ASR for Under-Resourced Languages from Probabilistic Transcription

Mark Hasegawa-Johnson[1], Adrian KC Lee[2], Ed Lalor[3], Preethi Jyothi[1], Daniel McCloy[2], Majid Mirbagheri[2], Giovanni di Liberto[3], Amit Das[1], Brad Ekin[2], Chunxi Liu[4], Vimal Manohar[4], Hao Tang[5], Nancy Chen[6], Paul Hager[7], Tyler Kekona[2], and Rose Sloan[8]

1. University of Illinois, 2. University of Washington, 3. Trinity College, Dublin, 4. Johns Hopkins University, 5. Toyota Technological Institute Chicago, 6. Institute for Infocomm Research, 7. MIT, 8. Yale University

March 1, 2016

1 Introduction

Automatic speech recognition (ASR) has the potential to provide database access, simultaneous translation, and text/voice messaging services to anybody, in any language, dramatically reducing linguistic barriers to economic success. To date, ASR has failed to achieve its potential, because successful ASR requires very large labeled corpora. Current methods require about 1000 hours of transcribed speech per language, transcribed at a cost of about 6000 hours of human labor; the human transcribers must be computer-literate, and they must be native speakers of the language being transcribed. In many languages, the idea of recruiting dozens of computer-literate native speakers is impractical, sometimes even absurd.

Instead of recruiting native transcripts in search of a perfect reference transcription, this paper proposes the use of probabilistic transcriptions. A probabilistic transcription is a probability mass function, $\rho_\Phi(\phi)$, specifying, as a real number between 0 and 1, the probability that any particular phonetic transcription ϕ is the correct transcription of the utterance. Prior to this work, machine learning has almost always assumed that the training dataset contains either deterministic transcriptions ($\rho_{DT}(\phi) \in \{0,1\}$, commonly called "supervised training") or completely untranscribed utterances (commonly called "unsupervised training," in which case we assume that $\rho_{LM}(\phi)$ is given by some *a priori* language model). This article proposes that, even in the absence of a deterministic transcript, there may be auxiliary sources of information that can be applied to create a probabilistic transcription whose entropy is lower than that of the language model, and that machine learning methods applied to the probabilistic transcription are able to make use of its reduced entropy in order to learn a better speech recognizer. In particular, this paper considers three useful auxiliary sources of information:

1. SELF TRAINING: ASR pre-trained in other languages is used to transcribe unlabeled training data in the target language.

2. MISMATCHED CROWDSOURCING: Human crowd workers who don't speak the target language are asked to transcribe it as if it were a sequence of nonsense syllables.

3. EEG DISTRIBUTION CODING: Humans who do not speak the target language are asked to listen to its extracted syllables, and their EEG responses are interpreted as a probability mass function over possible phonetic transcriptions.

2 Background

Consider the problem of developing speech technology in a language with few internet-connected speakers. Suppose we require that, in order to develop speech technology, it is necessary first to have (1) some amount of recorded speech audio, and (2) some amount of text written in the target language. These two requirements can be met by at least several hundred languages: speech audio can be recorded during weekly minority-language broadcasts on a local radio station, and text can be acquired from printed pamphlets and literacy primers. Recorded speech is, however, not usually transcribed; and the requirement of native language transcription is beyond the economic capabilities of many minority-language communities.

2.1 Existing Approaches to ASR in Under-Resourced Languages

Krauwer [33] defined an under-resourced language to be one that lacks one or more of: stable orthography, significant presence on the internet, linguistic expertise, monolingual tagged corpora, bilingual electronic dictionaries, transcribed speech, pronunciation dictionaries, or other similar electronic resources. Berment [4] defined a rubric for tabulating the resources available in any given language, and proposed that a language should be called "under-resourced" if it scored lower than 10.0/20.0 on the proposed rubric. By these standards, technology methods for under-resourced languages are most often demonstrated on languages that are not really under-resourced: for example, ASR may be trained without transcribed speech, but the quality of the resulting ASR can only be scientifically proven by measuring its phone error rate (PER) or word error rate (WER) using transcribed speech. The intention, in most cases, is to create methods that can later be ported to languages that truly lack resources.

The International Phonetic Alphabet (IPA [23]) is a set of symbols representing speech sounds (phones) defined by the principle that, if two phones are used in any language to make meaningful linguistic contrasts (i.e., they represent distinct phonemes), then those phones should have distinct symbolic representations in the IPA. This makes the IPA a natural choice for transcriptions used to train cross-language ASR systems, and indeed ASR in a new language can be rapidly deployed using acoustic models trained to represent every distinct symbol in the IPA [49]. However, because IPA symbols are defined phonemically, there is no guarantee of cross-language equivalence in the acoustic properties of the phones they represent. This problem arises even between dialects of the same language: a monolingual Gaussian mixture model (GMM) trained on five hours of Levantine Arabic can be improved by adding ten hours of Standard Arabic data, but only if the log likelihood of cross-dialect data is scaled by 0.02 [21].

Better cross-language transfer of acoustic models can be achieved, but only by using structured transfer learning methods, including neural networks (NN) and subspace Gaussian mixture models (SGMM). NN transfer learning can be categorized as tandem, bottleneck, pre-training, phone mapping, and multi-softmax methods. In a tandem system, outputs of the NN are Gaussianized, and used as features whose likelihood is computed with a GMM [19]; in a bottleneck system, features are extracted from a hidden layer rather than the output layer. Both tandem [54] and bottleneck [58] features trained on other languages can be combined with GMMs trained on the target language in order to improve word error rate (WER).

A hybrid ASR is a system in which the NN terminates in a softmax layer, whose outputs are interpreted as phone [44] or senone [9] probabilities. Knowledge of the target language phone inventory is necessary to train a hybrid ASR, but it is possible to reduce WER by first pre-training the NN hidden layers with multilingual data [20, 55]. A hybrid ASR can be constructed using very little in-language speech data by adding a single phone-mapping layer to the output of the multilingual NN; the phone mapping layer can be trained using a small amount of in-language speech data [52], even if context-dependent senones are mapped instead of phones [12]. A multi-softmax system integrates phone mapping into the original training procedure, by training a network with several different language-dependent softmax layers, each of which is the linear transform of a multilingual shared hidden layer. Multi-softmax systems have reduced WER in tandem [47], bottleneck [58], and hybrid [20] ASR.

SGMM transfer learning uses language-dependent GMMs, each of which is the linear interpolation of language-independent mean and variance vectors. SGMM can be combined with other methods for further improvement, e.g., 16% relative WER reduction was achieved in a Tamil ASR by combining SGMM with an acoustic data normalization technique [40], and further reductions were obtained in Afrikaans by using bottleneck features in an SGMM [22].

Self-training is a class of semi-supervised learning techniques in which ASR is first trained on labeled corpora in other languages, then used to label data in the target language. Self-labeled data in the target language are then used to train or adapt the ASR [38, 7]. Self-training is most useful when the in-language training data are first filtered, to exclude frames with confidence below a threshold. The posterior probability computed from the ASR lattice is a useful confidence score [57], but it is also possible to learn an improved confidence score by combining multiple sources of information [59].

Under-resourced languages often lack any pronunciation dictionary. It is possible to train a stable grapheme-to-phoneme transducer using a dictionary with 15,000 entries, and in some languages a dictionary of this size can be mined from sources such as Wiktionary [48]. In languages without any dictionary of this size, it may be possible to approximate pronunciation by treating each orthographic character as an acoustic model [30, 8, 16, 34]. Even an ambiguous G2P can often be disambiguated by the use of context-dependent graphemic models [30]; if the number of trigraphemes gets too large, acoustic models can be interpolated within an eigentrigrapheme space [32]. Optimal WER in Standard Arabic was achieved by using phoneme-based pronunciations for the most frequent 500 words, and grapheme-based pronunciations for all less frequent words [13]. In Amharic, optimal WER was achieved using a morpheme-based language model, combined with a hybrid acoustic model set including both triphones and context-dependent sylabic units [56]. In Hindi, optimal WER was achieved using a one-to-one character-based grapheme-to-phoneme (G2P) transducer (essentially a grapheme-based acoustic model), modified by a very small set (3) of surface phonological rules [27]. The three rules were proposed based on phonological descriptions of Hindi, then applied or discarded in response to application probabilities learned using a very small (200-word) pronunciation dictionary.

2.2 Mismatched Crowdsourcing

In [26], a methodology was proposed that bypasses the need for native language transcription: mismatched crowdsourcing sends target language speech to crowd-worker transcribers who have no knowledge of the target language, then uses explicit mathematical models of second language phonetic perception to recover an equivalent phonetic transcription (Fig. 1). Majority voting is re-cast, in this paradigm, as a form of error-correcting code (redundancy coding), which effectively increases the capacity of the noisy channel; interpretation as a noisy channel permits us to explore more effective and efficient forms of error-correcting codes.

Assume that cross-language phoneme misperception is a finite-memory process, and can therefore be modeled by a finite state transducer (FST). The complete sequence of representations from utterance lan-

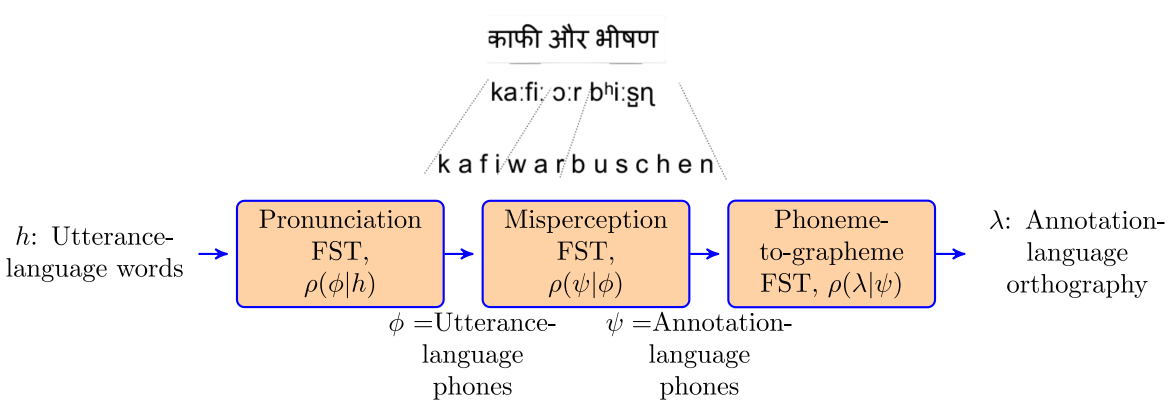

Figure 1: Mismatched Crowdsourcing: crowd workers on the web are asked to transcribe speech in a language they do not know. Annotation mistakes are modeled by a finite state transducer (FST) model of utterance-language pronunciation variability (reduction and coarticulation), composed with an FST model of non-native speech misperception (mapping utterance-language phones to annotation-language phones), composed with an inverted grapheme-to-phoneme (G2P) transducer.

guage to annotation language can therefore be modeled as a noisy channel represented by the composition of up to three consecutive FSTs (Fig. 1): a pronunciation model, a misperception model, and an inverted grapheme-to-phoneme (G2P) transducer. The pronunciation model is an FST representing processes that distort the canonical phoneme string during speech production, including processes of reduction and coarticulation. The misperception model represents the mapping of the uttered phone string (in symbols matching the phone set of the spoken language) to the perceived phone string (in symbols matching the phone set of the annotation language). Finally, the transcriber maps heard phones to nonsense words in the annotation language; the mapping from phones to orthography is an inverted G2P.

