Acquiring Speech Transcriptions Using Mismatched Crowdsourcing

Preethi Jyothi and **Mark Hasegawa-Johnson**
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
405 N. Mathews, Urbana, Illinois 61801

Abstract

Transcribed speech is a critical resource for building statistical speech recognition systems. Recent work has looked towards soliciting transcriptions for large speech corpora from native speakers of the language using crowdsourcing techniques. However, native speakers of the target language may not be readily available for crowdsourcing. We examine the following question: can humans unfamiliar with the target language help transcribe? We follow an information-theoretic approach to this problem: (1) We learn the characteristics of a noisy channel that models the transcribers' systematic perception biases. (2) We use an error-correcting code, specifically a repetition code, to encode the inputs to this channel, in conjunction with a maximum-likelihood decoding rule. To demonstrate the feasibility of this approach, we transcribe isolated Hindi words with the help of Mechanical Turk workers unfamiliar with Hindi. We successfully recover Hindi words with an accuracy of over 85% (and 94% in a 4-best list) using a 15-fold repetition code. We also estimate the conditional entropy of the input to this channel (Hindi words) given the channel output (transcripts from crowdsourced workers) to be less than 2 bits; this serves as a theoretical estimate of the average number of bits of auxiliary information required for errorless recovery.

1 Introduction

Recently, crowdsourcing has been explored as a valuable resource to speedily derive transcriptions for large speech databases, e.g. (Callison-Burch and Dredze 2010; Novotney and Callison-Burch 2010; Eskenazi et al. 2013). However, prior work has typically relied on the crowd workers being native speakers of the language in question. An interesting question to investigate is the possibility of recovering spoken words from *mismatched transcriptions*, i.e., transcriptions by crowd workers who do not know the language.

How does one hear (or rather, mishear) a foreign language? Indeed, the phenomenon of "cross-linguistic mondegreens" (mondegreen refers to the misinterpretation of a phrase as a result of it being nearly homophonic with the original) have captured the imagination of scholars and lay people alike.[1] On the face of it, recovering the original text from mismatched transcriptions seems like a very hard problem as crucial information about the original text is lost in such a transcription. However, even though no single individual can provide sufficient information to recover the foreign language transcriptions, using a crowd of workers might allow us to do so. We initiate a systematic inquiry into this question.

We formalize the problem of decoding from mismatched transcriptions as follows. We are given a corpus of speech in a foreign language and access to a crowd unfamiliar with that language. The crowd workers can be requested to listen to segments of this speech and provide us with English letter sequences that most closely match what they hear. The problem then is to recover a transcription of the original speech from these letter sequences.

Our main contributions in this work are:

- We follow an information-theoretic approach, by modeling the mismatched crowd transcribers as a noisy channel, along with using an encoding scheme (a repetition code) and a maximum likelihood decoding rule.
- We provide an information-theoretic analysis of how much information is lost in transmission through the mismatched channel; we derive a tight upper-bound for conditional entropy of the inputs given the outputs of the channel.
- We demonstrate the feasibility of our technique using an isolated word recovery task for Hindi – we predict transcriptions for isolated words in Hindi using mismatched transcriptions from crowd workers unfamiliar with Hindi. We successfully recover more than 85% of the words (and more than 94% in a 4-best list).

2 Noisy Channel Model for Mismatched Transcription

Mismatched transcription can be modeled as a noisy channel as shown in Figure 1. The input to the system is text in the foreign language, X. It is then encoded into speech,

[1] See, for example:
http://itre.cis.upenn.edu/~myl/languagelog/archives/005100.html
http://en.wikipedia.org/wiki/Homophonic_translation

