Hindawi Publishing Corporation
EURASIP Journal on Audio, Speech, and Music Processing
Volume 2008, Article ID 568737, 8 pages
doi:10.1155/2008/568737

Research Article

Voice-to-Phoneme Conversion Algorithms for Voice-Tag Applications in Embedded Platforms

Yan Ming Cheng, Changxue Ma, and Lynette Melnar

Human Interaction Research, Motorola Labs, 1925 Algonquin Road, Schaumburg, IL 60196, USA

Correspondence should be addressed to Lynette Melnar, melnar@labs.mot.com

Received 28 November 2006; Revised 15 July 2007; Accepted 26 September 2007

Recommended by Joe Picone

We describe two voice-to-phoneme conversion algorithms for speaker-independent voice-tag creation specifically targeted at applications on embedded platforms. These algorithms (*batch mode* and *sequential*) are compared in speech recognition experiments where they are first applied in a same-language context in which both acoustic model training and voice-tag creation and application are performed on the same language. Then, their performance is tested in a cross-language setting where the acoustic models are trained on a particular source language while the voice-tags are created and applied on a different target language. In the same-language environment, both algorithms either perform comparably to or significantly better than the baseline where utterances are manually transcribed by a phonetician. In the cross-language context, the voice-tag performances vary depending on the source-target language pair, with the variation reflecting predicted phonological similarity between the source and target languages. Among the most similar languages, performance nears that of the native-trained models and surpasses the native reference baseline.

Copyright © 2008 Yan Ming Cheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

A voice-tag (or name-tag) application converts human speech utterances into an abstract representation which is then utilized to recognize (or classify) speech in subsequent uses. The voice-tag application is the first widely deployed speech recognition application that uses technologies like Dynamic Time Warping (DTW) or HMMs in embedded platforms such as in mobile devices.

Traditionally, HMMs are directly used as abstract speech representations in voice-tag applications. This approach has enjoyed considerable success for several reasons. First, the approach is language-independent so it is not restricted to any particular language. With the globalization of mobile devices, this feature is imperative as it allows for speaker-dependent speech recognition for potentially any language or dialect. Second, the HMM-based voice-tag technology achieves high speech recognition accuracy while maintaining a low CPU requirement. The storage of one HMM per voice-tag, however, is rather significant for many embedded systems, especially for low-tier ones. Only as long as storage is kept under the memory budget of an embedded system by limiting the number of voice-tags, is the HMM-based voice-tag strategy acceptable. Usually, two or three dozen voice-tags are recommended for low-tier embedded systems, while high-tier embedded systems can support a greater number. Nevertheless, interest in constraining the overall cost of embedded platforms limits the number of voice-tags in practice. Finally, the HMM-based voice-tag has been successful because it is speaker-dependent and convenient for the user to create during the enrollment phase. A typical enrollment session in a speaker-dependent context requires only a few sample utterances to train a voice-tag HMM that captures both the speech abstraction and the speaker characteristics.

Today, speaker-independent and phoneme HMM-based speech recognizers are being included in mobile devices, and voice-tag technologies are mature enough to leverage the existing computational resources and algorithms from the speaker-independent speech recognizer for further efficiency. A name dialing application can recognize thousands of names downloaded from a phonebook via grapheme-to-phoneme conversion, and voice-tag technology is a convenient way of dynamically extending the voice-enabled

phonebook. In this type of application, voice-tag entries and phonetically transcribed name entries are jointly used in a speaker-independent context. Limiting voice-tags to two or three dozen in this scenario may no longer be practical, though extending the number significantly in a traditional HMM-based voice-tag application could easily surpass the maximum memory consumption threshold for low-tier embedded platforms. Given this, the speaker-dependency of the traditional approach may actually prevent the combined use of the voice-tag HMM technology and phonetically transcribed name entries.

Assuming that a set of speaker-independent HMMs already resides in an embedded platform, it is natural to think of utilizing a phonetic representation (phoneme strings or lattices) created from sample utterances as the abstract representation of a voice-tag, that is, voice-to-phoneme conversion. The phonetic representation of a voice-tag can be stored as cheaply as storing a name entry with a phonetic transcription, and it can be readily used in conjunction with the name entries obtained via grapheme-to-phoneme conversion, as long as such a voice-tag is speaker-independent. Voice-to-phoneme conversion, then, enhances speech recognition capability in embedded platforms by greatly extending recognition coverage.

As mentioned, a practical concern of voice-tag technology is user acceptability. Extending recognition coverage from dozens to hundreds of entries without maintaining or improving recognition and user convenience is not a viable approach. User acceptability mandates that the number of sample utterances required to create a voice-tag during enrollment be minimal, with one sample utterance being most favorable. However, in a speech recognition application with a large number of voice-tags, the recognition accuracy of each voice-tag tends to increase as its number of associated sample utterances increases. So, in order to achieve acceptable performance, more than one sample utterance is typically required during enrollment. Generally, a compromise of two to three sample utterances is usually considered acceptable.