Preliminary experiments in mismatched crowdsourcing were carried out[26] using Hindi speech excerpts extracted from podcasts [53]. Approximately one hour of speech was extracted from the podcasts (about 10000 word tokens in total) and phonetically transcribed by a Hindi speaker. The data were then segmented into very short speech clips (1 to 2 seconds long). The crowd workers were asked to listen to these short clips and provide English text, in the form of nonsense syllables, that most closely matched what they heard. The English text (λ) was aligned with the Hindi phone transcripts (ϕ) using a learned transducer, $\rho(\lambda|\phi)$, called a misperception G2P because it replaces both the misperception and G2P transducers from Fig. 1. Fig. 2 shows a schematic diagram of the misperception G2P, with learned Levenshtein distances. This FST probabilistically maps each Hindi phone to either a single English letter or a pair of English letters. The FST substitution costs, deletion costs and insertion costs are learned to maximize $\rho(\lambda|\phi)$ using the expectation maximization algorithm (EM) [10].

2.3 Electrophysiology of Speech Perception

The human auditory system is sensitive to within-category distinctions in speech sounds, and such precategorical perceptual distinctions may be lost in transcription tasks, where listeners must filter their percepts through the limited number of categorical representations available in their native language orthography. EEG distribution coding is a proposed new method that interprets the electrical evoked potentials of untrained listeners (measured by an electro-encephalograph or EEG) as a posterior probability distribution over the phone set of the utterance language (Fig. 3). Transcribers, in this scenario, listen to speech in both their native language and an unfamiliar non-native target language, while their EEG responses are recorded. From their responses to English speech, an English-language EEG phone recognizer is trained [11]. Misperception probabilities $\rho(\psi|\phi)$ are then estimated: for each non-native phone ϕ, the classifier outputs are interpreted as an estimate of $\rho(\psi|\phi)$ for all $\phi \in \Phi$, the native phone inventory.

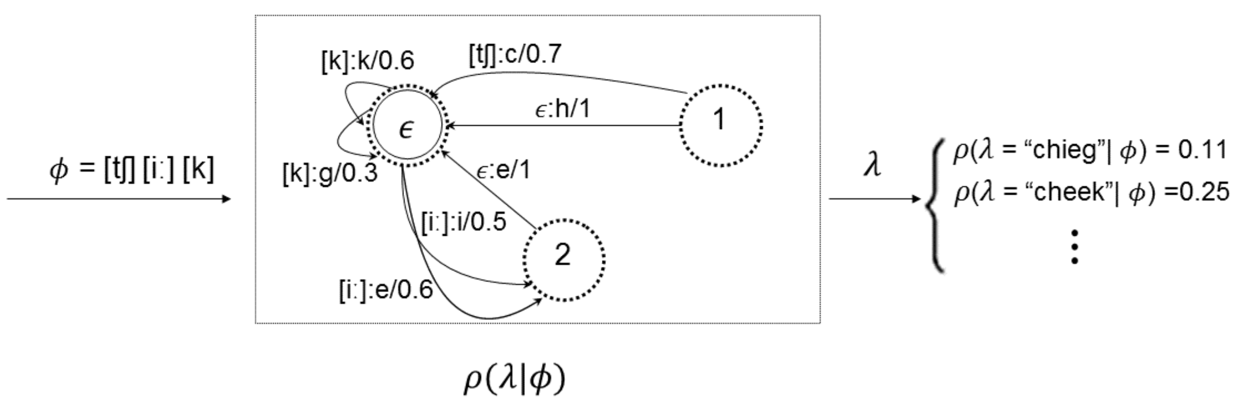

Figure 2: Mismatch FST model of $\rho(\lambda|\phi)$, the probability of English transcription λ given Hindi phone sequence ϕ. Figure modified slightly from [26].

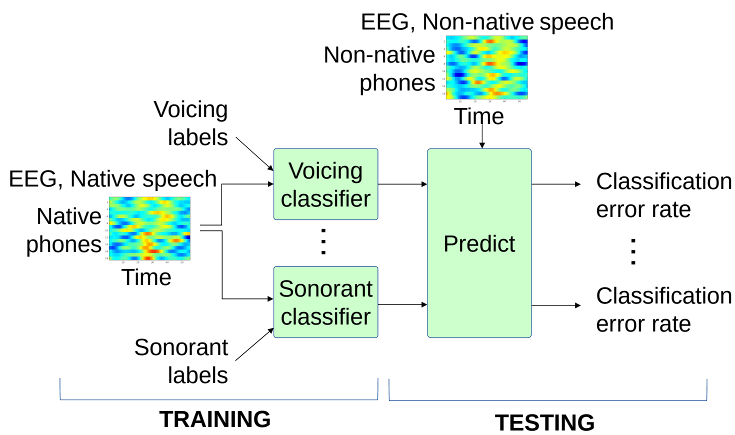

Figure 3: EEG responses are recorded while listeners hear speech in their native language and an unfamiliar non-native language. For each listener, a bank of distinctive feature classifiers are trained. Those classifiers are then applied to the EEG responses to non-native speech, estimating a listener-language transcription of the non-native speech.

3 Algorithms that Induce a Probabilistic Transcription

Three different experimental sources were tested for the creation of a PT. Self-training is now well-established in the field of under-resourced ASR; we adopted the algorithm of Vesely, Hannemann and Burget [57]. Mismatched crowdsourcing used original annotations collected using our own previously published methods [28]. EEG was not used independently here, but rather, was used to learn a misperception model applicable to the interpretation of mismatched crowdsourcing.

3.1 Self Training

The first set of PTs were computed using NN-HMM self-training. The Kaldi toolkit [46] was used to train a multilingual NN-HMM in six languages not including the target language. A training script provided by Vesely, Hannemann and Burget [57] was then used to compute a posterior probability $\pi(\phi_m^\ell|x_t^\ell)$ for each frame x_t^ℓ of audio in the target language as shown on the left side of Fig. 4. The phone posteriors, $\pi(\phi_m^\ell|x^\ell)$, were then used to compute senone posteriors $\pi(s_t^\ell|x^\ell)$ (right side of Fig. 4), which served as targets for re-estimating the neural network weights.

In contrast to the approach in [57], which used the best path alignment as the target in frame cross-entropy training, we achieved better performance using the posteriors as soft-targets (Fig. 4). However, we did follow the recommendation in [57] to scale the amount of transcribed data by 2 to create a good balance between transcribed and untranscribed data.

3.2 Mismatched Crowdsourcing

The second set of PTs were computed by sending audio in the target language to non-speakers of the target language (all transcribers were speakers of American English, and the plurality were monolingual), and asking them to write what they hear. Denote using T the set of text transcripts produced by these English-speaking crowd workers. Mismatched transcripts must be converted into the form of a pmf over target-language phone sequences, $\pi(\phi^\ell|T)$. As an intermediate step towards this goal, prior work [28] developed techniques to merge the transcripts in T into a distribution $\rho(\lambda|T)$ over representative crowd-worker transcripts, denoted λ. The representative transcripts, λ, use the same orthography as the crowd worker transcripts T (English nonsense syllables), but differ in reliability, as follows. Formation of λ from T involves data filtering (choosing the most

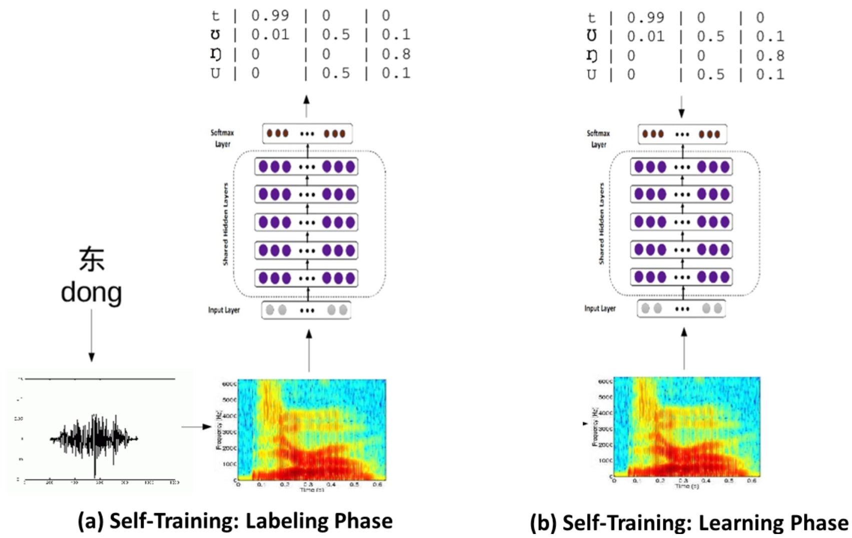

Figure 4: The self-training method of [57] includes a labeling phase and a learning phase. (a) Labeling phase: an ASR trained on other languages (here Cantonese) is used to compute posterior phone probabilities $\pi(\phi_t^\ell|x^\ell)$ in the test language (here Mandarin). (b) Learning phase: posterior phone probabilities are used as targets for DNN re-training.

representative 5 transcripts out of each set of 10, using pair-wise string edit distances among the transcripts), conversion of annotation-language orthography into a pseudo-phonetic code in order to represent common letter sequences (e.g., "ee" in English orthography represents the phoneme /i/), and a weighted voting scheme in which the weight of each transcript is proportional to the frequency with which it matches the other transcripts.

Once transcripts have been aligned and filtered to create the orthographic confusion network $\rho(\lambda|T)$, they are then translated into a distribution over phonemic transcriptions according to:

$$\begin{aligned}\rho(\phi|T) &= \sum_{\lambda} \rho(\phi|\lambda, T)\rho(\lambda|T) \\ &\approx \max_{\lambda} \rho(\phi|\lambda)\rho(\lambda|T) \\ &= \max_{\lambda} \left(\frac{\rho(\lambda|\phi)}{\rho(\lambda)}\rho(\phi)\right)\rho(\lambda|T) \end{aligned} \tag{1}$$

The terms other than $\rho(\lambda|T)$ in Equation (1) are estimated as follows. $\rho(\lambda)$ is modeled using a simple context-free prior over the letter sequences in λ. $\rho(\phi)$ is modeled using a bigram phone language model, trained on a corpus of Wikipedia text in the target language, converted into phone sequences as described in Section 4.3. $\rho(\lambda|\phi)$ is called the misperception G2P, as it maps to graphemes in the annotation language, λ, from phones in the utterance language, ϕ. This section describes methods that estimate $\rho(\lambda|\phi)$ directly; Section 3.3 describes methods that decompose $\rho(\lambda|\phi)$ into separate misperception and G2P transducers.

The misperception G2P ($\rho(\lambda|\phi)$) can be trained directly using the Carmel toolkit [31] as an FST mapping phones to letters based on representative transcripts λ (and their corresponding native transcripts) for speech *in languages other than the target language*. We assume that misperceptions depend more heavily on the annotation language than on the utterance language, and that therefore a model $\rho(\lambda|\phi)$ trained using a universal phone set for ϕ is also a good model of $\rho(\lambda|\phi)$ for the target language. Note that, while this assumption is not entirely accurate, it is necessitated by the requirement that no native transcriptions in the target language can be used in building any part of our system.

3.3 Estimating Misperceptions from Electrocortical Responses

The misperception G2P described in Section 3.2 was estimated using a combination of mismatched and deterministic transcripts of non-target languages. However, With a small amount of transcribed data in the utterance language, it is possible to estimate the misperception G2P using electrocortical measurements of non-native speech perception. In this approach, the misperception G2P is decomposed into two separate transducers, a misperception transducer $\rho(\psi|\phi)$, and an annotation-language G2P $\rho(\lambda|\psi)$:

$$\rho(\lambda|\psi) = \sum_{\psi} \rho(\lambda|\psi)\rho(\psi|\phi) \tag{2}$$

where ϕ is a phone string in the utterance language, ψ is a phone string in the annotation language, and λ is an orthographic string in the annotation language. $\rho(\lambda|\psi)$ is an inverted G2P in the annotation language, e.g., trained on the CMU dictionary of American English pronunciations [35]. $\rho(\psi|\phi)$ is the mismatch transducer, specifying the probability that a phone string ϕ in the utterance language will be mis-heard as the annotation-language phone string ψ.