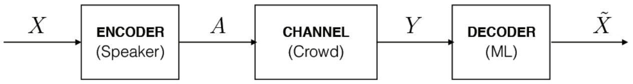

Figure 1: Noisy channel model for mismatched transcription

A, by a native speaker. This speech gets fed as input to the "crowd channel" which outputs an English transcription, Y. In a single invocation of the channel, one individual from the crowd listens to a segment of speech and writes down a sequence of English letters which is sent to the decoder. The decoder is considered to be successful if the output it produces, $\tilde{X}$, matches X. Ideally, we would like to use the maximum likelihood decoding rule[2] :

$$\begin{aligned}\tilde{x} &= \operatorname*{argmax}_x p(x|y) \\ &= \operatorname*{argmax}_x p(y|x) \cdot p(x) \qquad (1)\end{aligned}$$

In practice, one will use learned estimates $q(y|x)$ and $q(x)$ of $p(y|x)$ and $p(x)$, respectively. Even if we have accurate estimates of $p(y|x)$ and $p(x)$, this decoder cannot accurately recover x, since multiple values of x could be mapped to the same value of y. To get around this, one needs to resort to an error-correction code (Shannon 1948). However, in our case, we have limited freedom in designing this code as the original text, x, is not directly available for encoding: only the encoded speech, a, is available to us. Given this restriction, we resort to the use of a simple, repetition code illustrated in Figure 2. Note that the channel is invoked k times, once for each repetition of the speech segment, a, output by the encoder. Each of the k channel invocations could potentially involve a different individual transcriber from the crowd. In fact, we model the outputs $y^{(1)}, \ldots, y^{(k)}$, as identically and independently distributed (i.i.d.) samples produced by the crowd channel. The maximum likelihood decoder in this case outputs:

$$\begin{aligned}\tilde{x} &= \operatorname*{argmax}_x p(y^{(1)} \ldots y^{(k)}|x) \cdot p(x) \\ &= \operatorname*{argmax}_x p(x) \prod_{i=1}^{k} p(y^{(i)}|x) \qquad (2)\end{aligned}$$

Now, the maximum likelihood decoder can indeed recover the text message with probability tending to 1 as $k \to \infty$, as long as for every two distinct x and x' there is some y such that $p(y|x) \neq p(y|x')$.

An Information-Theoretic Analysis of the Channel

We can use information-theoretic tools to estimate how accurately the words can be decoded from mismatched transcriptions with the help of multiple labelers. Conditional entropy of the inputs (X) given the outputs of the channel (Y) captures the amount of information that is lost in transmission through the channel. We shall estimate an upper-bound of this conditional entropy using cross entropy, which can in turn be estimated from a corpus of data. We note that such an estimate can be computed even without an accurate model of the channel. However, the gap between conditional and cross entropy increases with the error in the model. Due to this gap, we run into the following issue.

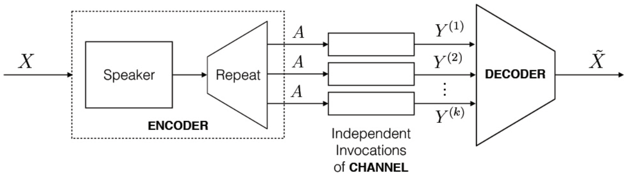

Figure 2: Noisy channel model for mismatched transcription with repetition coding

In our model, $q(y^{(1)} \ldots y^{(k)}|x) = \prod_{i=1}^{k} q(y^{(i)}|x)$ and hence, errors in our channel model $q(y^{(i)}|x)$ accumulate as k increases. As a result, the upper bound on conditional entropy becomes less tight with an increase in the number of repetitions. To counteract this effect, we propose augmenting the upper bound analysis using an auxiliary random variable. This random variable could intuitively be thought of as side channel information, indicating when the model needs to be corrected.

Let W be the auxiliary random variable, jointly distributed with (X, Y). Let $W = (Z, \hat{Z})$ where $\hat{Z}$ is a binary random variable which is a deterministic function of X, Y (discussed later) and $Z = \epsilon$, if $\hat{Z} = 0$ and $Z = X$, if $\hat{Z} = 1$. For brevity, we will use the notation p_0 and p_1 for $P(\hat{Z} = 0)$ and $P(\hat{Z} = 1)$, respectively.