Voice-to-phoneme conversion has been investigated in modeling pronunciation variations for speech recognition ([1, 2]), spoken document retrieval ([3, 4]) and word spotting ([5]) with noteworthy success. However, optimal conversion in the sense of maximum likelihood presented in these prior works requires prohibitively high computation, which prevents their direct deployment to an embedded platform. This crucial problem was resolved in [6], where we introduced our batch mode and sequential voice-to-phoneme conversion algorithms for speaker-independent voice-tag creation. In Section 2 we describe the batch mode voice-to-phoneme conversion algorithm in particular and show how it meets the criteria of low computational complexity and memory consumption for a voice-tag application in embedded platforms. In Section 3 we review the sequential voice-to-phoneme algorithm which both addresses user convenience during voice-tag enrollment and improves recognition accuracy. In Section 4, we discuss the experiment conditions and results for both the batch mode and sequential algorithms in a same-language context.

Finally, a legitimate concern confronting any voice-tag approach is its language extensibility. Increasingly, the task of extending a technology to a new language must consider the potential lack of sufficient target-language resources on which to train the acoustic models. An obvious strategy is to use language resources from a resource-sufficient source language to recognize a target language for which little or no speech data is assumed. Several studies have in fact explored the effectiveness of the cross-language application of phoneme acoustic models in speaker-independent speech recognition (see [7–10]). In [11], we demonstrated the cross-language effectiveness of the batch mode and sequential voice-to-phoneme conversion algorithms. Section 5 documents the cross-language voice-tag experiments and provides the results in comparison with that of those achieved in the same-language context. Here, we analyze the cross-language results in terms of predicted global phonological distance between the source and target languages. Finally, we share some concluding remarks in Section 6.

2. BATCH-MODE VOICE-TO-PHONEME CONVERSION

The principle idea of batch mode creation is to use a feature-based combination (here, DTW) collapsing M sample utterances (hereinafter *samples*) into a single "average" utterance. The expectation is that this "average" utterance will preserve what is common in all of the constituent samples while neutralizing their peculiarities. As mentioned in the previous section, the number of enrollment samples directly affects voice-tag accuracy performance in speech recognition. The greater the number of samples during the enrollment phase, the better the performance is expected to be.

Let us consider that there are M samples, $\mathbf{X_m}$ ($\mathbf{m} \in [\mathbf{1}, \mathbf{M}]$), available to a voice-tag in batch mode. $\mathbf{X_m}$ is a sequence of feature vectors corresponding to a single sample. For the purpose of this discussion, we do not distinguish between a sequence of feature vectors and a sample utterance in the remaining part of this paper. The objective here is to find the N-best phonetic strings, $\mathbf{P_n}$ ($\mathbf{n} \in [\mathbf{1}, \mathbf{N}]$), following an optimization criterion.

In prior works describing a batch mode voice-to-phoneme conversion method, the tree-trellis N-best search algorithm [12] is applied to find the optimal phonetic strings in the maximum likelihood sense [1, 2]. In [1, 2], the tree-trellis algorithm is modified to include a backward, time asynchronous, tree search. First, this modified tree-trellis search algorithm produces a tree of partial phonetic hypotheses for each sample using conventional time-synchronized Viterbi decoding and a phoneme-loop grammar in the forward direction. The M trees of phonetic hypotheses of the samples are used jointly to estimate the admissible partial likelihoods from each node in the grammar to the start node of the grammar. Then, utilizing the admissible likelihood of partial hypotheses, an A^*-search in the backward direction is used to retrieve the N best phonetic hypotheses, which maximize the likelihood. The modified tree-trellis algorithm generally falls into the probability combination algorithm category. Because this algorithm requires storing M trees of partial hypotheses simultaneously, with each tree being

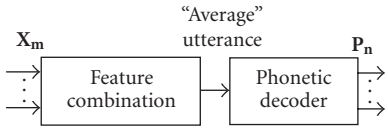

Figure 1: Voice-to-phoneme conversion in batch mode.

rather large in storage, it is expensive in terms of memory consumption. Furthermore, the complexity of the A^* search increases significantly as the number of samples increases. Therefore, this probability combination algorithm is not very attractive for deployment in mobile devices.

To meet embedded platform requirements, we use a feature-based combination algorithm to perform voice-to-phoneme conversion to create a voice-tag. By combining the M sample utterances of different lengths into a single "average" utterance, a simple phonetic decoder, e.g., the original form of the tree-trellis search algorithm with a looped phoneme grammar, can be used to obtain N best phonetic strings per voice-tag with minimum memory and computational consumption. Figure 1 depicts the system.