In principle, the mismatch transducer could be computed empirically from a phone confusion matrix, if experimental data on phone confusions were available for all phones in the target language, and those data were based on responses from listeners with the same language background as the crowd worker transcribers. These goals are hard to meet. An alternative is to use distinctive feature representations (originally proposed to characterize the perceptual and phonological natural classes of phonemes [24]) to predict misperceptions based on differences between the distinctive feature values of transcription- and target-language phones. Given the assumption that every distinctive feature shared by phones ϕ and ψ independently increases their confusion probability, their confusion probability can be expressed as

$$\rho(\psi|\phi) \propto \exp\left(-\sum_{k=1}^{K} w_k\right) \tag{3}$$

where $w_k(\phi)$ is the contribution of the k^{th} feature in the misperception of the of phone ϕ as phone ψ. If a feature is perceived similarly across the two languages its contribution would be lower when when the two phones share the same value of that feature. The assumption of independence is a simplifying assumption, given that many distinctive features have overlapping acoustic correlates. For example, the frequencies of the *two* lowest resonances of the vocal tract (the primary cues for vowel identity) are determined by articulatory gestures of the lips, jaw and tongue that are commonly represented by *three or more* distinctive features (e.g., height, backness, rounding, and advanced tongue root). Moreover, the weights w_k will probably also depend on properties of the speaker and listener (language, dialect, and idiolect), but data to train such a rich model do not exist.

However, a reasonable approximate model can be learned by assuming that w_k depend only on information about the listener, which can be incorporated via measurements of electrocortical activity. In particular, the weights w_k of the distinctive features can be set based on similarity of electrocortical responses (measured using EEG) as determined by a classifier trained on distinctive feature representations and electrocortical responses to the listener's native language phones. Thus, given a set of EEG response signals $y(\psi)$ recorded when a listener hears audio corresponding to phone ψ in the annotation language, and supposing that $g_k(y(\psi))$ is the output of a binary classifier trained on EEG responses to phones in his/her native language to detect the k^{th} distinctive feature of phone ψ[11], then the contributions in Eq. (3) can be estimated as

$$w_k = -\ln \Pr\left\{g_k(y(\psi)) = f_k(\psi)\right\} \tag{4}$$

4 Algorithms for Training ASR Using Probabilistic Transcription

A deterministic transcription is a sequence of phone symbols, $\phi^\ell = [\phi^\ell_1, \ldots, \phi^\ell_M]$ where ϕ^ℓ_m is a symbol drawn from the phone set of the utterance language. We assume that ϕ^ℓ_m can be encoded using an IPA symbol [23]. The superscript specifies that ϕ^ℓ is the transcription of the ℓ^{th} waveform in a database; the collection of all transcriptions is $\phi = \left\{\phi^1, \ldots, \phi^L\right\}$.

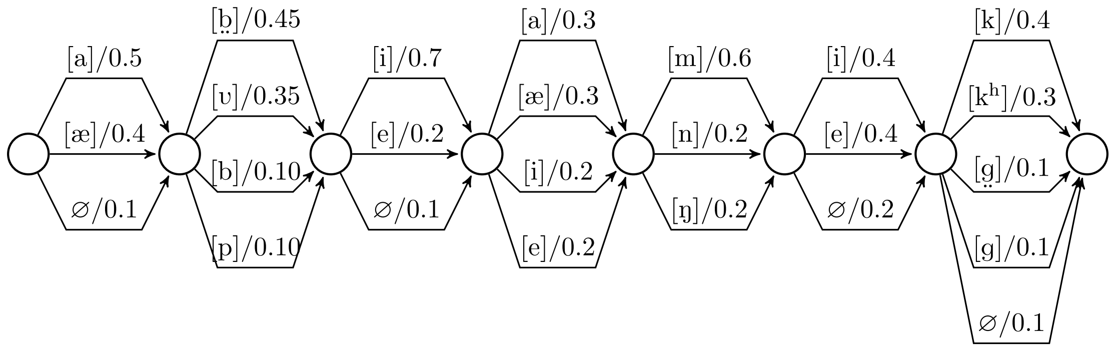

Figure 5: A probabilistic transcription (PT) is a probability mass function (pmf) over candidate phonetic transcriptions. All PTs considered in this paper can be expressed as confusion networks, thus, as sequential pmfs over the null-augmented space of IPA symbols. In this schematic example, $\varnothing$ is the null symbol, symbols in brackets are IPA, and numbers indicate probabilities.

A probabilistic transcription is a probability mass function (pmf) over the set of deterministic transcriptions. Capital letters denote random variables, lowercase denote instances, and blackboard font denote sets. Φ^ℓ_m is a random variable whose instance is $\phi^\ell_m \in \mathrm{V}_\phi$, where V_ϕ is the union of the set of IPA symbols with the null symbol ($\varnothing$), thus $\mathrm{V}_\phi = \{\ \varnothing$, [a],[i],[ʌ],[æ], $\ldots\}$, with cardinality $|\mathrm{V}_\phi|$ equal to one plus the number of distinct IPA symbols. Similarly, Φ^ℓ is a random variable whose instance is $\phi^\ell \in \mathrm{V}^*_\phi$, where V^*_ϕ denotes the set of all sequences composed of symbols in V_ϕ. Denote the probability of transcription ϕ^ℓ as $\rho_{\Phi^\ell}(\phi^\ell)$, where the symbol "$\rho$" is selected to emphasize that $\rho_{\Phi^\ell}(\phi^\ell)$ is a reference distribution—a distribution specified by the probabilistic transcription process, and not dependent on ASR model parameters. The distribution label Φ^ℓ is omitted when clear from the instance label, e.g., $\rho_{\Phi^\ell}(\phi^\ell)$ may be abbreviated as $\rho(\phi^\ell)$, but $\rho_{\Phi^\ell}(u)$ may not. A deterministic transcription is a degenerate probabilistic transcription, in which $\rho(\phi^\ell) \in \{0, 1\}$. A database consists of L speech waveforms, each containing T frames with M associated phone labels, $M < T$. Superscript denotes waveform index, while subscript denotes frame or phone index. Absence of either superscript or subscript denotes a collection, thus $\Phi = \left\{\Phi^1, \ldots, \Phi^L\right\}$ (with instance value $\phi = \left\{\phi^1, \ldots, \phi^L\right\}$) is the random variable over all transcriptions of the database. In all of the work described in this paper, the probabilistic transcription is represented as a confusion network [39], meaning that it is the product of independent symbol pmfs $\rho(\phi^\ell_m)$:

$$\rho(\phi) = \prod_{\ell=1}^{L} \rho(\phi^\ell) = \prod_{\ell=1}^{L} \prod_{m=1}^{M} \rho(\phi^\ell_m) \tag{5}$$

The pmf $\rho(\phi^\ell)$ can be represented as a weighted finite state transducer (wFST) in which edges connect states in a strictly left-to-right fasion without skips, and in which the edges connecting state m to state $m+1$ are weighted according to the pmf $\rho(\phi^\ell_m)$ (Fig. 5).

The ℓ^{th} waveform is represented by acoustic feature matrix $x^\ell = [x^\ell_1, \ldots, x^\ell_T]$, where x^ℓ_t is an acoustic feature vector. Its phone transcription $\phi^\ell = [\phi^\ell_1, \ldots, \phi^\ell_M]$ determines the sequence but not the durations of senones (HMM states) $s^\ell = [s^\ell_1, \ldots, s^\ell_T]$. An automatic speech recognizer (ASR) is a parameterized probability mass function, $\pi(x, s|\phi, \theta)$, specifying the dependence of random variables x and s on the phone transcription ϕ and the parameter vector θ, where the notation $\pi(\cdot)$ denotes a pmf dependent on the ASR parameter vector. We assume a hidden Markov model [3], therefore

$$\pi(x, s|\phi, \theta) = \prod_{\ell=1}^{L} \prod_{t=1}^{T} \pi(s^\ell_t | s^\ell_{t-1}, \phi^\ell, \theta)\pi(x^\ell_t | s^\ell_t, \phi^\ell, \theta)$$

4.1 Maximum Likelihood Training

Consider two observation-conditional sequence distributions $\pi(s,\phi|x,\theta)$ and $\pi(s,\phi|x,\theta')$, with parameter vectors θ and θ' respectively. The cross-entropy between these distributions is:

$$H(\theta\|\theta') = -\sum_{s,\phi}\pi(s,\phi|x,\theta)\ln\pi(s,\phi|x,\theta') \tag{6}$$

$$= \sum_{s,\phi}\pi(s,\phi|x,\theta)\left(\ln\pi(x|\theta') - \ln\pi(s,\phi,x|\theta')\right) \tag{7}$$

$$= \mathcal{L}(\theta') - Q(\theta,\theta') \tag{8}$$

where the data log likelihood, $\mathcal{L}(\theta')$, and the expectation maximization (EM) quality function, $Q(\theta,\theta')$ [10], are defined by

$$\mathcal{L}(\theta') = \ln\pi(x|\theta') \tag{9}$$

$$Q(\theta,\theta') = \sum_{s,\phi}\pi(s,\phi|x,\theta)\ln\pi(s,\phi,x|\theta') \tag{10}$$

The Kullback-Leibler divergence between $\pi(s,\phi|x,\theta)$ and $\pi(s,\phi|x,\theta')$ is $D(\theta\|\theta') = H(\theta\|\theta') - H(\theta\|\theta)$. Since $D(\theta\|\theta') \geq 0$ [50],

$$\mathcal{L}(\theta') - \mathcal{L}(\theta) \geq Q(\theta,\theta') - Q(\theta,\theta) \tag{11}$$

Given any initial parameter vector θ_n, the expectation maximization (EM) algorithm finds $\theta_{n+1} = \text{argmax}\, Q(\theta_n,\theta')$, thereby maximizing the minimum increment in $\mathcal{L}(\theta)$. For GMM-HMMs, the quality function $Q(\theta,\theta')$ is convex and can be analytically maximized; for DNN-HMMs it is non-convex, but can be maximized using gradient ascent.

The probability $\pi(x,s,\phi|\theta)$ is computed by composing the following three weighted FSTs:

$$\mathbf{AM}: s^\ell \rightarrow s^\ell/\pi(x^\ell|s^\ell,\phi^\ell,\theta) \tag{12}$$

$$\mathbf{HC}: s^\ell \rightarrow \phi^\ell/\pi(s^\ell|\pi^\ell,\theta) \tag{13}$$

$$\mathbf{PT}: \phi^\ell \rightarrow \phi^\ell/\rho(\phi^\ell) \tag{14}$$

where the notation has the following meaning. The probabilistic transcription, **PT**, is an FST that maps any phone string $\phi^\ell \in \mathrm{V}_\phi^*$ to itself. This mapping is deterministic and reflexive, but comes with a path cost determined by the transcription probability $\rho(\phi^\ell)$, as exemplified in Fig. 5. The HMM-expansion transducer, **HC**, maps any senone sequence s^ℓ to a phone sequence ϕ^ℓ. This FST is the composition of the **H** and **C** transducers defined by Mohri, Pereira and Riley [41]. This mapping is non-deterministic, and the path cost is determined by the HMM transition weights distribution $a_{ij} = \pi(s_t^\ell = j|s_{t-1}^\ell = i,\phi^\ell,\theta)$:

$$\pi(s^\ell|\phi,\theta) = \prod_{\ell=1}^{L}\prod_{t=1}^{T} a_{s_{t-1}^\ell s_t^\ell} \tag{15}$$

The acoustic modeling transducer **AM** maps any senone sequence to itself. This mapping is deterministic and reflexive, but comes with a path cost determined by the acoustic modeling probability

$$\pi(x^\ell|s^\ell,\phi^\ell,\theta) = \prod_{\ell=1}^{L}\prod_{t=1}^{T}\pi(x_t^\ell|s_t^\ell,\theta^\ell) \tag{16}$$

The joint probability $\pi(\phi^\ell,s^\ell,x^\ell|\theta)$ is computed by composing the FSTs, then finding the total cost of the path through $(\mathbf{AM}\circ\mathbf{HC}\circ\mathbf{PT})$ with input string ϕ^ℓ and output string s^ℓ. The posterior probability $\pi(\phi^\ell,s^\ell|x^\ell,\theta)$ is computed by pushing the composed FST, then finding the total cost of the path through push $(\mathbf{AM}\circ\mathbf{HC}\circ\mathbf{PT})$.

