ in the nature of the language model, fragment vs. word. The base vocabularies, though, are nearly the same, comparing to that used in building the fragment based language model. In the experiments here, N = 5 is used. Values of N from 3 to 7 were experimented with and 5 resulted in the best consistent overall performance. To get an understanding of the performance when the queries are not in the ASR vocabulary, results are presented on dataset *oov*. As is evident, the performance is considerably worse than on dataset *Hub4*. This is because dataset *oov* contains *only* OOV queries. In the sequel, the performance improvements on in and out of vocabulary queries is tracked on datasets *Hub4* and *oov* respectively.

Table 1. *Baseline system performance.*

Dataset	ASR	*p*%	*r*%
Hub4	SAT	67.83	67.15
oov	SAT	30.83	31.09

3.1.1. Comparative performance vs. ASR system complexity

In the results above (shown in table 1), the ASR system used speaker adapted training (SAT). Here, a performance comparison based on reducing the complexity of the acoustic models is presented. Three configurations of the ASR system are considered. System 1 is an ML system. System 2 is ML + fMPE (feature-space minimum phone error) [8]. The results in tables 1 and 2 show that the technique is robust in that it works well regardless of the complexity of the ASR system, and yet reflects improvements in those systems. The effect of reducing language model (LM) complexity, for the fMPE system, is reported in table 3. The original was a 4-gram model. The performance degrades somewhat for 3 of the cases, but interestingly, it actually improves very slightly for dataset *oov* with the 3-gram LM.

Table 2. *Baseline approach for various system configurations.*

Dataset	ASR	*p*%	*r*%
Hub4	ML	65.21	64.52
Hub4	fMPE	67.50	66.71
oov	ML	20.79	22.44
oov	fMPE	28.80	29.94

Table 3. *Baseline approach for fMPE configuration with lower LM complexity.*

Dataset	LM	*p*%	*r*%
Hub4	3-gram	61.93	61.55
Hub4	2-gram	59.84	58.81
oov	3-gram	29.34	29.90
oov	2-gram	27.19	28.15

4. CONSTRAINED MATCH WITH HIGHER ORDER CONFUSION

In this section, the baseline approach is expanded in light of observations made on the properties of data alignment error. In particular, when the decoded word strings are expanded into phone strings, the types of edit mistakes that can occur are constrained. Thus it is possible to be more efficient and accurate by taking this into consideration in the similarity measure.

Consider again the hypothesis and query vectors $\mathbf{h}$ and $\mathbf{q}$ given in equations 2 and 1. Whereas $\mathcal{M}(\mathbf{q}, \mathbf{h})$ has been used to indicate the level of match between query and hypothesis, a better measure could perhaps try to approximate $P(\mathbf{h}|\mathbf{q})$ or $P(\mathbf{q}|\mathbf{h})$. In this paper, these are referred to as high order confusions.

Because of the fact that the phone based transcriptions are derived from the 1-best decoded word sequences, it seems reasonable not to consider random edits when comparing the hypothesis sequences and the query sequence. Toward this end, the approach described here will not use the weighted edit distance-like measure as described in equations 3 through 8. Deletions and insertions will not be allowed when matching the N-gram subsequences and thus the measure is constrained by how well the N-grams align. We refer to this procedure as constrained match.

Table 4. *Results for constrained match.*

Dataset	ASR	*p*%	*r*%
Hub4	fMPE	66.68	78.30
Hub4	SAT	72.85	85.39
oov	SAT	32.42	34.66

Carrying the reasoning further, it makes sense that the standard phone confusion matrix should be updated to include, phone bi-gram, tri-gram, and in general N-gram confusions. That is, since errors result essentially from decoding errors, the rate at which higher order sequences are substituted should help in determining the final score of how well a document matches a query.

However, it can be problematic to estimate the general N-gram confusion rates because most will likely not be seen in reasonable amounts of training data. But the substitutions that do occur are used to estimate a contribution to the match score. Based on each confusion that is seen in the training data, a normalized matrix is created. The method is similar to that used in estimating the parameters of the single phone confusions, but here higher order confusions are considered. Since this matrix could have very high dimensionality, we store only the non-zero components. We have used approximately 10 hours of held out telephony and broadcast news data to estimate the high order confusion parameters in the experiments.

The following is proposed as a measure that captures the behavior of $P(\mathbf{h}|\mathbf{q})$ and yet is applicable when training data for the higher order confusions is limited. Let G_{max} be the highest G for which there are estimates of the G-gram confusions in the system (G is used here so as not to be confused with the N used for the size of the index entries). Here G_{max} will be 3 since up to 3-gram confusions are considered. Each index entry $\mathbf{h}_i$ and each N-gram in the query $\mathbf{q}_j$ is further parsed into overlapping G-grams shifted by 1 phone, denoted for example by $G_1(\mathbf{h}_i)$ for 1-grams of entry $\mathbf{h}_i$ or $G_3(\mathbf{q}_j)$ for

3-grams of query N-gram $\mathbf{q}_j$. These sets are ordered such that n^{th} element is the n^{th} occurring G-gram from left to right. As a notational convenience $G_3(\mathbf{q}_j)[n]$ will be used to indicated the n^{th} element below. Since N is the length of the sequences to match, there will be N 1-grams, $N-1$ 2-grams, and $N-2$ 3-grams.