Using $(Z, \hat{Z})$, conditional entropy can be upper-bounded as follows:

$$\begin{aligned}H(X|Y) &= H(X|Y, Z, \hat{Z}) + I(Z, \hat{Z}; X|Y) \\ &\leq H(X|Y, Z, \hat{Z}) + H(Z, \hat{Z}) \\ &= H(X|Y, Z, \hat{Z}) + H(\hat{Z}) + H(Z|\hat{Z}) \\ &= p_0 \cdot H(X|Y, Z, \hat{Z} = 0) \\ &\qquad + p_1 \cdot H(X|Y, Z, \hat{Z} = 1) + H(\hat{Z}) \\ &\qquad + p_0 \cdot H(Z|\hat{Z} = 0) + p_1 \cdot H(Z|\hat{Z} = 1) \\ &= p_0 \cdot H(X|Y, Z, \hat{Z} = 0) + H(\hat{Z}) \\ &\qquad + p_1 \cdot H(Z|\hat{Z} = 1) \\ &= p_0 \cdot H(X|Y, \hat{Z} = 0) + H(\hat{Z}) + p_1 \cdot H(Z|\hat{Z} = 1) \\ &= p_0 \cdot H(X|Y, \hat{Z} = 0) + H(\hat{Z}) + p_1 \cdot H(X|\hat{Z} = 1) \\ &\leq p_0 \cdot H(X|Y, \hat{Z} = 0) + H(\hat{Z}) + p_1 \log|\mathcal{X}| \qquad (3)\end{aligned}$$

where the first inequality follows from the fact that mutual information is non-negative. From the definition of $\hat{Z}$, we also have $H(X|Y, Z, \hat{Z} = 1) = 0$ and $H(Z|\hat{Z} = 0) = 0$. The last inequality is because $H(X|\hat{Z} = 1) \leq log|\mathcal{X}|$ where $\mathcal{X}$ is the alphabet of X.

[2] Here, $\tilde{x}$, x and y are values of the random variables $\tilde{X}$, X and Y, respectively.

Note that setting $\hat{Z} = 0$ always (i.e., $p_0 = 1$) has the effect of not using an auxiliary random variable which corresponds to the typical cross-entropy bound. We shall specify a better choice of $\hat{Z}$ after describing how the first term in Equation 3 is estimated.

To estimate the quantities in Equation 3, we rely on a data set with N samples (x_i, y_i), $i = 1$ to N. Without loss of generality, we assume that the first N_0 samples have $\hat{Z} = 0$ and the rest have $\hat{Z} = 1$. Then, p_0 is estimated as $\frac{N_0}{N}$. The first term in Equation 3 is estimated as:

$$\begin{aligned} H(X|Y, \hat{Z} = 0) &= \mathrm{E}_{(x,y)\sim X,Y|\hat{Z}=0}[-\log p(x|y, \hat{Z} = 0)] \\ &\leq \mathrm{E}_{(x,y)\sim X,Y|\hat{Z}=0}[-\log q(x|y, \hat{Z} = 0)] \\ &\approx \frac{1}{N_0} \sum_{i=1}^{N_0} -\log q(x_i|y_i, \hat{Z} = 0) \\ &\leq \frac{1}{N_0} \sum_{i=1}^{N_0} -\log q(x_i|y_i) \qquad (4) \end{aligned}$$

where the first inequality follows from the fact that cross-entropy is an upper bound on entropy. The second inequality is because for all $i \in [N_0]$, $q(\hat{Z} = 0|x_i, y_i) = 1$, which implies $q(y_i, \hat{Z} = 0|x_i) = q(y_i|x_i)$ and then

$$\begin{aligned} q(x_i|y_i, \hat{Z} = 0) &= \frac{q(y_i, \hat{Z} = 0|x_i)q(x_i)}{q(y_i, \hat{Z} = 0)} \\ &= \frac{q(y_i|x_i)q(x_i)}{q(y_i, \hat{Z} = 0)} \geq \frac{q(y_i|x_i)q(x_i)}{q(y_i)} = q(x_i|y_i) \end{aligned}$$

where $q(x_i|y_i)$ is computed using a standard invocation of the Bayes' rule:

$$q(x_i|y_i) = \frac{q(y_i|x_i)q(x_i)}{\sum_{x\in\mathcal{X}} q(y_i|x)q(x)}$$

where $q(y_i|x) = \prod_{j=1}^{k} q(y_i^{(j)}|x)$. Substituting Equation 4 in Equation 3, we get:

$$\begin{aligned} H(X|Y) \lessapprox \frac{1}{N} &\left((N - N_0)\log|\mathcal{X}| + \sum_{i=1}^{N_0} -\log q(x_i|y_i) \right) \\ &+ H_2(\frac{N_0}{N}) \qquad (5) \end{aligned}$$

where $H_2(p) = p \log \frac{1}{p} + (1 - p) \log \frac{1}{1-p}$ is the binary entropy function.

To complete the description of the bound in Equation 3, we need to define $\hat{Z}$. We recall that $\hat{Z}$ is a deterministic function of X, Y. By inspecting Equation 3 and Equation 4, a natural choice for $\hat{Z}$ is:

$$\hat{Z}(x, y) = \begin{cases} 1 & \text{if } -\log q(x|y) > \log|\mathcal{X}| \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

Indeed, if we ignored the term $H(\hat{Z})$ in Equation 3 and used the cross-entropy bound from Equation 4, then this choice of $\hat{Z}$ minimizes the bound.

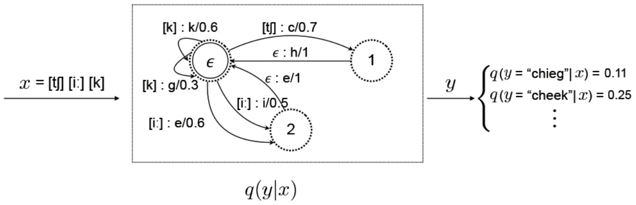

Figure 3: Noisy channel model, $q(y|x)$, implemented using finite state transducers.

3 Implementation Details

Both the channel and decoding algorithm are efficiently implemented using finite state transducers, as described below.

Estimating the Channel

We observe pairs of training instances, (x, y), where x refers to a phonetic representation of text in the foreign language and y is the corresponding English letter sequence produced by the crowd worker. We then apply maximum likelihood training using the Expectation Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) to estimate $q(y|x)$. Such models have been successfully used in machine transliteration problems (Knight and Graehl 1998).

The channel model is implemented using finite-state transducers (FSTs). Figure 3 shows a schematic diagram of our FST representation of the channel. The FST in Figure 3 probabilistically maps each phone in x to either a single English letter or a sequence of two English letters. A few mappings are shown in Figure 3 for purposes of illustration only; the FST model we build has transitions from every phone to every letter and two-letter sequence. The weights on the arcs of the FST model are negative log probabilities; these are learned using EM to maximize the likelihood of the observed data. We used the USC/ISI Carmel finite-state toolkit[3] for EM training and the OpenFst toolkit[4] (Allauzen et al. 2007) for all finite-state operations.

The channel model could be further expanded into a cascade of FSTs using additional domain-specific information. For example, the errors introduced by the speaker (the encoder in Figure 1) can be modeled using an FST based on distinctive features (Chomsky and Halle 1968) or articulatory features (e.g., Livescu 2005). However, in this work, we restrict ourselves to a simpler model and nevertheless obtain low error rates for our task of isolated word recovery. We leave it for future work to build more sophisticated channel models which might be required for more complex tasks like continuous speech recovery.

[3] http://www.isi.edu/licensed-sw/carmel/

[4] http://www.openfst.org

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$y^{(i)}$	poring	puda	raab	waatap	fotup	forup	puddop	pooda	puda	purap	poduck	purap	poodal	foora	foodap
IND	*reporting*	paḍā	rāhat	*voting*	*voter*	*foreign*	padtā	pūjā	paḍā	**pūrā**	kaḍak	**pūrā**	paḍtāl	*for*	*voting*
CUM	*reporting*	paḍegā	paḍegā	paḍegā	pratī	*foreign*	paḍā	**pūrā**	**pūrā**	**pūrā**	**pūrā**	**pūrā**	**pūrā**	**pūrā**	**pūrā**

Table 1: An illustrative example showing Turker transcriptions and decoded words for a Hindi word, **pūrā**. Row labeled $y^{(i)}$ shows the mismatched transcriptions from the Turkers. IND refers to decoding using only the transcription from the i^{th} Turker. CUM refers to decoding using the first i Turker transcriptions.