The parameter vector θ includes the HMM transition probabilities, $a_{ij} = \pi(s_t^\ell = j|s_{t-1}^\ell = i,\phi,\theta)$, and the parameters of the acoustic model $b_j(x_t^\ell) = \pi(x_t^\ell|s_t^\ell = j,\theta)$. Computing the analytical maximum or gradient of $Q(\theta,\theta')$ requires summation over all possible state alignments $s \in \mathrm{V}_s$. The summation can be performed efficiently using the Baum-Welch algorithm, but experimental tests reported in this paper did not do so, for reasons described in the next subsection.

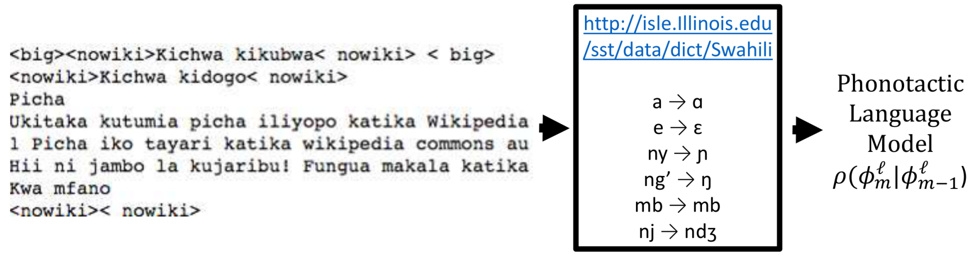

Figure 6: A phonotactic language model (a bigram language model over phone sequences) can be trained using text data downloaded from Wikipedia (left), then converted into phone strings in the target language using a simple character-based grapheme-to-phoneme transducer (center). In this example, the target language is Swahili.

4.2 Segmental Viterbi Training

The previous subsection demonstrates that $\mathcal{L}(\theta')$ can be increased, at each step of the EM algorithm, by maximizing $Q(\theta,\theta')$. Though $Q(\theta,\theta') - Q(\theta,\theta)$ is a lower bound on $\mathcal{L}(\theta') - \mathcal{L}(\theta)$, Q has properties that make it undesirable as an optimizer for $\mathcal{L}$. Suppose, as often happens, that there is a poor phone sequence, ϕ^p, that is highly unlikely given the correct parameter vector θ^*, meaning that $\pi(\phi^p, s, x|\theta^*)$ is very low. Suppose that the initial parameter vector, θ, is less discriminative, so that $\pi(\phi^p, s, x|\theta) > \pi(\phi^p, s, x|\theta^*)$. In this case $Q(\theta,\theta^*)$ is dominated by the term $\pi(\phi^p, s, x|\theta)\ln\pi(\phi^p, s, x|\theta^*)$, therefore θ^* will never show up as the optimizer of $Q(\theta,\theta')$. Indeed, the best speech recognizer is a parameter vector θ^* that completely rules out poor transcriptions, setting $\pi(\phi^p, s, x|\theta^*) = 0$; but in this case $Q(\theta,\theta^*) = -\infty$, so the EM algorithm can never find a parameter vector θ^* that sets to zero the probability of a poor transcription.

Deterministic transcription does not have this problem, because the transcription specifies the phone sequence. With probabilistic transcription, however, the problem is quite common: if the human transcribers fail to rule out ϕ^p (e.g., because the correct and incorrect transcriptions are perceptually indistinguishable in the language of the transcribers), then the EM algorithm will also never learn to rule out ϕ^p. EM is unable to learn zero-valued probabilities.

EM's inability to learn zero-valued probabilities can be ameliorated by using the segmental K-means algorithm [25], which bounds $\mathcal{L}(\theta')$ as $\mathcal{L}(\theta') - \mathcal{L}(\theta) \geq R(\theta,\theta') - \mathcal{L}(\theta)$, where

$$R(\theta,\theta') = \ln\pi(s^*(\theta), \phi^*(\theta), x|\theta') \tag{17}$$

$$s^*(\theta), \phi^*(\theta) = \operatorname*{argmax}_{s,\phi} \pi(s,\phi|x,\theta) \tag{18}$$

Given an initial parameter vector θ, therefore, it is possible to find a new parameter vector θ' with higher likelihood by computing its maximum-likelihood senone sequence and phoneme sequence $s^*(\theta), \phi^*(\theta)$, by maximizing θ' with respect to $s^*(\theta)$ and $\phi^*(\theta)$, and by then replacing θ with θ' only if $R(\theta,\theta')$ is greater than $\mathcal{L}(\theta) = \ln\sum\pi(s,\phi,x|\theta)$. In practice, maximizing $R(\theta,\theta')$ rather than $Q(\theta,\theta')$ is useful for probabilistic transcription because it reduces the importance of poor phonetic transcriptions.

4.3 Using a Language Model During Training

A PT contains significant amount of information beyond any single transcript extracted from the PT. Motivated by this, the statistics for the MAP estimation are accumulated from a lattice derived from the cascade $\mathbf{AM}\circ\mathbf{HC}\circ\mathbf{PT}$, rather than reducing the PT to its single best path. Though it is disadvantageous to reduce a PT to its best path, it is nevertheless advantageous to incorporate as much information as possible from the language model during adaptation. Define $\mathbf{G}$ to be an FST representing the modeled phone bigram probability $\pi(\phi^\ell|\theta) = \prod_{m=1}^{M}\pi(\phi^\ell_m|\phi^\ell_{m-1},\theta)$. By assumption, such information is not available from speech: we assume that there is no transcribed speech in the target language. A reasonable proxy, however, can be constructed from text.

Fig. 6 shows text data downloaded from wikipedia in Swahili, and a segment of a rule-based, character-by-character G2P for the Swahili language [18]. By passing the former through the latter, it is possible to

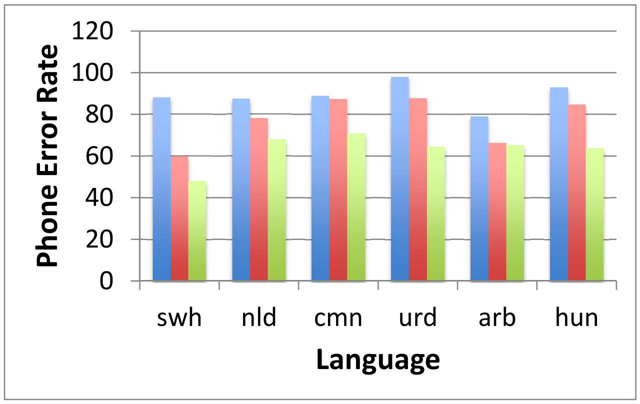

Figure 7: PER of the 1-best path: a measure of the quality of probabilistic transcriptions acquired from mismatched crowdsourcing. Native transcriptions were available in six languages: Swahili (SW), Dutch (DT), Mandarin (MN), Urdu (UR), Arabic (AR), and Hungarian (HG). Probabilistic transcriptions were decoded using three different methods per language: using a universal phoneme set (tallest bar in each language), using a phoneme set specific to the target language (middle bar in each language), and using a phonotactic language model derived from wikipedia texts (shortest bar in each language).

generate synthetic phoneme sequences in the target language.

Phone error rate of the 1-best path through the mismatched crwodsourcing confusion network are shown in Fig. 7. As shown, the use of a phonotactic language model, derived from wikipedia text, reduced phone error rate by about 10% absolute, in each language.

Composing $\mathbf{PT} \circ \mathbf{G}$ is complicated, however, by the presence of null transitions in the PT. A null transition in the PT matches a non-event in the language model, for which normal FST notation has no representation. In order to compose the PT with the language model, therefore, it is necessary to introduce a special type of "non-event" symbol, here denoted "#2", into the language model (Fig. 8). As shown in Fig. 8, a language model "non-event" is a transition that leaves any state, and returns to the same state (a self-loop). Such self-loops, labeled with the special symbol "#2" on both input and output language, are added to every state in the phonotactic language model (left-hand side of Fig. 8). The probabilistic transcript, then, is augmented with the special symbol "#2" as the input-language symbol for every null-output edge (output symbol is $\phi_m^\ell = \epsilon$).

4.4 Maximum *A Posteriori* Adaptation

Training from PTs can be improved by starting from a multilingual ASR, and adapting its parameters to PTs in the target language. The Bayesian framework for maximum *a posteriori* (MAP) estimation has been widely applied to GMM and HMM parameter estimation problems such as parameter smoothing and speaker adaptation [15].

Formally, for an unseen target language, we denote its acoustic observations $x = (x_1^1, \ldots, x_T^L)$, and its acoustic model parameter set as θ, then the MAP parameters are defined as:

$$\theta_{\text{MAP}} = \operatorname*{argmax}_{\theta} \pi(\theta|x) = \operatorname*{argmax}_{\theta} \pi(x|\theta)\pi(\theta) \tag{19}$$

where we use multilingual baseline GMM-HMM parameters to assign the conjugate prior hyperparameters in $\pi(\theta)$, and take the modes of the prior distributions as the initial model parameter estimates.

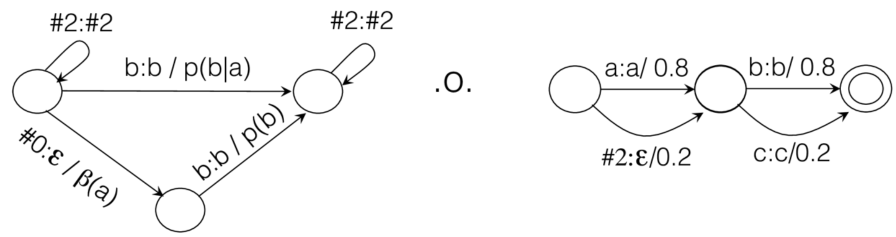

Figure 8: Deletion edges in the probabilistic transcript (edges with the special null output symbol, ϵ), required special handling in order to use information from a phonotactic language model. As shown, a new type of null symbol, "#2", was invented to represent the input for every PT edge with an ϵ output (right). Such edges were only allowed to match with state self-loops, newly added to the language model (left) in order to consume such non-events in the transcript.