$$\begin{aligned}\mathcal{S}(\mathbf{q}_i, \mathbf{h}_j) &= \alpha_1 \sum_{n=1}^{N} f(P(G_1(\mathbf{h}_i)[n]|G_1(\mathbf{q}_i)[n])) \\ &+ \alpha_2 \sum_{n=1}^{N-1} f(P(G_2(\mathbf{h}_i)[n]|G_2(\mathbf{q}_i)[n])) \\ &+ \alpha_3 \sum_{n=1}^{N-2} f(P(G_3(\mathbf{h}_i)[n]|G_3(\mathbf{q}_i)[n])) \quad (12)\end{aligned}$$

Again, $f()$ is the identity function. The parameters α_i need to be specified. Here, the values of $\{1, 1/3, 1/3\}$ were used. These values were not based on any training, but chosen *a priori*. The new document score is computed as follows. Given again a document D and a query N-gram $\mathbf{q}$, the best matching index element corresponding to the document is now given by

$$\mathbf{h}^*(D, \mathbf{q}) = \arg \max_{\mathbf{h} \in \mathcal{I}/D} \mathcal{S}(\mathbf{q}, \mathbf{h}).$$

Letting N_Q = the number of N-grams extracted from the query as before, the new score is

$$\frac{1}{N_Q} \sum_{\mathbf{q}_i \in Q} \mathcal{S}(\mathbf{q}_i, \mathbf{h}^*(D, \mathbf{q}_i)). \quad (13)$$

The technique can be efficient since most of the information can be pre-computed and it no longer uses dynamic programming (edit distance). The pre-computation could, for example, cache the scores for the most commonly occurring set of N-grams. Then when a query is expanded, if all the resulting N-grams are in the chosen set, scoring is essentially a matter of lookup. At this point many techniques could additionally be used for further efficiency. As an example, the index could be arranged hierarchically to allow an efficient search. Prefiltering mechanisms could also be used to reduce the set of index elements that need to be considered. An approach using vector space modelling is described in [9].

In table 4 the results are presented for the case where the edit distance is replaced by the constrained match, as in equation 13, with α_1 = 1 and the other α_i = 0 (here, we ran the fMPE configuration only for the *Hub4* dataset). Comparing tables 1 and 2 to table 4, the effects of this change are seen to be quite dramatic. In particular, both recall and precision are improved. Consider again that only the set of documents with the top score can be returned. We hypothesize that with the baseline measure, false alarms sometimes scored higher than correct hits, preventing the correct ones from being returned. Next, we evaluate the contribution of adding higher order confusions, with the results given in table 5. Again, we note that both recall and precision are improved. Moreover, the improvements are seen for both in vocabulary and out of vocabulary queries. It is expected that if the α_i were trained on held out data, the performance improvement would be even greater.

Table 5. *Higher order confusions (all info).*

Dataset	ASR	*p*%	*r*%
Hub4	SAT	74.25	86.43
oov	SAT	34.16	35.41

5. CONCLUSIONS

We have shown that in realistic scenarios where the speed of data acquisition must be balanced with the accuracy of subsequent information retrieval, the ability to efficiently incorporate higher order confusions, in addition to the traditional unigram confusions, was able to achieve gains in performance. The method was introduced in the context of a phone N-gram based indexing and approximate search scheme. The initial baseline approach used a variation on the weighted edit distance as a measure of phone string similarity. Results showed that this approach was comparable to others in the literature for the same data. Moreover, the results were fairly stable across a variety of ASR systems, which is important due to variations in how the data to be indexed may be collected. Then, the approach was modified and extended to use alignment and include higher order phone sequence confusions to reflect ASR errors more accurately. The new approach is more efficient and resulted in substantial gains. Moreover, two different data sets were studied to differentiate performance based on in vocabulary and out of vocabulary queries. Improvements were seen for both the in and out of vocabulary queries, though the improvements were greater for the in vocabulary case.

6. ACKNOWLEDGMENTS

The authors would like to acknowledge Brian Kingsbury for his assistance with the ASR systems and Olivier Siohan for discussions on path-based graph indexing and fragment language models. This work was partially supported by the Defense Advanced Research Projects Agency under contract No. HR0011-06-2-0001. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the U.S. government and no official endorsement should be inferred.

7. REFERENCES

[1] K. Ng, "Subword-based approaches for spoken document retrieval," in *Ph.D. thesis, MIT*, February 2000.

[2] A. Amir, A. Efrat, and S. Srinivasan, "Advances in phonetic word spotting," in *CIKM*, 2001.

[3] B. Logan, P. Moreno, and JM V. Thong, "Approaches to reduce the effects of OOV queries on indexed spoken audio," in *HPL-2003-46, Cambridge Research Laboratory, HP Laboratories Cambridge*, March 2003.

[4] O. Siohan and M. Bacchiani, "Fast vocabulary-independent audio search using path-based graph indexing," in *INTERSPEECH*, September 2005.

[5] J. Mamou, D. Carmel, and R. Hoory, "Spoken document retrieval from call-center conversations," in *SIGIR, Seattle*, 2006.

[6] T. Hori, I. L. Hetherington, T. Hazen, and J. R. Glass, "Open-vocabulary spoken utterance retrieval using confusion networks," in *ICASSP*, 2007.

[7] Stanley F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in *Proceedings of Eurospeech*, 2003.

[8] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *ICASSP, Philadelphia*, 2005.

[9] B. Matthews, U. Chaudhari, and B. Ramabhadran, "Fast audio search using vector space modelling," in *Submitted to ASRU*, 2007.