DTW is of particular interest to us because of the memory and computation efficiency of its implementation in an embedded platform. Many embedded platforms have a DTW library specially tuned to the target hardware. Given two utterances, $\mathbf{X_i}$ and $\mathbf{X_j}$ ($\mathbf{i} \neq \mathbf{j}$ and $\mathbf{i}, \mathbf{j} \in [\mathbf{1}, \mathbf{M}]$), a trellis can be formed with $\mathbf{X_i}$ and $\mathbf{X_j}$ being horizontal and vertical axes, respectively. Using a Euclidean distance and DTW algorithm, the best path can be derived, where "best path" is defined as the lowest accumulative distance from the lower-left corner to the upper-right corner of the trellis. A new utterance $\mathbf{X_{i,j}}$ can be formed along the best path of the trellis, $\mathbf{X_{i,j}} = \mathbf{X_i} \oplus \mathbf{X_j}$, where $\oplus$ is denoted as the DTW operator. The length L of the new utterance is the length of the best path.

Let:

$$\mathbf{X_{i,j}} = \{x_{i,j}(0), \ldots, \mathbf{x_{i,j}(t)}, \ldots, \mathbf{x_{i,j}(L_{i,j} - 1)}\},$$
$$\mathbf{X_i} = \{\mathbf{x_i X_i} = \{\mathbf{x_i}(0), \ldots, \mathbf{x_i}(\varsigma), \ldots, \mathbf{x_i}(\mathbf{L_i} - 1)\} \text{and}$$
$$\mathbf{X_j} = \{\mathbf{x_j(0)}, \ldots, \mathbf{x_j}(\tau), \ldots, \mathbf{x_j}(\mathbf{L_j} - 1)\},$$

where $\mathbf{t}$, ς, τ are frame indices.

We define $\mathbf{x_{i,j}(t)} = (\mathbf{x_i}(\varsigma) + \mathbf{x_j}(\tau))/2$, where $\mathbf{t}$ is the position on the best path aligned to the ςth frame of $\mathbf{X_i}$ and the τth frame of $\mathbf{X_j}$ according to the DTW algorithm. Figure 2 sketches the feature combination of two samples.

Given M samples $\mathbf{X_1} \ldots \mathbf{X_M}$ and the feature combination algorithm of two utterances, there are many possible ways of producing the final "average" utterance. Through experimentation we have found that they all achieve statistically similar speech recognition performances. Therefore, we define the "average" utterance as the cumulative operation of the DTW-based feature combination:

$$\mathbf{X_{1,2,3,\ldots,M}} = (\cdots((\mathbf{X_1} \oplus \mathbf{X_2}) \oplus \mathbf{X_3}) \cdots \oplus \mathbf{X_M}). \quad (1)$$

The cumulative operation provides a storage advantage for the embedded system.Independent of M, only two utterances need to be stored at any instance: the intermediate

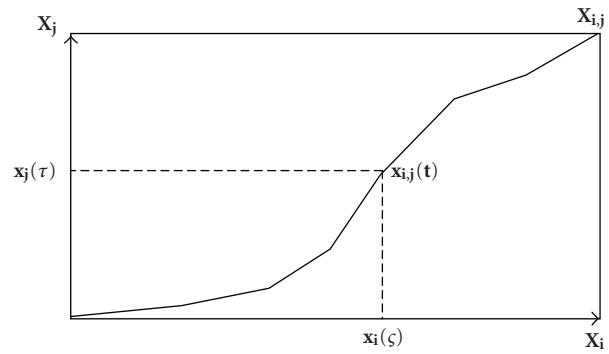

Figure 2: DTW-based feature combination of two sample utterances.

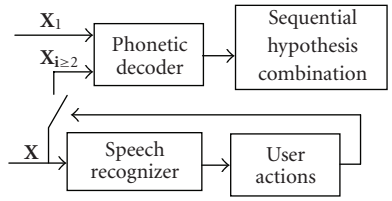

Figure 3: Sequential combination of hypothetical results of a phonetic decoder.

"average" utterance and the next new sample utterance. The computation complexity of the batch-mode voice-to-phoneme conversion is the **M**-1 DTW operation, one tree decoding with a phoneme-loop grammar and a trellis decoding for N-best phonemic strings. As a comparison, the approach in [1, 2] needs to store at least one sample utterance and **M** rather large phonemic trees; furthermore, it requires the computation of M tree decoding with the phoneme loop grammar and a trellis decoding. Thus, the proposed batch mode algorithm is more suited for an embedded system.

3. SEQUENTIAL VOICE-TAG CREATION

Sequential voice-tag creation is based on the hypothesis combination of the outputs of a phonetic decoder of **M** samples. In this approach, only one sample per voice-tag is required to create N initial seed phonetic strings, $\mathbf{P_n}$, using a phonetic decoder. The phonetic decoder used here is the same as described in the previous section. If good phonetic coverage is exhibited by the phonetic decoder (i.e., good phonetic robustness of the trained HMMs), with initial seed phonetic strings the recognition performance of voice-tags is usually acceptable, though not maximized. Each time a voice-tag is successfully utilized (a positive confirmation of the speech recognition result is detected and the corresponding action is implemented—e.g., the call is made), the utterance is reused as another sample to produce additional N phonetic strings to update the seed phonetic strings of the voice-tag through performing hypothesis combination. This update can be performed repeatedly until a maximum performance is reached. Figure 3 sketches this system.