In a GMM-HMM, the acoustic model is

$$\pi(x_t^\ell | s_t^\ell = j, \theta) = \sum_{k=1}^{K} c_{jk} \mathcal{N}\left(x_t^\ell | \mu_{jk}, v_{jk}\right)$$

whose parameters $\theta = \{c_{jk}, \mu_{jk}, v_{jk}\}$ include mixture weights, mean vectors, and variance vectors. Maximum likelihood trains these parameters by computing $\gamma_{\ell t}(j,k) = \pi(k|s_t^\ell = j, \phi^\ell, \theta)\pi(s_t^\ell = j, \phi^\ell, \theta)$, then accumulating weighted average acoustic frames with weights given by $\gamma_{\ell t}(j,k)$. Segmental K-means quantizes $\pi(s_t^\ell = j|\phi^\ell, \theta) \to \{0,1\}$ using forced alignment, then proceeds identically. MAP adaptation assigns, to each parameter, a conjugate prior $\pi(\theta)$ with mode equal to $\bar{\theta}$ (the parameters of the multilingual baseline), and with a confidence hyperparameter τ_θ, resulting in re-estimation formulae that are linearly interpolated between the baseline parameters $\bar{\theta}$ and the statistics of the adaptation data:

$$c'_{jk} = \frac{\tau_c \bar{c}_{jk} + \sum_{\ell,t} \gamma_{\ell t}(j,k)}{\sum_\kappa \left(\tau_c \bar{c}_{j\kappa} + \sum_{\ell,t} \gamma_{\ell t}(j,\kappa)\right)}, \quad \mu'_{jk} = \frac{\tau_\mu \bar{\mu}_{jk} + \sum_{\ell,t} \gamma_{\ell t}(j,k) x_t^\ell}{\tau_\mu + \sum_{\ell,t} \gamma_{\ell t}(j,k)}, \quad v'_{jk} = \frac{\tau_v \bar{v}_{jk} + \sum_{\ell,t} \gamma_{\ell t}(j,k)(x_t^\ell - \mu_{jk})^2}{\tau_v + \sum_{\ell,t} \gamma_{\ell t}(j,k)}$$

In our setting, the initial value of $\bar{\mu}_{jk}$ is obtained from the multilingual baseline model, and μ'_{jk} eventually converges to a model for the target language data.

4.5 Neural Networks

The NN acoustic model is

$$\pi(x_t^\ell | s_t^\ell = j, \phi^\ell, \theta) \propto y_t^\ell(j) = \frac{1}{c_j} \frac{\exp\left(w_j^T h_t(x, \theta_h)\right)}{\sum_k \exp\left(w_k^T h_t(x, \theta_h)\right)}$$

whose parameters $\theta = \{c_j, w_j, \theta_h\}$ include the senone priors c_j, the softmax weight vectors w_j, and the parameters defining the hidden nodes $h_t(x, \theta_j)$. NNs are trained by using a GMM-HMM to compute an initial senone posterior, $\pi(s_t^\ell = j, x^\ell | \theta)$, then minimizing the cross-entropy between the estimated senone posterior and the neural network output $y_t^\ell(j)$. The cross entropy is measured as

$$H(S^\ell \| Y^\ell) = -\sum_{t=1}^{T} \sum_j \pi(s_t^\ell = j) \ln y_t^\ell(j) \tag{20}$$

The gradient of Eq. (20) with respect to its model parameters is

$$\nabla_\theta H(S^\ell \| Y^\ell) = -\sum_{t=1}^{T} \sum_j \frac{\pi(j, x^\ell | \theta)}{y_t^\ell(j)} \nabla_\theta y_t^\ell(j) \tag{21}$$

Speech	Language Code (ISO 639-3)						
(# phones)	arb	yue	nld	hun	cmn	swh	urd
Train	32486	32693	27314	29461	28571	30009	21275
Dev	8208	6860	6943	7873	8244	7658	5808
Eval	8191	8638	6582	7474	7035	7441	3689

Table 1: Data statistics for seven SBS languages listing number of phones in the training/development/evaluation sets.

NN training with deterministic transcriptions is improved by quantizing $\pi(s_t^\ell, x^\ell|\theta) \rightarrow \{0,1\}$ using forced alignment. Preliminary experiments showed that forced alignment also improves the accuracy of NNs trained from probabilistic transcriptions: the best path through the PT, and the best alignment of the resulting senones to the waveform, were both computed using forced alignment. The resulting best senone string was used to train a DNN using Eq. (21).

5 Experimental Methods

Our goal is to train a phone recognition system for a given target language in which no native transcriptions are available. We assume that we have access to unspoken texts and to untranscribed audio in the target language, but not to transcribed audio (Sec. 5.1). Mismatched crowdsourcing was interpreted using a multilingual misperception G2P (Sec. 5.2), or using a language-specific misperception G2P estimated using EEG (Sec. 5.3). Baseline multilingual systems are trained using native transcriptions from several different languages not including the target language (Section 5.4). Finally, parameters of the acoustic model are adapted using PTs in the target language, derived from either self-training or from mismatched transcriptions (Sec. 5.5).

5.1 Data

Speech data were extracted from publicly available podcasts [53] hosted in 68 different languages. In order to generate test corpora (in which it is possible to measure phone error rate), advertisements were posted at the University of Illinois seeking native speakers willing to transcribe speech in any of these 68 languages. Of the ten transcribers who responded, six people were each able to complete one hour of speech transcription (the other four dropped out). One additional language was transcribed by workers recruited at I^2R in Singapore, yielding a total of seven languages with native transcriptions suitable for testing an ASR: Arabic (arb), Cantonese (yue), Dutch (nld), Hungarian (hun), Mandarin (cmn), Swahili (swh) and Urdu (urd).

The SBS radio podcasts are not entirely homogeneous in the target language and contain utterances interspersed with segments of music and English. A simple GMM-based language identification system was developed as a first pass over the podcasts in order to isolate regions that correspond mostly to the target language. These long segments were then split into smaller $\approx$ 5-second segments. This was to enable easy labeling by the native transcribers, and more importantly to allow for the collection of mismatched transcriptions that required the speech segments to be short. To further check that only speech clips in the target language were retained, the native transcribers were asked to omit any 5-second clips that contained music, significant amounts of noise, English speech or speech from multiple speakers. The resulting transcribed speech clips roughly amounted to 45 minutes of speech in Urdu and 1 hour of speech in the remaining six languages. The orthographic transcriptions for these clips were then converted into phonemic transcriptions using language-specific dictionaries and G2P mappings (these resources are detailed in Section 5.4). For each language, we chose a random 40/10/10 minutes split into training, development and evaluation sets. Table 1 describes the resulting training, development and evaluation sets.

5.2 Mismatched Crowdsourcing

Mismatched transcriptions were collected from crowd workers (Turkers) on Amazon Mechanical Turk [1] for all the data listed in Table 1. The crowdsourcing task setup is described in [28]. Each 5-sec speech

	Language (ISO 639-3 Code)						
	arb	yue	nld	hun	cmn	swh	urd
Dev set (1-best PER)	65.8	66.4	68.9	63.7	70.9	47.6	67.2
Eval set (1-best PER)	66.2	67.8	70.9	63.5	69.6	50.3	70.5

Table 2: Error rates (PER) of probabilistic transcripts computed from mismatched crowdsourcing (non-native human listeners): Phone error rate (PER) of the 1-best path through the probabilistic transcription, $\phi^* = \operatorname{argmax} \rho(\phi|T)$, development and evaluation sets.

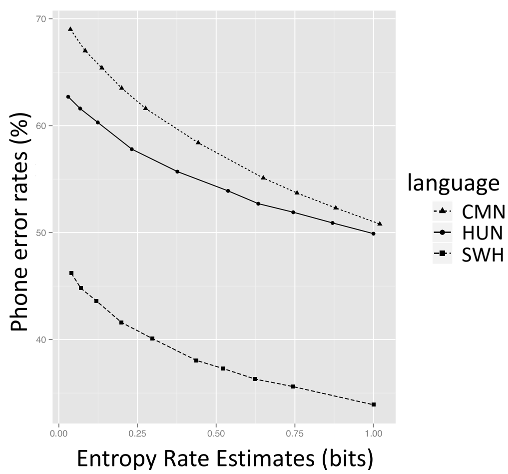

Figure 9: Phone error rates plotted against entropy rate estimates of phone sequences in three different languages.

segment was further split into 4 non-overlapping segments to make the non-native listening task easier. The crowdsourcing task was set up as described in [28]; briefly, the short segments were played to Turkers, who transcribed what they heard (typically in the form of nonsense syllables) using English orthography. Each short recording segment was transcribed by 10 distinct Turkers. More than 2500 Turkers participated in these tasks, with roughly 30% of them claiming to know only English. (Spanish, French, German, Japanese, Chinese were some of the other languages listed by the Turkers.)

A crude measure of the quality of the PTs is given by the phone error rate between $\phi^* = \operatorname{argmax}_\phi \rho(\phi|T)$ and the reference phone sequences. Table 2 lists these 1-best error rates on the SBS development and evaluation sets, for all seven languages. However, the 1-best error rates do not accurately reflect the extent of information in the PTs that can be leveraged during ASR adaptation. Consider, for example, the four Hindi phones [p,$\mathrm{p^h}$,b,b̤]. An attentive English-speaking transcriber must choose between the two letters <p,b> in order to represent any of these four phones. The misperception G2P therefore maps the letters <p,b> into a distribution over the phones [p,$\mathrm{p^h}$,b,b̤]. There is no reason to expect that the maximizer of $\rho(\phi|\lambda)$ is correct, but there is good reason to expect the correct answer to be a member of a short N-best list ($N \leq 4$ phones/grapheme). A fuller picture is therefore obtained by considering a collection of sequences ϕ that are almost as probable as ϕ^* according to our model. Figure 9 shows the trend of phone error rates (for three languages) obtained by using collections ϕ of increasing size, plotted against an entropy estimate of ϕ, e.g., 1 bit of entropy allows two equally probable choices for each phone in ϕ. We note that the phone error rates significantly drop across all languages, staying within 1 bit of entropy per phone, illustrating the extent of information captured by the PTs.

V_ϕ	$N_{M2O}(V_\phi)$	V_ϕ	$N_{M2O}(V_\phi)$	V_ϕ	$N_{M2O}(V_\phi)$
spa	0.862	yue	1.280	cmn	1.531
por	1.152	jpn	1.333	amh	1.844
nld	1.182	vie	1.393	hun	1.857
deu	1.258	kor	1.429	hin	2.848

Table 3: Frequency of many-to-one mappings $N_{M2O}(V_\phi)$ between phoneme inventory V_ϕ and the inventory of English. Languages are represented by their ISO 639-3 codes.

5.3 EEG Recording and Analysis

To compute distinctive feature weights used in estimating the misperception transducer as shown in Eqs. (3) and (4), recordings of cortical activity in response to non-native phones were made using EEG. Signals were acquired using a BrainVision actiCHamp system with 64 channels and 1000 Hz sampling rate. All methods were approved by the University of Washington Institutional Review Board.

Auditory stimuli used to evoke the electrocortical responses comprised consonant-vowel (CV) syllables representing three languages: English, Dutch and Hindi. The inclusion of only two non-English languages in the auditory stimuli was dictated by the relatively high number of repetitions required to achieve good signal-to-noise ratio from averaged EEG recordings. The choice of Dutch and Hindi was governed by their inclusion in the SBS subset used to train the misperception G2P as described in Section 3.2, and the relative similarities between their phoneme inventories and the phoneme inventory of English. In particular, we chose languages with relatively many (Hindi) or few (Dutch) "many-to-one mappings" between the non-English phoneme inventory and the phoneme inventory of English, estimated based on distinctive feature representations of the phonemes in each language (as given in the PHOIBLE database [43]). Such "many-to-one mappings" are expected to pose a problem for the non-native transcription task being modeled by the misperception transducer, so to test the limits of our design we chose languages that differed greatly in this property. Note that, although Hindi podcasts were not included in the SBS training data described in Section 5.1, colloquial spoken Hindi and Urdu are extremely similar phonologically [29], and considering that the auditory stimuli for the EEG portion of this experiment are simple CV syllables, it is reasonable to consider Hindi and Urdu as equivalent for the purpose of computing feature weights for the misperception transducer.