Decoding Rule

The maximum likelihood decoding rule, described in Equation (2), can be rephrased as:

$$\begin{aligned}\tilde{x} &= \operatorname*{argmax}_x q(x) \prod_{i=1}^{k} q(y^{(i)}|x) \\ &= \operatorname*{argmin}_x -\log(q(x)) + \sum_{i=1}^{k} -\log(q(y^{(i)}|x)) \qquad (7)\end{aligned}$$

In order to evaluate the channel model without a language model bias, we use a uniform distribution for X: i.e., $q(x)$ is identical for all x. For a given x, y, $-\log q(y|x)$ is approximated as the shortest path with input x and output y in the channel FST.[5] In order to accommodate for the high variability in the crowd channel, we may modify the decoding rule using a scaling function, $\mathcal{F}$ as follows:

$$\tilde{x} = \operatorname*{argmin}_x \sum_{i=1}^{k} \mathcal{F}(\log(q(y^{(i)}|x))) \qquad (8)$$

4 Experiments

Experimental Setup

We gained access to crowd workers using Amazon's Mechanical Turk (MTurk)[6] – an online marketplace where workers (Turkers) perform simple tasks (also called human intelligence tasks, or HITs) for nominal amounts of money. We chose Hindi as the foreign language in our experiments and requested only Turkers unfamiliar with Hindi to attempt the HITs. A total of 134 Turkers participated in our experiments. 119 of them provided information about the languages they are familiar with (apart from English). 73 Turkers listed no language other than English; Spanish was the most frequently listed language (21 Turkers), followed by French (9), Japanese (8), Chinese (5) and German (5); 12 other languages were listed by 3 or fewer Turkers.

We extracted Hindi speech from Special Broadcasting Service (SBS, Australia) radio podcasts[7] consisting of mostly spontaneous, semi-formal speech. Our data corpus comprised approximately one hour of speech selected from the above podcasts, containing speech from five interviewers totaling close to 10,000 word tokens. This speech was then manually transcribed at the phonetic level by a Hindi speaker. A training set and evaluation set were constructed using short segments of speech extracted from this corpus. It was ensured that segments of speech used in training did not overlap with any segments used in the evaluation set.

For the training set, we segmented all our data into short speech segments, approximately 1 or 2 seconds long. The reason for choosing short segments was to make the transcription task easier for the Turkers. Indeed, it was observed in a pilot experiment that longer speech segments tend to result in transcripts with large proportions of Hindi phones being deleted. For our evaluation data, we excised 200 isolated words from our 1 hour corpus, that were well-articulated. We created a vocabulary comprising all the words in our data, along with the 1000 most frequent words from Hindi monolingual text in the EMILLE corpus (Baker et al. 2002). The total size of our vocabulary was 2444 words.

For the Hindi speech in both the training and evaluation data sets, the Turkers were asked to provide English text that most closely matched what they heard; they were urged not to use valid English words as far as possible. We also conducted an experiment where the Turkers were specifically asked to provide only valid English words that most closely matched the Hindi speech.

Each word in the evaluation set was transcribed by 15 distinct Turkers. Thus, for each word itself, our i.i.d. assumption for the crowd channel is reasonable. However, there is a correlation across words since we allowed a single Turker to attempt multiple words. Nevertheless, this correlation is limited as the HITs completed quickly. And on average, each Turker provided 22 transcriptions (since 134 Turkers provided a total of 3000 transcriptions from 15 repetitions for 200 words).