The objective of this method is to discover a sequential hypothesis combination algorithm that leads to maximum performance. We use a hypothesis combination based on a consensus hierarchy displayed in the best phonetic strings of samples. The consensus hierarchy is expressed numerically in a phoneme n-gram histogram (typically a monogram or bigram is used).

3.1. Hierarchy of phonetic hypotheses

To introduce the notion of "hierarchy of phonetic hypotheses," let us begin by showing a few examples. Suppose we have three samples of the name "Austin." The manual phonetic transcription of this name is /= s t I n/. The following list shows the best string obtained by the phonetic decoder for each sample:

$$\begin{aligned} \mathbf{P}_1(\mathbf{X}_1): &\ \text{f pau ɔ s I n d}, \\ \mathbf{P}_1(\mathbf{X}_2): &\ \text{k pau ɔ f I n d}, \\ \mathbf{P}_1(\mathbf{X}_3): &\ \text{pau ɔ f I n}. \end{aligned} \tag{2}$$

The next list shows the three best phonetic strings of the first sample obtained by the phonetic decoder:

$$\begin{aligned} \mathbf{P}_1(\mathbf{X}_1): &\ \text{f pau ɔ s I n d}, \\ \mathbf{P}_2(\mathbf{X}_1): &\ \text{t pau ɔ s I n d}, \\ \mathbf{P}_3(\mathbf{X}_1): &\ \text{f pau ɔ s I n z}. \end{aligned} \tag{3}$$

Examining the results of the phonetic decoder, we observe that there are some phonemes that are very stable across the samples and the hypotheses; further, these phonemes tend to correspond to identical phonemes in the manual transcription. It is also observed that other phonemes are quite unstable across the samples. These derive from the peculiarities of each sample and the weak constraint of the phoneme-loop grammar. Similar observations are also made in [13], where stable phonemes in particular are termed the "consensus" of the phonetic string. Since we are interested in embedded voice-tag speech recognition applications in potentially diverse environments, we investigated phoneme stability in both favorable and unfavorable conditions. Our investigation shows that in a noisy environment some phonemes still remain stable while others become less stable compared to a more quiet environment. In general however, a hierarchical phonetic structure for each voice-tag can be easily detected, independent of the environment. At the top level of the hierarchy are the most stable phonemes that reflect the consensus of all instances of the voice-tag abstraction. The phonemes at the middle level of the hierarchy are less stable but are observed in the majority of voice-tag instances. The lowest level in the hierarchy includes the random phonemes, which must be either discarded or minimized in importance during voice-to-phoneme conversion.

3.2. Phoneme n-gram histogram-based sequential hypothesis selection

To describe the hierarchy of a voice-tag abstraction, we utilize a phoneme n-gram histogram. The high frequency phoneme n-grams correspond to "consensus" phonemes, the median frequency to "majority" phonemes and the low frequency to "random" phonemes of a voice-tag. In this approach, it is straightforward to estimate sequentially the n-gram histogram via a cumulative operation. e.g., one can use the well-known *relative entropy* measure [14] to compare two histograms. Another favorable attribute of this approach is that the n-gram histogram can be stored efficiently. For instance, given a phonetic string of length L, there are at most L monograms, $L+1$ bigrams or $L+2$ trigrams without counting zero frequency n-grams. In practice, the n-gram histogram of a voice-tag is estimated based only on the best phonetic strings of the previous utterances and the current utterance of the voice-tag. We ignore all but the best phonetic string of any utterance in the histogram estimation because the best phonetic string differs from the second best or third best by only one phoneme by definition. This "defined" difference may not be helpful in revealing the majority and random phonemes in a statistical manner; it may even skew the estimated histogram.

The sequential hypothesis combination algorithm is provided below:

Enrollment (or initialization): Use one sample per voice-tag to create N phonetic strings via a phonetic decoder as the current voice-tag; use the best phonetic string to create the phoneme n-gram histogram for the voice-tag.

Step 1. Given a new sample of a voice-tag, create N new phonetic strings (via the phonetic decoder); update the phoneme n-gram histogram of the voice-tag with the best phonetic string of the new sample.

Step 2. Estimate a phoneme n-gram histogram for each phonetic string for N current and N new phonetic strings of the voice-tag.

Step 3. Compare the phoneme n-gram histogram of the voice-tag with that of each phonetic string using a distance metric, such as *relative entropy* measure; select N phonetic strings, the histograms of which are closest to the histogram of the voice-tag histogram, as the updated voice-tag representation.

Step 4. Repeat steps 1–3 if a new sample is available.

