FAST UNCONSTRAINED AUDIO SEARCH IN NUMEROUS HUMAN LANGUAGES

J. Scott Olsson

University of Maryland
Dept. of Mathematics
College Park, MD
olsson@math.umd.edu

Jonathan Wintrode, Matthew Lee

U.S. Department of Defense
Ft. George G. Meade, MD 20755-6514
{jcwintr,melee}@ncsc.mil

ABSTRACT

We present a system to index and search conversational speech using a scoring heuristic on the expected posterior counts of phone n-grams in recognition lattices. We report significant improvements in retrieval effectiveness on five human languages over a strong 1-best baseline. The method is shown to improve the utility (mean average precision) of the retrieved lattices' rank order and to do so with a search cost negligible compared to the fastest yet known methods for the linear scanning of phonetic lattices.

Index Terms— Information retrieval, Speech processing, Speech recognition, Natural languages, Natural language interfaces

1. INTRODUCTION

Over the past decade, significant progress has been made towards systems capable of indexing and searching large volumes of spoken communications[1]. This area of interest is generally referred to as spoken document retrieval (SDR). A SDR system enables a user to enter a natural-language or word-based query and to retrieve spoken documents (files, utterances, etc.) containing those terms.

Many current systems combine an automatic speech recognition (ASR) process, which decodes speech into word-based text, with a text-based information retrieval system. While these systems perform reasonably well when applied to data with low ASR word error rates and medium sized vocabularies (e.g., broadcast news), alternative methods are often necessary. This is particularly so in conversational speech, in which irregular prosody and out of vocabulary (OOV) terms are prevalent. For information seekers, it is precisely because these OOV terms (e.g., names of people and places) are *rare* that they are *informative*—and so special care must be taken to support their detection. This paper focuses on detecting these rare terms—vocabulary independent audio search.

Research in vocabulary independent SDR has largely focused on subword indexing methods [2] and efficiently searching more complex representations of the recognition hypotheses, such as phonetic lattices [3, 4]. Because a linear search through many lattices is still costly, two-stage search systems have also been considered. Two-stage search uses a fast, high recall, low precision filtering system to produce a candidate set of lattices for further scanning. In [5], discriminating fragments of phone sequences are indexed to produce an unordered set of these candidates. Our work is similar in that we use an inverted index on lattice features (in our case, expected phone n-gram counts) to retrieve the segments. Our approach differs in its choice of indexing unit, its focus on very fast search (we do not allow for a costly second stage) and in that we produce an *ordered* list of lattices. We focus on improving the utility (i.e., the rank order) of the lattices in one stage.

Our results are also significant in the breadth of languages we examine. We report experiments in English, Spanish, Mandarin Chinese, Persian Farsi, and Levantine Arabic conversational speech. The methods developed are universally applicable, and have thusfar been extended to handle many of the world's languages. This motivates our emphasis on approaches such as phonetic lattice indexing which do not require the impracticable costs of training large vocabulary continuous speech recognizers on resource poor languages.

2. LATTICE GENERATION

Before we can index or search our spoken documents, we run phonetic recognition to produce a compact set of hypotheses—a phonetic lattice—for the phone sequences observed in the audio. The acoustic models used to produce these lattices are created using HTK's embedded training functionality. For the experiments presented in this work, new models are trained for each language. Prior to training, word-based transcriptions are converted to phonemes using a rule-based transliterator (RBT) [6]. The HMM models are then trained as left-context dependent phones and have three states, each with 17 Gaussian mixtures.

Phonetic lattices are produced using an alternative Viterbi implementation called the Token Passing Model [7] contained within HTK. In this formulation, a token is passed from state to state which contains the log probability of the current path as well as a record of the previous states or models already

visited. At each state, a copy of the present token is passed to each of the connecting states. When a state is passed multiple tokens, they are each examined and only the token with the highest probability is retained. For lattice generation, multiple tokens can be saved at each state in order to retain multiple hypotheses. Consequently, as the process is directed to retain more tokens at each state, a larger number of hypotheses are recorded, resulting in a deeper lattice. Increasing the number of tokens, however, incurs a higher computational cost, both during decoding and, later, indexing.