Language similarity was defined as the number of many-to-one mappings ($N_{M2O}(V_\phi)$) between the English phoneme inventory (V_ψ) and the non-native phoneme inventory V_ϕ. Using distinctive feature representations of the phonemes in each inventory (as given in the PHOIBLE database [43]), a many-to-one mapping was defined by finding, for each non-native phoneme ϕ, the English phoneme $\psi^*(\phi)$ to which it is most similar:

$$\psi^*(\phi) = \operatorname{argmin} \sum_k \delta\left(f_k(\psi) \neq f_k(\phi)\right) \tag{22}$$

The number of many-to-one collisions is then defined as

$$N_{M2O}(V_\phi) = \frac{1}{|V_\psi|} \sum_{\phi_1 \neq \phi_2} \delta\left(\psi^*(\phi_1) = \psi^*(\phi_2)\right) \tag{23}$$

where $|V_\psi|$ is the size of the English phoneme inventory. The frequency of many-to-one mappings is listed in Table 3 for several languages.

To construct the auditory stimuli, two vowels and several consonants were selected from the phoneme inventory of each language (18 consonants for English, 17 for Dutch, and 19 for Hindi). Choice of consonants was made so as to emphasize differences in the many-to-one relationships between English-Dutch and English-Hindi, while maintaining roughly equal numbers of consonants for each language. The consonants chosen for each language are given in Table 4; the vowels chosen were the same for all three languages (/a/ and /e/).

Two native speakers of each language (one male and one female) were recorded (44100 sample rate, 16 bit depth) speaking multiple repetitions of the set of CV syllables for their language. Three tokens of each unique syllable were excised from the raw recordings (24414 downsampled, RMS normalized) Recorded syllables had an average duration of 400 ms, and were presented via headphones to one monolingual American English

Language	Consonant phones used in the EEG experiment																	
eng	p	t	k	p^{h}	t^{h}	k^{h}	tʃ	tʃ^{h}	f	θ	ʃ	v	ð	z	m	n	l	ɹ	
nld	p	t	ɣ	p^{h}	t^{h}	k^{h}		tʃ^{h}	f		ʃ	v		z	m	n	l	ʀ	j
hin	p b	t̪ d̪ ʈ ɖ	k g	b$^{\text{ɦ}}$	t̪^{h} ʈ^{h} d̪$^{\text{ɦ}}$ ɖ$^{\text{ɦ}}$	k^{h} g$^{\text{ɦ}}$						ʋ			m	n̪ ɳ			

Table 4: Consonant phones used in the EEG experiment represented using IPA. Vertical alignment of cells suggests many-to-one mappings expected based on distinctive feature values from PHOIBLE.

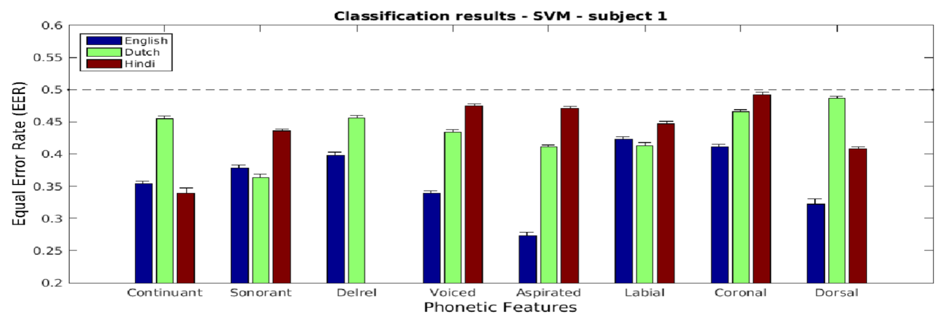

Figure 10: Classifiers were trained to observe EEG signals, and to classify the distinctive features of the phone being heard. Equal error rates are shown for English (the language used in training; train and test data did not overlap), Dutch, and Hindi. Dashed line shows chance=50%.

listener. The stimuli were presented in 9 blocks of 15 minutes per block, for a total of 135 minutes. Syllables were presented in random order with an inter-stimulus interval of 350 ms. 21 repetitions of each syllable were presented, for a grand total of 9072 syllable presentations.

EEG recordings were divided into 500 ms epochs beginning XXX ms before each syllable onset. The epoched data were coded with a subset of distinctive features that minimally defined the phoneme contrasts of the English consonants. Where more than one choice of features was sufficient to define those contrasts, preference was given to features that reflect differences in temporal as opposed to spectral features of the consonants, due to the high fidelity of EEG at reflecting temporal envelope properties of speech. [11] The final set of features chosen was: continuant, sonorant, delayed release, voicing, aspiration, labial, coronal, and dorsal.

Epoched and feature-coded EEG data *for the English syllables only* were used to train a support vector machine classifier for each feature. The classifiers were then used (without re-training) to classify the EEG responses to the Dutch and Hindi syllables. Fig. 10 shows equal error rates of these classifiers when applied to the three languages.

Eq. (3) defines a log-linear model of $\rho(\psi|\phi)$, the probability that a non-English phoneme ϕ will be perceived as English phoneme ψ. Denote by $\rho_U(\psi|\phi)$ the model of Eq. (3) with uniform weights for all distinctive features ($w_k = \alpha$, a tunable constant). Denote by $\rho_{EEG}(\psi|\phi)$ the same model, but with weights w_k derived from EEG measurements (Eq. (4)). Fig. 11 shows these two confusion matrices: $\rho_U(\psi|\phi)$ on the left, $\rho_{EEG}(\psi|\phi)$ on the right. The entropy of the uniform weighting, $\rho_U(\psi|\phi)$, is too low: when a Dutch phoneme ϕ has a nearest-neighbor $\psi^*(\phi)$ in English, then few other phonemes are considered to be possible confusions. $\rho_{EEG}(\psi|\phi)$ has a very different problem: since distinctive feature classfiers have been trained for only a small set of distinctive features, there are large groups of phonemes whose confusion probabilities can not be distinguished (giving the figure its block-diagonal structure). The faults of both models can be ameliorated by interpolating them in some way, e.g., by computing the linear interpolation $\rho_I(\psi|\phi) = a\rho_U(\psi|\phi) + (1-a)\rho_{EEG}(\psi|\phi)$ for some constant $0 \leq a \leq 1$.

5.4 Multilingual Baselines

The goal of building a multilingual system is two-fold. One is to setup a baseline for generalizing to an unseen language without any labeled audio corpus. The other is have the baseline serve as a starting point

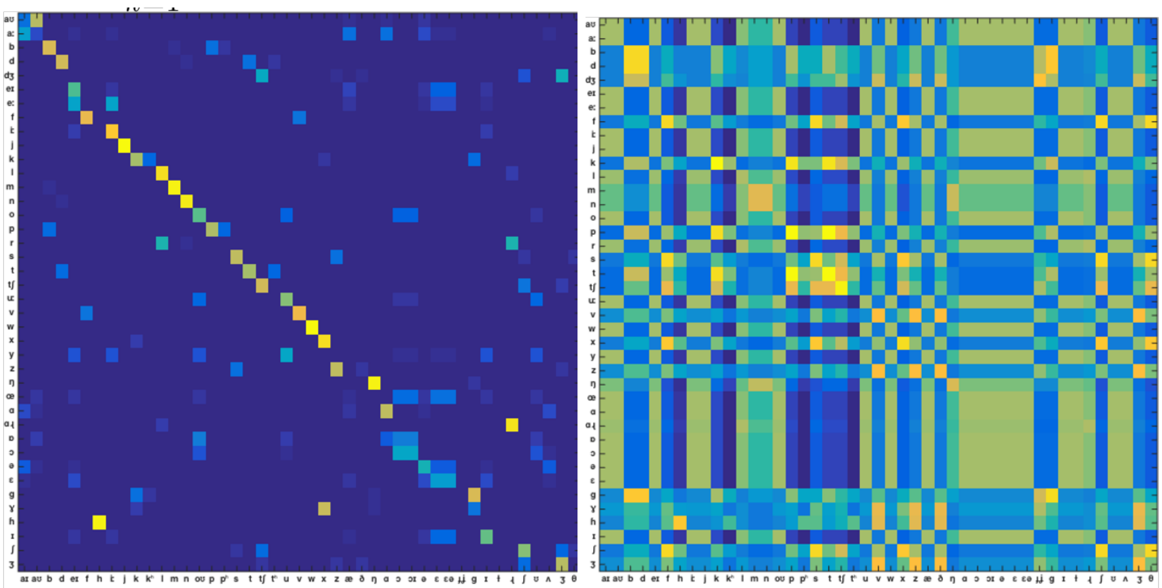

Figure 11: Phoneme confusion probabilities between English phonemes (column) and Dutch phonemes (row) using models in which the log probability is proportional to distance between the corresponding distinctive features. Left: all features have the same weight. Right: feature weights equal negative log error rate of EEG signal classifiers.

for adaptation.

The dataset consists of 40 minutes of labeled audio for training, 10 minutes for development, 10 minutes for testing for each language. The orthographic transcriptions are converted into phonetic transcriptions in the following steps. Beginning with a list of the IPA symbols used in canonical descriptions of all seven languages, any symbol appearing in only one languages was merged with a symbol differing by only one distinctive feature; this process proceeded until each remaining phone symbol is represented in at least two languages. English words are identified and converted to phones with an English G2P trained using CMU-dict [36]. We take the canonical pronunciation of a word if the word appears in a lexicon, otherwise estimate the word's pronunciation using a G2P. The Arabic dictionary is from the Qatari Arabic Corpus [14], the Dutch dictionary is from CELEX v2 [2], the Hungarian dictionary was provided by BUT [17], the Cantonese dictionary is from I^2R, the Mandarin dictionary is from CALLHOME [6], and the Urdu and Swahili G2Ps were compiled from rule-based descriptions of the orthographic systems in those two languages [18].

Each HMM was trained with data from six languages, tuned (hyperparameters) on the development set of the seventh language, and tested on the evaluation set of the seventh language. The lexicon of the target language was not used during testing, but two types of language-dependent specialization were allowed. In the first type of specialization, the universal phone set was restricted at test time to output only phones in the target language. In the second type of specialization, a target-language phone bigram language model was trained using phone sequences converted from text. The texts were collected from Wikipedia articles linked from the main page of each language crawled once per day over four months. Table 5 compares results using universal phone set and phone language model to those obtained using language-dependent phone set and phone language model. Without a language specific phone set and phone language model, it is hard for a multilingual system to generalize to an unseen language. This is true even if the system has seen closely related languages such as Mandarin when tested on Cantonese. As an oracle experiment, we also train language dependent HMMs for each individual language with 40 minutes of labeled audio. Results are shown in Table 5.

From the comparison of different baseline systems, we can reach the following conclusions. First, the standard speech pipeline is able to train speech recognizers using SBS data. Even with only 40 minutes of training data, a NN is able to outperform a GMM. Second, however, the standard speech pipeline generalizes well to languages that were seen in the training corpus, but performs poorly on unseen languages. Using a

data	acoustic model	language model	yue	hun	cmn	swh
multilingual	GMM	multilingual	79.64 (79.83)	77.13 (77.85)	83.28 (82.12)	82.99 (81.86)
multilingual	NN	multilingual	78.62 (77.58)	75.98 (76.44)	81.86 (80.47)	82.30 (81.18)
multilingual	GMM	text	68.40 (68.35)	68.62 (66.90)	71.30 (68.66)	63.04 (64.73)
multilingual	NN	text	66.54 (65.28)	66.08 (66.58)	65.77 (64.80)	64.75 (65.04)
monolingual	GMM	transcript	32.77 (34.61)	39.58 (39.77)	32.21 (26.92)	35.33 (46.51)
monolingual	NN	transcript	27.67 (28.88)	35.87 (36.58)	27.80 (23.96)	34.98 (41.47)

Table 5: PERs of unadapted multilingual systems on the evaluation sets along with monolingual systems. PERs on the development sets are in parentheses. Text-based language models are trained using phone sequences computed by applying a G2P to independent wikipedia texts in the target language. Transcript-based language models are trained using phone sequences computed by applying a G2P to native transcripts of the training data.

language-specific phonotactic language model gives significant improvement over the language-independent phonotactic model, but nevertheless significantly underperforms a system that has seen the test language during training.