4. SAME-LANGUAGE EXPERIMENTS

The database selected for this evaluation is a Motorola-internal American English name database which contains a mixture of both landline and wireless calls. The database consists of spoken proper names of variable length. These names are representational of a cross-section of the United States and predominantly include real-use names of European, South Asian, and East Asian origin. No effort was made to control the length of the average name utterance, and no bias was provided toward names with greater length. Most callers speak either Standard American English or

Northern Inland English, though there are a number of other English speech varieties represented as well, including foreign-accented English (especially Chinese and Indian language accents). The database is divided into voice-tag creation and evaluation sets where the creation set has 85 name entries corresponding to 85 voice-tags, and each name entry comprises three samples spoken by a single speaker in different sessions. Thus the creation set is speaker-dependent. The purpose of designing a speaker-dependent creation set is that we expect that any given voice-tag will be created by a single user in real applications and not by multiple users. The evaluation set contains 684 utterances of the 85 name entries. Most speakers of a name entry in the evaluation set are different from the speaker of the same name entry in the creation set, though, due to name data limitations, a few speakers are the same for both sets. In general, however, our evaluation set is speaker-independent.

We use the ETSI advanced front-end standard for distributed speech recognition [15]. This front end generates a feature vector of 39 dimensions per frame and the feature vector contains 12 MFCC plus energy and their delta and acceleration coefficients. The phonetic decoder is the MLite++ ASR search engine, a Motorola proprietary HMM-based search engine for embedded platforms, with a phoneme loop grammar. The search engine uses both context-independent (CI) and context-dependent (CD) sub word and speaker-independent HMMs, which were trained on a much larger speaker-independent American English database than the above spoken name database.

For comparative purposes, the 85 name entries were carefully transcribed by a phonetician. The number of transcriptions per name entry is varied from 1 to many (due to pronunciation differences associated with the distinct speech varieties), with an average of 3.89 per entry. Using these reference transcriptions and a word-bank grammar (i.e., a list of words with equal probability) of the 85 name entries, the baseline word accuracies of 91.67% and 92.69% are obtained on the speaker-independent test set of the spoken name database with CI and CD HMMs, respectively.

4.1. Same-language experiments using batch mode voice-to-phoneme conversion

To illustrate the effectiveness of the DTW-based feature combination algorithm, consider the following phonetic strings generated from (i) a single sample, (ii) an "average" of two samples and (iii) an "average" of three samples of the name "Larry Votta" (where "mn" signifies mouth noise). The manual reference transcription of this name is /l ɛ r iʲ v ɔ t *a*/.

In general, those phonetic strings generated from more samples have more agreement with the manual transcription than those generated with fewer samples. In particular, the averaged strings tend to eliminate the "random" phonemes (like / ŋ/, /ɐ/, /j/, /z/, and /n/,) and preserve the "consensus" and "majority" phonemes (like /l/, /eʲ/, /r/, /iʲ/, and /ɔ/)—which tend to be associated with the manual transcription. Therefore, the DTW-based feature combination does preserve the commonality of the samples, which is expected to be the abstraction of a voice-tag. It is worthwhile to note that the "new" phoneme, which may not necessarily be in original sample utterance, can be generated by the phonetic decoder from the "average" utterance. For instance, /v/ in $\mathbf{P}_1(\mathbf{X}_{1,2,3})$ is not part of $\mathbf{P}_1(\mathbf{X}_1)$, $\mathbf{P}_1(\mathbf{X}_2)$ or $\mathbf{P}_1(\mathbf{X}_3)$.

Table 1

(i)	$\mathbf{P}_1(\mathbf{X}_1)$	mn	l	eʲ		r	ŋ	b	ɔ	ð	ɔ	mn
	$\mathbf{P}_1(\mathbf{X}_2)$		j	eʲ		r	iʲ	ɐ	ɔ	b	*a*	mn
	$\mathbf{P}_1(\mathbf{X}_3)$	mn	l	eʲ	l	r	iʲ	z	ɔ	b	ɔ	n
(ii)	$\mathbf{P}_1(\mathbf{X}_{1,2})$		l	eʲ		r	iʲ	b	ɔ	ð	*a*	mn
	$\mathbf{P}_1(\mathbf{X}_{2,3})$	mn	l	eʲ	l	r	iʲ	b	ɔ	b	*a*	mn
(iii)	$\mathbf{P}_1(\mathbf{X}_{1,2,3})$	mn	l	eʲ		r	iʲ	b	ɔ	v	*a*	mn

Table 2: Voice-tag word accuracy obtained by batch mode voice-to-phoneme conversion.

	Word Accuracy (%)			
"Average"	Single sample	Two samples	Three samples	Baseline
CI HMMs	87.43	89.33	92.84	91.67
CD HMMs	85.38	89.77	91.23	92.69

Table 2 shows the speech recognition performances of the batch mode voice-to-phoneme conversion. Word accuracy is derived from the test set of the name database where a voice-tag is created with three phonetic strings from a single sample, an "average" of two samples, and an "average" of all three samples of the voice-tag in the speaker-dependent training set.