Many other parameters of the lattice generation process can also significantly impact the system's balance of performance and speed. Two of the most relevant control parameters are the insertion penalty and language model scale. It was discovered empirically that the parameter combination that maximizes phone accuracy does not consequently maximize search performance. In fact, the parameters that maximize retrieval accuracy (measured as mean average precision), tend to produce a phone lattice with a moderately higher level of insertion errors. For this work, we biased the results against our new approach and chose parameters which roughly maximized mean average precision on our baseline search system.

3. LATTICE INDEXING

After mapping an audio file to its phonetic lattice representation, we must further transform it to facilitate efficient search. Specifically, we wish to extract a set of features which can be stored and retrieved quickly (e.g., in an inverted index) and which succinctly represent the phonetic information observed in the audio.

Given a lattice L containing many paths ℓ (i.e., $\ell \in L$), the expected number of occurrences for phone n-gram X over all paths is

$$E_{P_L}[C(X)] = \sum_{\ell \in L} P_L(\ell)\, C_\ell(X).$$

Here, $C_\ell(X)$ denotes the number of times phone n-gram X occurs in lattice path ℓ. The posterior distribution $P_L(\ell)$ is defined as

$$P_L(\ell) = \frac{\exp\{\sum_{\alpha \in \ell} S(\alpha)\}}{\sum_{\nu \in L} \exp\{\sum_{\beta \in \nu} S(\beta)\}},$$

where $\exp\{\cdot\}$ denotes exponentiation, as we assume the score $S(\alpha)$ for an arc α on the path is a log probability (perhaps simply the sum of the acoustic and language model log probabilities). In practice, these values may be efficiently computed using a variant of the forward-backward algorithm. This functionality is currently supported by the SRI language modeling toolkit [8].

For each lattice in the corpus, we compute the expected phone n-gram counts for $n \leq N$. Phone n-grams having expected count less than τ are discarded. These n-grams and their associated counts are then indexed using a straightforward inverted index. While larger $\tau's$ decrease the total index size, we found the default choice of $\tau = 1 \times 10^{-4}$ to produce manageable indices and excellent results. A preliminary study did show mean average precision monotonically decreasing for increasing τ. We fix N at $N = 5$.

4. SEARCHING

A rule-based transliterator (RBT) is first used to map query words into their phonetic components [6]. This mapping may use both context sensitive rules and, when available, pronunciation dictionaries.

After mapping the query into it's phone sequence, we extract a set Q of phone subsequences with length n, $N - \Delta \leq n \leq N$. The integer Δ simply parameterizes the smallest indexing unit used for the search. Note, if the full query's phone sequence is smaller than N, the length of the sequence is used as the largest unit for search. Naturally, indexing sequences are also constrained to have a positive length. For example, if we are searching for *goodness* with $N = 5$ and $\Delta = 1$, we first apply the RBT

$$\textit{goodness} \xrightarrow{\text{RBT}} [\text{g ʊ d n ɪ s}],$$

and then extract the phone subsequence set

$$Q = \{\text{g ʊ d n}, \text{ʊ d n ɪ}, \text{d n ɪ s}, \text{g ʊ d n ɪ}, \text{ʊ d n ɪ s}\}.$$

If instead we are searching for *fun* (with the same N and Δ), we extract the subsequence set

$$Q = \{\text{f ʌ}, \text{ʌ n}, \text{f ʌ n}\}.$$

We roughly expect a larger value of Δ to improve retrieval when phone recognition is very poor (i.e., when our query phone subsequences will not have accurately indexed counts for $n = N$). On the other hand, if Δ is too large, the very short phone subsequences utilized will be only poor discriminators for the underlying terms (e.g., lattices not containing *goodness* may nevertheless include the phone 1-gram ʊ).

To compute the score for a query and lattice L, we sum the posterior expected n-gram counts associated with each element of Q,

$$score(query, L) = \sum_{q \in Q} \log\{E_{P_L}[C(q)]\}. \qquad (1)$$

The logarithm can be thought of as a damping function which, due to it's singularity at $\log\{0\}$, acts to aggressively penalize lattices having a near zero count for some phone subsequence. We smooth the counts by giving absent subsequences a very small count $\epsilon \ll 1$, which parameterize the penalty for a missing n-gram. We found our results to be rather insensitive to choice of ϵ, which we set to $\epsilon = 1 \times 10^{-15}$. Note, because Q contains more subsequences of shorter lengths and

because every short subsequence is itself a portion of a longer sequence, lattices are most severely penalized if they do not contain the short elements of Q. This conforms to our intuition that lattices lacking even the shortest phone subsequences in Q are unlikely to correspond to our query term.