Isolated Word Recovery Experiments

The isolated word recovery problem is a variant of the mismatched transcription problem, in which the inputs X correspond to isolated Hindi words drawn from a vocabulary $\mathcal{X}$. For this problem, the maximum-likelihood decoding rule (Equation 2) can be implemented by enumerating $q(y|x)$ for all $x \in \mathcal{X}$. As described before, when $y = (y^{(1)}, \ldots, y^{(k)})$, instead of $q(y|x)$ we can use $\sum_{i=1}^{k} \mathcal{F}(\log(q(y^{(i)}|x)))$ as in Equation 8. As the scaling function $\mathcal{F}$, we use the square root function, i.e., $\mathcal{F}(\alpha) = \sqrt{\alpha}$.

Table 1 gives an illustrative example of inputs from the Turkers and the output from our decoder. The second row ($y^{(i)}$) shows all fifteen Turker transcripts for a Hindi word,

[5] This is a good approximation, as is often the case in similar models, since one alignment tends to be far more likely than others.

[6] http://www.mturk.com

[7] http://www.sbs.com.au/podcasts/yourlanguage/hindi/

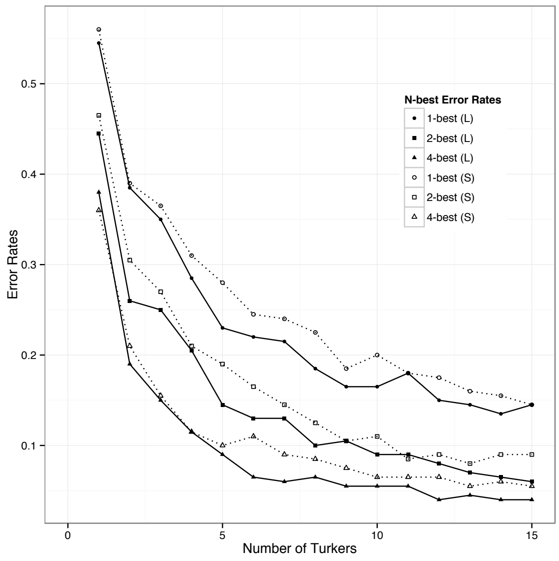

Figure 4: N-best error rates for varying number of Turkers and two different training set sizes. (L) and (S) indicate two different training set sizes.

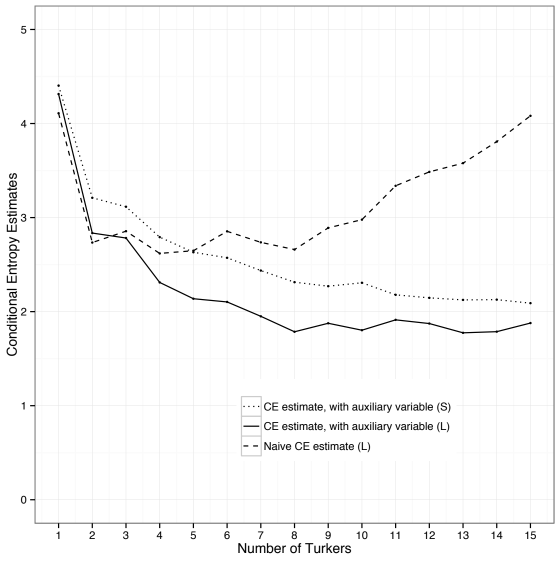

Figure 5: Estimates of conditional entropy (CE) of X given Y on the evaluation set for varying number of Turkers.

pūrā;[8] English words occurring in the vocabulary are italicized. The next row shows the result of decoding each of these Turkers individually. Note that only two of them result in the correct word (the words paḍā and *voting* also appear twice). The last row shows the result of decoding using the first i Turker transcriptions. Here, the correct word appears once we see a sufficiently large set of Turker transcriptions.

Figure 4 shows N-best error rates for channel models trained on instances from two different training sets: (S) corresponds to a training set comprising only the 1-second utterances and (L) consists of both 1-second and 2-second utterances. (S) and (L) contain 3992 and 7720 pairs of Hindi phone/English letter sequences, respectively. The N-best error rate is computed by considering how often the correct word appears within the top-N scoring words predicted by our channel model. We refer to the 1-best error rate simply as the error rate.