As expected, the performance increases when the number of averaged samples per voice-tag increases. When three samples are used the performance is very close to the baseline performance. It is interesting to note that the CI HMMs yield a better performance than the CD HMMs. Further investigation reveals that when varying the ASR search engine configuration, such as the penalty at phoneme boundaries, the performance of the CI HMMs degrades drastically while that of the CD HMMs remains consistent.

4.2. Same-language experiments using sequential voice-to-phoneme conversion

In this section we only investigate the phoneme monogram and bigram voice-tag histograms for the sequential voice-to-phoneme conversion. Tables 3 and 4 show the speech recognition performance on the test set of the spoken name database with up to three hypothetical phonetic strings generated per each sample via the phonetic decoder. In these results the voice-tags are created sequentially from the speaker-dependent training set of the name database with both monogram and bigram phoneme histograms.

In these experiments, it is observed that word accuracy generally increases when two or more samples are used: CD HMMs outperform both the CI HMMs and the CD HMMS derived from manual transcriptions. It is also noted that both

Table 3: Voice-tag word accuracy obtained by sequential voice-to-phoneme conversion with bigram phoneme histogram.

	Word accuracy (%) with bigram phoneme histogram			
Sequentially combine hypotheses from	Single sample	Two samples	Three samples	Baseline
CI HMMs	87.7	89.9	90.4	91.67
CD HMMs	87.3	94.0	95.2	92.69

Table 4: Voice-tag word accuracy obtained by sequential voice-to-phoneme conversion with monogram/bigram phoneme histogram and CD HMMs.

	Word accuracy (%) with CD HMMs			
Sequentially combine hypotheses from	Single sample	Two samples	Three samples	Baseline
Monogram histogram	87.3	93.9	95.6	92.69
Bigram Histogram	87.3	94.0	95.2	92.69

monogram and bigram phoneme histograms yield similar performances depending on the number of samples used.

In order to understand the implication of the number of hypothetical phoneme strings generated per sample, in Table 5 we show the performance results while varying this number:

This table shows the word accuracies obtained with CD HMMs, three samples per voice-tag, a bigram phoneme histogram, which is estimated based on the best hypothetical phoneme string of each sample, and 1–7 hypothetical phoneme strings per sample. The best result is 96.1% word accuracy, which is achieved with as little as four hypothetical phoneme strings per sample. Considering both user convenience and recognition performance, we suggest that 3 or 4 hypothetical phoneme strings per sample might be optimal for sequential voice-to-phoneme conversion for voice-tag applications.

5. CROSS-LANGUAGE EXPERIMENTS

In practice, evaluating cross-language performance is complex and poses distinct challenges to same-language performance evaluation. In general, cross-language evaluation can be approached by two principle strategies. One strategy creates voice-tags in several target languages by using language resources, such as HMMs and a looped phoneme grammar, from a single source language. The weakness of this strategy is that it is difficult to normalize the linguistic and acoustic differences across the target languages, a necessary step in creating an evaluation database. The other strategy creates voice-tags in a single target language by using language resources from several distinct source languages. The weakness of this strategy is that language resources differ significantly and it cannot be expected that each source language will be trained with the same amount and type of data. Because we can compare our training data in terms of quantity and type, we opted to pursue the second strategy for the cross-language experiments presented here.

Table 5: Word accuracy obtained by sequential voice-to-phoneme conversion considering hypothetical phoneme string number.

Word accuracy (%) of bigram phoneme histogram							
# of hypos	1	2	3	4	5	6	7
CD HMM	84.2	91.8	95.2	96.1	95.8	96.1	96.1

We selected seven languages as source languages: British English (en-GB), German (de-DE), French (fr-FR), Latin American Spanish (es–LatAm), Brazilian Portuguese (pt-BR), Mandarin (zh-CN-Mand) and Japanese (ja-JP). For each of the source languages, we have sufficient data and linguistic coverage to train generic CD HMMs. The phoneme loop grammar of each source language is constructed from the phoneme set of that language.

Since we used American English in the same-language sequential and batch mode voice-to-phoneme conversion experiments above, and thus have these results for comparison, we selected American English as the target language in the following cross-language experiments. For these, we use the same name database, phonetic decoder, and baseline that we used in the same-language experiments.