One advantage of this search heuristic is that Equation 1 may be computed very efficiently. Queries having a phone sequence of length m require only $|Q| = \sum_{n=N-\Delta}^{N} m - n + 1$ lookups in the inverted index—a marginal cost in comparison to the fastest known methods for the scanning of lattices. At the same time, because $score(query, L)$ is proportional to the probability of a term occurring in the audio, it naturally provides a suitable ranking function for the lattices [9].

5. EXPERIMENTS

To evaluate the search performance of this approach, test data in the form of audio and transcripts was assembled in the following languages: English, Spanish, Mandarin, Levantine, and Persian. Each test set was drawn from a corpora of conversational telephone speech and was excluded from data used to train the associated phonetic recognizer. All data used for training and testing are publicly available and were obtained through the Linguistic Data Consortium[1]. Table 1 details the source and size of each of the test sets used in the experiments presented here. A list of query words was generated from the actual transcripts for the evaluation audio in each language. With the exception of Mandarin, all words containing 3 or more characters were included in the query list. For Mandarin, as single characters correspond to whole words, all words were included. We have maintained the same evaluation sets from [10] to establish a strong baseline for our measurements.

As in [10], we report mean average precision (MAP). Precision p is simply the proportion of documents retrieved (at a point in a ranked list) which are relevant. To compute MAP, the precisions at each relevant document in a query's ranked list are averaged. The mean of these averages is then computed over the set of all queries. For an average query's ranked list, MAP gives the expected precision at a relevant document, and so is a measure of the utility of a search system for a user. Formally, if $p(r)$ denotes the precision at cut-off rank r for a system returning D documents with R total relevant documents, and $rel(r)$ is a binary function indicating the relevance of a given rank, then the average precision $\hat{p}$ for a query is

$$\hat{p} = \frac{\sum_{r=1}^{D} p(r)rel(r)}{R}.$$

The MAP is then simply the average of $\hat{p}$ over all queries.

For the lattice search results that follow, we used 5 tokens during the HTK decoding process to produce the lattices, indexed n-grams of up to length 5, and searched with n-grams of length 3, 4, and 5 using parameter values $N = 5$ and $\Delta = 2$. We do not claim any optimality in these parameters and, although they yield encouraging results, they certainly merit further examination.

[1] http://www.ldc.upenn.edu/

	3 Tokens		5 Tokens	
Language	Speed (xRT)	MAP	Speed (xRT)	MAP
English	0.36x	29.6	0.89x	**30.5**
Spanish	0.36x	24.5	1.22x	**25.4**

Table 2. Comparison of indexing speed and search performance (in MAP) using 3-token and 5-token settings for lattice generation.

The baseline 1-best search approach seeks to optimize the search of errorful 1-best output using a weighted match with the query terms. The phones of the query are matched with the phones of the 1-best output by a dynamic programming minimum edit distance calculation that uses weights for phone substitutions, insertions, and deletions from a set of language-specific confusion matrices. The confusion matrix for a language seeks to represent a mapping from the phonetic space of reference transcripts (transliterated by the same mechanism used for queries) to the phonetic space of the recognizer output. This mapping encapsulates recognition error, incorrect transcription error, and pronunciation variation not captured by the query transliterator.

5.1. Results

Our experiments with the proposed indexing and search strategy demonstrate a significant improvement in MAP performance over our baseline 1-best results. For comparison, Table 3 includes both the results from [10] and the results of a 1-best search using an updated version of the phonetic recognizer used for the lattice search. The improvements in the most recent phone recognizer are primarily due to additional training data and various algorithmic refinements including: the RBT, the forced alignment procedure, language model training, and parameter optimization (as mentioned in Section 2).

Table 3 shows that the updated 1-best recognizer produces significant improvements in terms of both phonetic accuracy and mean average precision. However, substantial additional gains are achieved by employing the proposed lattice-based approach. Improvements over the newest 1-best technique were measured for each of the five languages tested. The relative improvement in MAP ranged from 15% for Mandarin Chinese to 106% for Persian Farsi.