We see that the error rates consistently decrease with the increasing number of Turkers. We note that using transcripts from a single Turker results in an error rate of more than 50% whereas using all Turkers brings the error rate down to less than 15%. This plot exhibits a trend of diminishing returns with the first additional Turker giving the greatest decrease in error rates.

We note that if the words that appeared in the evaluation set have a large number of similar sounding words in the vocabulary, we can expect the task of word recovery to be harder. To quantify this inherent confusability, we consider word neighborhood statistics. Two words are said to be t-neighbors if one can be converted to the other using at most t edit operations (i.e., phone substitutions, insertions and deletions). For each word in the evaluation set, we compute the size of its t-neighborhood: on average, each word has 142 words that are 3 or less edit operations apart, indicating sizable neighborhoods of fairly confusable words.

In order to provide a benchmark for the error rates, we also describe an oracle baseline system. For every word in the evaluation set, let us assume an oracle provides its 1-neighborhood set i.e. the set of most confusable words in the vocabulary. The oracle baseline system chooses one of the 1-neighborhood words at random as the output word. Such a system, where the number of candidate words to choose from is considerably smaller, would still give us an error rate of 39.5% as opposed to our 1-best system that gives a 14.5% error rate.

Transcriptions using valid English words Since the mapping from English letters to phonemes is far from deterministic, the pronunciation that the Turker intended to communicate cannot be exactly determined by our decoding algorithm. One possible approach to circumventing this issue is to require the transcribers to use a phonetic alphabet. However, since Turkers cannot be expected to be familiar with such an alphabet, an alternative is to require the Turkers to provide transcripts in the form of one or more *valid* English words that most closely matched the Hindi speech. This ensures that the Turkers have more accurate information about how their transcripts will be interpreted. On the other hand, the Turkers may not be able to find a valid English word that is close to what they perceive, and will be forced to provide words which lose valuable information.

We experimentally determined the effect of this restriction, by repeating our entire experiment, but requiring the Turkers to use only valid English words in their transcripts. The English words were then mapped to their respective pronunciations using the CMU-dictionary (Weide 2007).

[8] We use the International Alphabet of Sanskrit Transliteration (IAST) for the Hindi text.

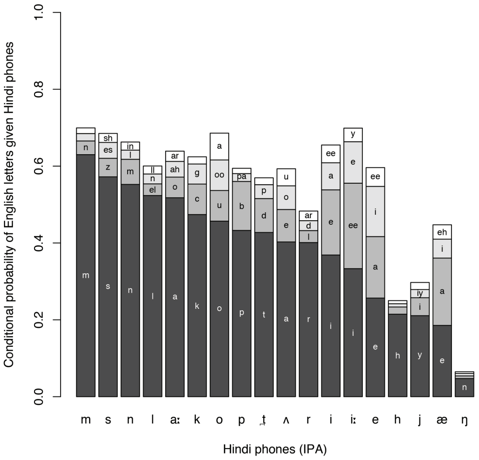

Figure 6: Hindi sounds (phones labeled using IPA symbols) with probabilistic mappings to English letter sequences.

The channel model was estimated using EM as described in Section 3, except y was set to be English phone sequences instead of letter sequences. The 1-best error rate increased to 30% in this experiment (from 14.5% when allowing English nonsense syllables). This suggests that the effect of losing information by restricting the transcription sequences (to valid English words) overshadows the advantage of a more deterministic phonetic interpretation of the Turker transcripts.

Conditional Entropy Estimates

As described in Equation 5 of Section 2, we proposed a generalized upper-bound estimate for conditional entropy (CE) using an auxiliary random variable. This can be used to get tighter upper bounds than a straightforward cross-entropy bound. Figure 5 illustrates the significance of the tighter upper-bound: the naive upper-bound increases (after an initial drop) with increasing number of Turkers, contrary to the fact that more information from larger number of Turkers can only reduce the conditional entropy. This is because, as mentioned in Section 2, the cross-entropy bound degrades as the gap between the channel and its model increases, which in turn increases as the number of Turkers increase. On the other hand, our upper bounds using the auxiliary random variable (defined by Equation 5 with $\log |\mathcal{X}| \approx 11.26$ and Equation 6) do not suffer from this artifact and are clearly tighter, as seen in Figure 5.