5.1. Cross-language experiments using batch mode and sequential voice-to-phoneme conversion

In this investigation, the individual cross-language voice-tag recognition performances are compared to both the same-language results and to each other. To do the latter, a phonological similarity study is conducted between the target language (American English) and each of the selected evaluation languages, the prediction being that cross-language performance would correlate to the relative phonological similarity of the source languages to the target language. We use a pronunciation dictionary as each language's phonological description in order to ensure task independence; because each language's pronunciations are transcribed in a language-independent notation system (similar to the International Phonetic Alphabet), cross-language comparison is possible [16]. Phoneme-bigram (biphoneme) probabilities collected from each dictionary are used as the numeric expression of the phonological characteristics of the corresponding language. The distance between the biphoneme probabilities of each source language and that of the target language is then measured. This metric thus explicitly provides a biphoneme inventory and phonotactic sequence importance. It also implicitly incorporates phoneme inventory and phonological complexity information. Using this method, the distance score is an objective indication of phonological similarity in the source-target

language pair, where the smaller the distance value between the languages, the more similar the pair (see [7] for an in-depth discussion of this biphoneme distribution distance).

The languages that we use in these evaluations are from four language groups defined by genetic relation: (i) Germanic: en-US, en-GB, and de-DE; (ii) Romance: fr-FR, pt-BR, es-LatAm; (iii) Sinitic: zh-CN-Mand and (iv) Japonic: ja-JP. In general, it is expected that closely related languages and contact languages (languages spoken by people in close contact with speakers of the target language [17]), will exhibit greatest phonological similarity. The distance scores relative to American English are provided in the last column of Table 6. Note that the Germanic languages are measured to be the most similar to American English. In particular, the British dialect of English is least distant to American English, and German, the only other Germanic language in the evaluation set, is next. German is followed by French in phonological distance, and French and English are languages with centuries of close contact and linguistic exchange.

This preliminary study thus both substantiates in a quantitative way linguistic phonological similarity assumptions and provides a reference from which to evaluate our results. Based on this study, it is our expectation that cross-language voice-tag application performance will be degraded relative to the voice-tag application performance in the same-language setting, and that the severity of the degradation will be a function of phonological similarity.

Table 6 also shows the cross-language voice-tag application performances of the sequential and batch mode voice-to-phoneme conversion algorithms, where the acoustic models are trained on the seven evaluation languages while the voice-tags are created and applied on American English, a distinct target language. For reference, we also include the American English HMM performance as a baseline.

Apart from the exceptional performance of Mandarin using the sequential phoneme conversion algorithm, the performances generally adhere to the target-source language pair similarity scores identified above. Voice-tag recognition with British English-trained HMMs achieve a word accuracy of 91.37% and recognition with German-trained HMMs realize 90.5%. The higher-than-expected performance rate of Mandarin may be due to some correspondences between the American English and Mandarin databases. The American English database consists of a minority of second language speakers of English, especially native Chinese and Indian speakers.

Thus, the utterances used to train the American English models include some Mandarin-language pronunciations of English words. Secondly, the Mandarin models are embedded with a significant amount of English material (English loan words, e.g.), reflecting a modern reality of language use in China.

The cross-language evaluations show significant performance differences between the two voice-creation algorithms across all of the evaluated languages. The differences are in accordance with our observation in the same-language evaluation. Although there are degradations, the performances of sequential voice-tag creation with HMMs trained on the languages most phonologically similar to American English are very close to the reference performance (92.69%) where the phonetic strings of voice-tags were transcribed manually by an expert.

Table 6: Word accuracies of voice-tag recognition with batch mode and sequential voice-tag creations in cross-language experiments.

Sources	Word Acc. (%) on Target language Voice-tag creations		Distance
	Sequential	Batch	
en-US (baseline)	95.2	91.23	0
en-GB	91.37	87.13	0.61
de-DE	90.50	86.99	1.46
fr-FR	89.91	85.09	1.81
pt-BR	82.75	74.42	1.82
zh-CN-Mand	92.11	84.94	1.85
ja-JP	78.07	67.69	1.90
es-LatAm	89.62	83.33	1.92

6. DISCUSSION AND CONCLUSIONS

We presented two voice-to-phoneme conversion algorithms, each of which utilizes a phonetic decoder and speaker-independent HMMs to create speaker-independent voice-tags. However, these two algorithms employ radically different approaches for sample combination. It is difficult to theoretically compare the algorithms' creation process complexities, though we have observed that the creation processes of both algorithms require similar computational resources (CPU and RAM). The voice-tag created by both algorithms is a set of phonetic strings that require very low storage, making them suitable for embedded platforms. So, for a voice-tag with **N** phonetic strings of an average length L, a voice-tag requires N times L bytes to store phonetic strings. For continuous improvement of voice-tag representation, the sequential creation algorithm retains a phoneme n-gram histogram per voice-tag, which requires approximately $2L$ bytes.In a typical case where $N = 3$ and $L = 7$, 21 bytes are needed for each voice-tag created by the batch-mode algorithm, while the sequential creation algorithm requires 35 bytes for each voice-tag. Both algorithms are shown to be effective in voice-tag applications, as they yield speech recognition performances either comparable to or exceeding a manual reference in same-language experiments.