As mentioned previously, 5 tokens were used for the lattice generation process. While this resulted in our largest observed gains in retrieval performance, the computational cost of maintaining 5 tokens per state during Viterbi decoding is substantial. Table 2, however, shows that the computational cost of lattice generation can be significantly mitigated by re-

Language	Source	Duration (minutes)	Utterances	Unique Query Words
English	CallHome	148.08	2175	2204
Spanish	CallHome	225.8	3908	3817
Mandarin	CallHome	83.2	2888	1845
Levantine	EARS	649.8	6648	9890
Persian	CallFriend	239.7	5903	4777

Table 1. Source and collection statistics for search evaluation data sets.

	1-best (2005)		1-best (2006)		Lattice
Language	Acc.	MAP	Acc.	MAP	MAP
English	38.1	22.5	43.4	23.2	**30.5**
Spanish	48.4	15	50.9	21.5	**25.4**
Mandarin	37.5	5.7	42.2	9.8	**11.3**
Levantine	-	6.9	37.2	8.1	**12.3**
Persian	30.5	4.0	43.5	5.3	**10.9**

Table 3. Phone accuracy and search effectiveness for five languages using the original 1-best recognized phonetic output (2005), the 1-best output from the updated phone recognizer (2006), and the proposed lattice-based system.

ducing the number of tokens to 3. For both English and Spanish, reducing the number of tokens resulted in increases in decoding speeds by factors of 3 and 2, respectively, while suffering only minor losses in MAP. It should be noted that all experiments were conducted on a Linux machine with an AMD 2.0 GHz processor.

One strength of our baseline system is that it accounts for the confusability on phones by estimates directly measured on held out data. The lattice method, on the other hand, does not incorporate any notion of nearness in mismatched phones, so that we might expect it to perform worse on languages with low recognition accuracy. Table 3 demonstrates that this is not the case. The lattice indexing approach is surprisingly robust in the presence of recognition error, presumably because a sufficient number of alternative phone hypotheses are represented by the lattices. As Table 2 indicates, this remains true even with many fewer paths in the lattice (i.e., for fewer tokens).

6. CONCLUSION

We have proposed a lattice indexing and search procedure for spoken utterance retrieval of conversational speech. The method efficiently indexes and searches phonetic lattices, showing significant improvements in retrieval performance over our baseline, while maintaining a system that is faster than real-time. By demonstrating significant performance increases across five languages, we have shown the method to be surprisingly robust both to variation in human language and the error characteristics of phonetic recognition systems.

While our purpose was to maximize the utility of the retrieved lattices as quickly as possible, future work might extend this approach by reranking the returned list with a more costly lattice scanning system [3]. We also hope to explore larger collection sizes. For very large collections, an index may potentially grow to become unmanageable. To constrain its size, we may consider the careful selection of indexing features by observable phonotactic constraints. We hope we have provided a sound basis for this work.

7. REFERENCES

[1] J. Garofolo, G. Auzanne, and E. Voorhees, "The TREC spoken document retrieval task: A success story," Proceedings of the TREC-9 Conference, 2000.

[2] K. Ng and V.W. Zue, "Subword-based approaches for spoken document retrieval," *Speech Commun.*, vol. 32, no. 3, pp. 157–186, 2000.

[3] J.T. Foote, S.J. Young, G.J.F. Jones, and K.S. Jones, "Unconstrained keyword spotting using phone lattices with application to spoken document retrieval," *Computer Speech and Language*, vol. 11, pp. 207—224, 1997.

[4] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *HLT-NAACL*, 2004.

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

[6] P. Schone, "Low-Resource Autodiacritization of Abjads for Speech Keyword Search," INTERSPEECH, 2006.

[7] S.J. Young, N.H. Russel, and J.H.S. Thornton, "Token passing: A simple conceptual model for connected speech recognition systems," Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Dept., 1989.

[8] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Intl. Conf. on Spoken Language Processing*, 2002.

[9] S.E. Robertson, "The Probability Ranking Principle in IR," *Journal of Documentation*, pp. 281–286, 1977.

[10] P. Schone, P. McNamee, G. Morris, G. Ciany, and S. Lewis, "Searching conversational telephone speech in any of the world's languages," IA-2005, McLean, VA.