5.5 MAP Adaptation to Probabilistic Transcripts

The baseline and the adapted models were implemented using Kaldi [46]. In order to efficiently carry out the required operations on the cascade $\mathbf{AM} \circ \mathbf{HCPT}$, we carefully designed $\mathbf{PT}$ as an acceptor defined as $\text{proj}_{\text{input}}(\widehat{PT})$, where $\widehat{PT}$ is a WFST mapping phone sequences to English letter sequences obtained as a cascade of WFSTs modeling the distributions shown in Equation 1, and $\text{proj}_{\text{input}}$ refers to projecting onto the input labels. For the purposes of computational efficiency, the cascade for $\widehat{PT}$ includes an additional WFST restricting the number of consecutive deletions of phones and insertions of letters (to a maximum of 3 in our experiments). We use two additional disambiguation symbols [42], apart from the ones used in typical Kaldi recipes, to determinize these insertions and deletions in $\widehat{PT}$. MAP adaptation for the acoustic model was carried out for a number of iterations (12 for yue & cmn, 14 for hun & swh, with a re-alignment stage in iteration 10).

6 Experimental Results

This section reports two types of results. First, subsection 6.2 reports improvements in the quality of probabilistic transcription using information acquired from EEG signals. Second, subsection 6.1 reports the accuracy of ASR trained using probabilistic transcription.

6.1 ASR Trained Using Probabilistic Transcriptions

This section demonstrates that PT adaptation improves the generalization capability of multilingual ASR to an unseen target language. Adaptation to ASR-derived PTs (self-training) significantly reduces PER, as has been previously reported [57]. PTs derived from human mismatched crowdsourcing provide significant further PER reduction.

Table 6 presents phone error rates (PERs) on the evaluation (and development) sets for four different languages. Mult-L corresponds to the multilingual baseline error rates reproduced from Table 5. PERs of GMM-HMM systems are reproduced in rows 4-7; PERs of DNN-HMM systems are reproduced in rows 9-12.

ST refers to the NN-HMM multilingual baselines adapted with automatically generated PTs (ASR self-training). Self-training was only performed using DNN systems; no self-training of GMMs was performed. Asterisk denotes a system whose eval-set PER is significantly lower than the eval-set PER of the MULT-L baseline in the same row ($p < 0.05$, MAPSSWE test computed using the NIST `sc_stats` tool [45]). **(TODO: RUN THE TESTS, INSERT ASTERISKS)**

Language Code	Multilingual (MULT-L)	Self-training (ST)	Mult-L + PT adaptation		
			(PT-ADAPT)	% Rel. redn % over MULT-L	% Rel. redn % over ST
GMM-HMM					
yue	68.40 (68.35)		**57.20 (56.57)**	16.4 (17.1)	10.3 (9.2)
hun	68.62 (66.90)		**56.98 (57.26)**	16.9 (14.3)	10.2 (9.9)
cmn	71.30 (68.66)		**58.21 (57.85)**	18.4 (15.7)	10.3 (9.7)
swh	63.04 (64.73)		**44.31 (48.88)**	29.6 (24.6)	24.7 (18.4)
DNN-HMM					
yue	66.54 (65.28)	63.79 (62.46)			
hun	66.08 (66.58)	63.53 (63.50)	**xxx (58.4)**	xxx (12.3)	xxx (8.0)
cmn	65.77 (64.80)	64.90 (64.00)	**xxx (53.1)**	xxx (18.1)	xxx (17.0)
swh	64.75 (65.04)	58.76 (59.81)	**xxx (48.6)**	xxx (25.3)	xxx (18.7)

Table 6: PERs on the evaluation and development sets (latter within parentheses) before and after adaptation with PTs. Asterisk = Eval set PER significantly lower than the PER of the MULT-L baseline in the same row.

	mismatched crowdsourcing	feature-based	EEG-induced +feature-based
PER	70.43	69.44	68.61

Table 7: Comparison of PERs for probabilistic transcription for the Dutch on the evaluation set using different schemes to compute misperception G2Ps.

PT-ADAPT corresponds to PERs from multilingual speech recognition systems that have been adapted to PTs in the target language. We observe substantial PER improvements using PT-ADAPT over MULT-L across all four languages. We also find that PT adaptation consistently outperforms the ST systems for all four languages. The relative reductions in PER compared to both baselines are listed in the last two columns. This suggests that adaptation with PTs is providing more information than that obtained by model self-training alone. It is also interesting that we obtain significantly larger PER improvements with PTs for Swahili compared to the other three languages. We conjecture this may be partly because Swahili's orthography is based on the Roman alphabet unlike the other three languages. Since the mismatched transcripts also used the Roman alphabet, the PTs derived from them may more closely resemble the native Swahili transcriptions (from which the phonetic transcriptions are derived).

It is also useful to compare the performance of GMM-HMM systems (rows 4-7 of Table 6) with the performance of DNN-HMM systems (rows 9-12). In the MULT-L setting, an ASR trained using six languages is then applied to an unseen seventh language, without adaptation; in this setting, the DNN consistently outperforms the GMM (the difference is not statistically significant, because we have only ten minutes per language of training data, but it is consistent across languages). In the PT-ADAPT setting, either GMMs or DNNs are adapted using PTs in the target language. PT adaptation improves the performance of both types of ASR, but the DNN does not improve as much as the GMM.

6.2 Misperception Transducer Trained Using EEG

In order to evaluate the effectiveness of the EEG-induced misperception transducer we looked at the transcription accuracies for the Dutch language when performed using 1) mismatched crowdsourcing 2) feature-based misperception transducer computed using uniform weighting, $\rho_U(\psi|\phi)$ 3) EEG-induced transducer combined with the feature-based transducer, $\rho_I(\psi|\phi)$. To Combine the two transducers, the value of the parameter α was optimized on a separate development set. As shown in Table 7 phone error rates were respectively improved 1.5% and 2.5% relative to mismatched crowdsourcing respectively by using the feature-based and combined misperception transducers.

7 Discussion

Models of human neural processing systems have often been used to inspire improvements in machine-learning systems (for a catalog of such approaches and a warning, see [5]). These systems are often called neuromorphic, because the system is engineered to mimic the behavior of human neural systems. In contrast to that approach, our incorporation of EEG signals into ASR resonates with the Human Aided Computing approach used in computer vision. [51, 60] Together with our EEG work presented here, this class of approach represents a less explored direction for design of machine learning systems, whereby recorded neural data (rather than neuro-inspired models) are used as a source of prior information to improve system performance. Therefore, our work here suggests that, by thinking about the kinds of prior information required by a machine learning system, engineers and neuroscientists can work together to design specific neuroscience experiments that leverage human abilities and provide information that can be directly integrated into the system to solve an engineering problem.

DNN-HMM outperforms the GMM-HMM in all baseline conditions, but not when adapted using PTs. Preliminary analysis suggests that the DNN is more adversely affected than the GMM by label noise in the PTs. A DNN is trained to match the senone posterior probabilities ($\pi(s_t^\ell|x^\ell,\phi^\ell,\theta)$ computed by a first-pass GMM-HMM. Many papers have demonstrated that entropy in the senone posteriors is detrimental to DNN training, and that the senone posteriors should therefore be quantized ($\pi(s_t^\ell)\rightarrow\{0,1\}$) prior to DNN training. In PT adaptation, however, entropy is unavoidable, and quantizing the forced alignment doesn't necessarily help. Table 2 showed that the 1-best path through the PT is only correct for 29-49% of all phones, depending on language. There is good reason for this: the transcribers don't speak the target language, so they find some of its phone pairs to be perceptually indistinguishable! Future work will seek methods that can improve the robustness of DNN training in the face of label noise.

This paper has tentatively defined an "under-resourced language" to be one that lacks transcribed speech data. Other authors have proposed that if a language lacks transcribed speech, ASR can be initialized in that language by adapting a multilingual baseline trained on other languages. Other authors have proposed, and Table 6 confirms, that significant error reductions can be achieved using self-training: by automatically labeling speech in the target language, and adding the self-labeled data to the training set. Table 6 shows that further error rate reductions can be achieved using mismatched crowdsourcing: by asking non-speakers of the target language to write down what they hear, and by interpreting their nonsense orthography as information about the phonetic content of the utterances. The PER of mismatched crowdsourcing (Table 2) is almost as high as the PER of cross-language ASR (Table 5), but the information provided by mismatched crowdsourcing is superior to that provided by self-training in the sense that it trains a better ASR.

In a sense, though, all of the results presented in this article, and all results presented in every other article published on the subject of under-resourced ASR, are artificial and disingenuous: ASR is trained without deterministic transcripts, but is then tested by comparing its output to a deterministic transcript. In order to test ASR in a language that truly lacks deterministic transcripts, it is necessary to define an error metric that requires only PTs. For example, suppose we define MPER (minimum phone error rate) to be the string edit distance between the ASR output and the closest-matching string in the PT. MPER is ill-defined: depending on how many different types of speech perceptual errors are considered as possibilities, it is possible to create a PT in which even the most low-probability non-native perceptual error is still listed as a possibility. The best results are obtained by pruning the PT, thus if `1best` : FST $\rightarrow$ FST is an operator computing the one-best path through an FST, `prune` : FST, $\Re\rightarrow$ FST is an operator that prunes low-probability edges, and **E** is an FST computing string-edit distance, then a potentially useful error metric can be defined from PTs according to

$$\text{MPER}(\beta)=\text{cost}\left(\text{1best}\left(\text{1best}\left(\mathbf{ASR}\right)\circ\mathbf{E}\circ\text{prune}\left(\mathbf{PT},\beta\right)\right)\right) \tag{24}$$

where the pruning operation removes, from each slot m, any phone whose negative log probability $-\ln\rho(\phi_m^\ell)$ is higher than the minimum-cost path by a difference greater than β:

$$-\ln\hat{\rho}_{\Phi_m^\ell}(k)=\begin{cases}0 & \text{if }\ln\max_j\left(\frac{\rho_{\Phi_m^\ell}(j)}{\rho_{\Phi_m^\ell}(k)}\right)<\beta\\ \infty & \text{otherwise}\end{cases} \tag{25}$$

Three candidate measures are shown in Table 8. Each column shows the difference between PER or MPER

Error Metric	hun	cmn	swh	jap
PER_M $-\mathrm{PER}_A$	66.90-57.26 =9.64	68.66-57.85 =10.81	64.73-46.88 =15.85	
$\mathrm{MPER}_A(0.1)$ $-\mathrm{MPER}_M(0.1)$				
$\mathrm{MPER}_A(1)$ $-\mathrm{MPER}_M(1)$				
$\mathrm{MPER}_A(1000)$ $-\mathrm{MPER}_M(1000)$				

Table 8: PER computed using a native transcription (where available), and minimum PER computed using probabilistic transcriptions.

of two different systems: a multilingual ASR (column MULT-L in Table 6, called PER_M or MPER_M) and a PT-adapted ASR (column PT-ADAPT in Table 6, called PER_A or MPER_A). The second row is true PER, if known; we were unable to hire a Japanese transcriber, so no true PERs are available in the last column of the table. Remaining rows show MPER defined as the PER between the ASR output and the closest-matching path through the PT phone lattice. MPER(1000) performs very little pruning, and is therefore too small to be useful; MPER(0.1) usually prunes away all paths except the best path, and is therefore too sensitive to label noise in the PT. Of the measures tested, MPER(1) seems to correlate best with the true differences between PERs of multilingual and PT-adapted systems, though of course it is not possible to measure significance of the correlation with only three data points.