In the batch mode voice-to-phoneme conversion algorithm, we focused on preserving the input feature vector commonality among multiple samples as a voice-tag abstraction by developing a feature combination strategy. In the sequential voice-to-phoneme conversion approach, we investigated the hierarchy of phonetic consensus buried in the hypothetical phonetic strings of multiple example utterances. We used an n-gram phonetic histogram accumulated sequentially to describe the hierarchy and to select the most

relevant hypothetical phoneme strings to represent a voice-tag.

We demonstrated that the voice-to-phoneme conversion algorithms are not only applicable in a same-language environment, but may also be used in a cross-language setting without significant degradation. For the cross-language experiments, we used a distance metric to show that performance results associated with HMMs trained on languages phonologically similar to the target language tend to be better than results achieved with less similar languages, such that performance degradation is a function of source-target language similarity, providing database characteristics, such as intralingual speech varieties and borrowings, are considered. Our experiments suggest that a cross-language application of a voice-to-phoneme conversion algorithm is a viable solution to voice-tag recognition for resource-poor languages and dialects. We believe this has important consequences given the globalization of mobile devices and the subsequent demand to provide voice technology in new markets.

REFERENCES

[1] T. Holter and T. Svendsen, "Maximum likelihood modeling of pronunciation variation," *Speech Communication*, vol. 29, no. 2–4, pp. 177–191, 1999.

[2] T. Svendsen, F. K. Soong, and H. Purnhagen, "Optimizing baseforms for HMM-based speech recognition," in *Proceedings of the 4th European Conference on Speech Communication and Technology (EuroSpeech '95)*, pp. 783–786, Madrid, Spain, September 1995.

[3] S. J. Young, M. G. Brown, J. T. Foote, G. J. F. Jones, and K. Sparck Jones, "Acoustic indexing for multimedia retrieval and browsing," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, vol. 1, pp. 199–202, Munich, Germany, April 1997.

[4] P. Yu and F. Seide, "Fast two-stage vocabulary-independent search in spontaneous speech," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 1, pp. 481–484, Philadelphia, Pa, USA, March 2005.

[5] K. Thambirratnam and S. Sridharan, "Dynamic match phone-lattice search for very fast and accurate unrestricted vocabulary keyword spotting," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 1, pp. 465–468, Philadelphia, Pa, USA, March 2005.

[6] Y. M. Cheng, C. Ma, and L. Melnar, "Voice-to-phoneme conversion algorithms for speaker-independent voice-tag applications in embedded platforms," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU '05)*, pp. 403–408, San Juan, Puerto Rico, November-December 2005.

[7] C. Liu and L. Melnar, "An automated linguistic knowledge-based cross-language transfer method for building acoustic models for a language without native training data," in *Proceedings of the 9th European Conference on Speech Communication and Technology (InterSpeech '05)*, pp. 1365–1368, Lisbon, Portugal, September 2005.

[8] J. J. Sooful and E. C. Botha, "Comparison of acoustic distance measures for automatic cross-language phoneme mapping," in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP '02)*, pp. 521–524, Denver, Colo, USA, September 2002.

[9] T. Schultz and A. Waibel, "Fast bootstrapping of LVCSR systems with multilingual phoneme sets," in *Proceedings of the 5th European Conference on Speech Communication and Technology (EuroSpeech '97)*, pp. 371–373, Rhodes, Greece, September 1997.

[10] T. Schultz and A. Waibel, "Polyphone decision tree specialization for language adaptation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00)*, vol. 3, pp. 1707–1710, Istanbul, Turkey, June 2000.

[11] Y. M. Cheng, C. Ma, and L. Melnar, "Cross-language evaluation of voice-to-phoneme conversions for voice-tag application in embedded platforms," in *Proceedings of the 9th International Conference on Spoken Language Processing (InterSpeech '06)*, pp. 121–124, Pittsburgh, Pa, USA, September 2006.

[12] F. K. Soong and E.-F. Huang, "A tree-trellis based fast search for finding the N-best sentence hypotheses in continuous speech recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '91)*, vol. 1, pp. 705–708, Toronto, Ontario, Canada, May 1991.

[13] G. Hernandez-Abrego, L. Olorenshaw, R. Tato, and T. Schaaf, "Dictionary refinements based on phonetic consensus and non-uniform pronunciation reduction," in *Proceedings of the 8th International Conference on Spoken Language Processing (ICSLP '04)*, pp. 1697–1700, Jeju Island, Korea, October 2004.

[14] T. M. Cover and J. A. Thomas, *Element of Information Theory*, John Wiley & Sons, New York, NY, USA, 1991.

[15] ES 201 108, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; front-end feature extraction algorithm; compression algorithms," 2000.

[16] L. Melnar and J. Talley, "Phone merger specification for multilingual ASR: the Motorola polyphone network," in *Proceedings of the 15th International Congress of Phonetic Sciences (ICPhS '03)*, pp. 1337–1340, Barcelona, Spain, August 2003.

[17] R. Trask, *A Dictionary of Phonetics and Phonology*, Routledge, London, UK, 1996.