8 Conclusions

Transcriptions from Mismatched Crowdsourcing are very noisy. Nevertheless, ASR adapted using Probabilistic Transcriptions beats a multilingual ASR. Adaptation using PTs consistently provides substantial PER improvements over the multilingual GMM-HMM baselines for every language evaluated, which demonstrates that the PTs of target language can be effectively used to adapt a multilingual ASR system to the unseen target language, by exploiting the model-based MAP estimation approach. Also, we find PT adaptation also consistently outperforms the semi-supervised baselines, showing that adaptation with PTs posts more efficacy than the model self-training alone.

Errors in mismatched crowdsourcing are reduced using phonotactic language models, even if those must come from text.

EEG responses can be used to estimate confusion matrices. Entropy is lowest in native language, second lowest in a similar language, and highest in a dissimilar language.

9 Acknowledgments

The work reported here was started at JSALT 2015 in UW, Seattle, and was supported by JHU via grants from NSF (IIS), DARPA (LORELEI), Google, Microsoft, Amazon, Mitsubishi Electric, and MERL. Parts of this work were previously published in [37].

References

[1] Amazon. Mechanical Turk. `http://www.mturk.com`.

[2] R Baayen, R Piepenbrock, and L Gulikers. CELEX2. Technical Report LDC96L14, Linguistic Data Consortium, 1996.

[3] James Baker. The dragon system — an overview. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 23:24–29, 1975.

[4] V. Berment. *Méthodes pour informatiser des language des langues et des groupes de langues peu dotées*. PhD thesis, J. Fourier University, Grenoble, 2004.

[5] Hervé Bourlard, Hynek Hermansky, and Nelson Morgan. Towards increasing speech recognition error rates. *Speech Communication*, 18:205–231, 1996.

[6] Alexandra Canavan and George Zipperlen. CALLFRIEND egyptian arabic. Technical Report LDC96S49, Linguistic Data Consortium, 1996.

[7] Özgür Cetin. Unsupervised adaptive speech technology for limited resource languages: A case study for Tamil. In *Workshop on Spoken Language Technology for Under-Resourced Languages (SLTU)*, Hanoi, Vietnam, 2008.

[8] P. Charoenpornsawat, S. Hewavitharana, and T. Schultz. Thai grapheme-based speech recognition. In *Human Language Technologies Conference (HLT)*, 2006.

[9] George E. Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio, Speech and Language*, 20(1):30–42, 2012.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[11] Giovanni di Liberto, James A. O'Sullivan, and Edmund C. Lalor. Low-frequency cortical entrainment to speech reflects phoneme-level processing. *Current Biology*, 25(19):2457–2465, 2015.

[12] Van Hai Do, Xiong Xiao, Eng Siong Chng, and Haizhou Li. Context dependant phone mapping for cross-lingual acoustic modeling. In *International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 16–19, 2012.

[13] Mohamed Elmahdy, Mark Hasegawa-Johnson, and Eiman Mustafawi. Hybrid phonemic and graphemic modeling for arabic speech recognition. *International Journal of Computational Linguistics*, 3(1):88–96, 2012.

[14] Mohamed Elmahdy, Mark Hasegawa-Johnson, and Eiman Mustafawi. Development of a tv broadcasts speech recognition system for qatari arabic. In *The 9th edition of the Language Resources and Evaluation Conference (LREC 2014)*, Reykjavik, Iceland, 2014.

[15] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

[16] S. Gizaw. Multiple pronunciation model for Amharic speech recognition system. In *Workshop on Spoken Language Technology for Under-Resourced Languages (SLTU)*, Hanoi, Vietnam, 2008.

[17] František Grézl, Martin Karafiaát, and Karel Veselý. Adaptation of multilingual stacked bottle-neck neural network structure for new language. In *Proc. ICASSP*, pages 7704–7708, 2014.

[18] Mark Hasegawa-Johnson. WS15 dictionary data. Downloaded 9/24/2015 from http://isle.illinois.edu/sst/data/dict, 2015.

[19] Hynek Hermansky, Dan Ellis, and Sangita Sharma. Tandem connectionist feature extraction for conversational HMM systems. In *Proc. ICASSP*, Istanbul, Turkey, 2000.

[20] Jui-Ting Huang, Jing Li, Dong Yu, Li Deng, and Yifan Gong. Cross language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Proc. ICASSP*, 2013.

[21] Po-Sen Huang and Mark Hasegawa-Johnson. Cross-dialectal data transferring for gaussian mixture model training in arabic speech recognition. In *International Conference on Arabic Language Processing (CITALA)*, pages 119–122, 2012.

[22] David Imseng, Petr Motlicek, Hervé Bourlard, and Philip N. Garner. Using out-of-language data to improve an under-resourced speech recognizer. *Speech Communication*, 56:142 – 151, 2014.

[23] International Phonetic Association (IPA). International phonetic alphabet, 1993.

[24] R. Jakobson, G. Fant, and M. Halle. Preliminaries to speech analysis. Technical Report 13, MIT Acoustics Laboratory, 1952.

[25] Bing-Hwang Juang and Lawrence Rabiner. The segmental K-means algorithm for estimating parameters of hidden Markov models. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 38(9):1639–1641, 1990.

[26] Preethi Jyothi and Mark Hasegawa-Johnson. Acquiring speech transcriptions using mismatched crowd-sourcing. In *Proceedings of AAAI*, 2015.

[27] Preethi Jyothi and Mark Hasegawa-Johnson. Improving Hindi broadcast ASR by adapting the language model and pronunciation model using a priori syntactic and morphophonemic knowledge. In *Proc. Interspeech*, 2015.

[28] Preethi Jyothi and Mark Hasegawa-Johnson. Transcribing continuous speech using mismatched crowd-sourcing. In *Proceedings of Interspeech*, 2015.

[29] Yamuna Kachru. Hindi-Urdu. In Bernard Comrie, editor, *The world's major languages*, pages 399–416. Oxford University Press, New York, 1990.

[30] Stephan Kanthak and Hermann Ney. Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition. In *Proc. ICASSP*, 2002.

[31] Kevin Knight and Yaser Al-Onaizin. A primer on finite-state software for natural language processing. Downloaded 3/1/2016 from `http://www.isi.edu/licensed-sw/carmel/`, 1999.

[32] Tom Ko and Brian Mak. Eigentrigraphemes for under-resourced languages. *Speech Communication*, 56:132 – 141, 2014.

[33] S. Krauwer. The basic language resource kit (BLARK) as the first milestone for the language resources roadmap. In *Proc. International Workshop Speech and Coputer (SPECOM-2003)*, pages 8–15, Moscow, Russia, 2003.

[34] V.-B. Le and Laurent Besacier. Automatic speech recognition for under-resourced languages: Application to vietnamese. *IEEE Trans. Audio, Speech and Language*, 17(8):1471–1482, 2009.

[35] Kevin Lenzo. The CMU pronouncing dictionary. Downloaded 1/31/2016 from `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`, 1995.

[36] Kevin Lenzo. The CMU pronouncing dictionary. Downloaded 9/24/2015 from http://www.speech.cs.cmu.edu/cgi-bin/cmudict, 2015.

[37] Chunxi Liu, Preethi Jyothi, Hao Tang, Vimal Manohar, Rose Sloan, Tyler Kekona, Mark Hasegawa-Johnson, and Sanjeev Khudanpur. Adapting ASR for under-resourced languages using mismatched transcriptions. In *Proc. ICASSP*, 2016.

[38] Jonas Lööf, Christian Gollan, and Hermann Ney. Cross-language bootstrapping for unsupervised acoustic model training: rapid development of a Polish speech recognition system. In *Proceedings of Interspeech*, 2009.

[39] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer, Speech and Language*, 14(4):373–400, 2000.

[40] Aanchan Mohan, Richard Rose, Sina Hamidi Ghalehjegh, and S. Umesh. Acoustic modelling for speech recognition in Indian languages in an agricultural commodities task domain. *Speech Communication*, 56:167 –180, 2014.

[41] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16:69–88, 2002.

[42] Mehryar Mohri, Fernando Pereira, and Michael Riley. Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing*, pages 559–584. Springer, 2008.

[43] Steven Moran, Daniel R. McCloy, and Richard A. Wright. PHOIBLE: Phonetics Information Base and Lexicon Online, 2013.

[44] Nelson Morgan and Hervé Bourlard. Continuous speech recognition. *IEEE Signal Processing Magazine*, 12(5):25–42, 1995.

[45] D.S. Pallet, W.M. Fisher, and J.G. Fiscus. Tools for the analysis of benchmark speech recognition tests. In *Proc. ICASSP*, volume 1, pages 97–100, 1990.

[46] D. Povey, A. Ghoshal, et al. The Kaldi speech recognition toolkit. *Proc. of ASRU*, 2011.

[47] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana. On the use of a multilingual neural network front-end. In *Proc. Interspeech*, pages 2711–2714, 2008.

[48] Tim Schlippe, Sebastian Ochs, and Tanja Schultz. Web-based tools and methods for rapid pronunciation dictionary creation. *Speech Communication*, 56:101 – 118, 2014.

[49] Tanja Schultz and Alex Waibel. Experiments on cross-language acoustic modeling. In *INTERSPEECH*, 2001.

[50] Claude Shannon and William Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, 1949.

[51] Pradeep Shenoy and Desney S. Tan. Human-aided computing: utilizing implicit human processing to classify images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 845–854, Florence, 2008. ACM Press.

[52] Ke Chai Sim and Haizhou Li. Context sensitive probabilistic phone mapping model for cross-lingual speech recognition. In *Proc. Interspeech*, pages 2175–2718, 2008.

[53] Special Broadcasting Services Australia. Podcasts in Your Language. `http://www.sbs.com.au/podcasts/yourlanguage`.

[54] A. Stolcke, F. Grezl, M.-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri. Cross-domain and cross-lingual portability of acoustic features estimated by multilayer perceptrons. In *Proc. ICASSP*, 2006.

[55] P. Swietojanski, A. Ghoshal, and S. Renals. Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR. In *Proc. IEEE Workshop on Spoken Language Technology (SLT)*, 2012.

[56] Martha Yifiru Tachbelie, Solomon Teferra Abate, and Laurent Besacier. Using different acoustic, lexical and language modeling units for ASR of an under-resourced language – Amharic. *Speech Communication*, 56:181 – 194, 2014.

[57] Karel Vesely, Mirko Hannemann, and Lukas Burget. Semi-supervised training of Deep Neural Networks. In *Proceedings of ASRU*, 2013.

[58] Karel Vesely, Martin Karafiát, Frantisek Grezl, Marcel Janda, and Ekaterina Egorova. The language-independent bottleneck features. In *Proceedings of SLT*, 2012.

[59] Ngoc Thang Vu, Franziska Kraus, and Tanja Schultz. Cross-language bootstrapping based on completely unsupervised training using multilingual A-stabil. In *Proceedings of ICASSP*, 2011.

[60] Jun Wang, Eric Pohlmeyer, Barbara Hanna, Yu-Gang Jiang, Paul Sajda, and Shih-Fu Chang. Brain state decoding for rapid image retrieval. In *Proceedings of the 17th ACM international conference on multimedia*, pages 945–954. ACM Press, 2009.