Channel Characteristics

Figure 6 visualizes the main probabilistic mappings from Hindi phones (labeled using the International Phonetic Alphabet, IPA) to English letter sequences, as learned by EM with a uniform initialization. We only show Hindi phones with 1000 or more occurrences in the training data. Mappings with conditional probabilities less than 2% are omitted, along with phone deletions. This plot reveals some fairly systematic patterns of mismatch. For example, unaspirated voiceless stop phones in Hindi such as "p" and "k" were sometimes perceived to be their voiced English counterparts, "b" and "g", respectively. Voiceless stops in Hindi are unaspirated but are typically aspirated in word-initial or stressed syllables in English. This causes them to be confused for their voiced counterparts when Hindi speech is transcribed by English speakers.

Phone Classes (in Hindi)	Conditional Entropy (in bits)
All phones	2.90
All vowels	3.05
All consonants	2.79
Consonants also in English	2.67
Consonants not in English	3.20

Table 2: Conditional entropy of English letters given Hindi phones for different phone classes, according to our model.

To quantitatively analyze the variability in the English letters given Hindi phones, we compute the conditional entropy $H(Y|X)$ where X is a single Hindi phone and Y is an English letter sequence of length 1 or 2, according to the model. We do this for four different classes of phones shown in Table 2.[9] We see that for the class of Hindi consonants that do not appear in English, the conditional entropy is highest, suggesting that Turkers have higher uncertainty about transcribing these sounds. Conversely, the phone class of consonants that appear both in Hindi and English has the lowest conditional entropy. The vowel class, compared to the class of all consonants, shows higher variability probably due to higher variability in spelling vowel sounds.

5 Discussion

In our current experiments, we restricted ourselves to an isolated word recovery task. However, our techniques are amenable to being extended to a continuous speech recovery task, with the help of a language model. Specifically, a language model implemented as a weighted FST can be incorporated within our decoder, instead of enumerating through a list of all hypotheses (Mohri, Pereira, and Riley 2002).

This work does not attempt to estimate the reliability of Turkers by exploiting information about the multiple tasks completed by the same Turker. While there has been prior work on reliable crowdsourcing (Karger, Oh, and Shah 2011; 2013; Vempaty, Varshney, and Varshney 2014) which exploits such information, it has focused on classification tasks with a small number of classes. In contrast, our setting of isolated word recovery involves a much larger label space, which becomes exponentially larger if we move to the continuous speech setting (described in the previous paragraph). Incorporating reliability estimates in these settings will require new techniques which we leave for future work.

There has been a significant amount of prior work in linguistics on language-specific perception errors, often within the context of second language learning (for some early

[9] A consonant in Hindi is considered to also be present in English if its IPA symbol appears in the English phonetic alphabet.

reviews of this work, see (Yamada and Tohkura 1992; Pisoni and Lively 1995; Akahane-Yamada 1996)). Interestingly, our approach suggests a way to study second language perception more broadly using noisy channel models.

Our generalized cross-entropy bound from Equation 5 serves as a quantitative measure of how well spoken words in one language can be communicated by mismatched transcribers in another language. This measure could be used to evaluate the effect of different features of the spoken words (e.g., read speech vs. spontaneous speech, fast vs. slow speech, speech in a tonal language vs. in a non-tonal language, etc) as well as the compatibility between the spoken language and the transcribers' native language. For example, one could expect tonal languages like Mandarin and Vietnamese to have a higher conditional entropy than Hindi for an English transcriber. This could be an interesting direction for future work.

6 Conclusions

This work establishes, for the first time, the possibility of acquiring speech transcriptions in a foreign language using crowdsourced workers unfamiliar with that language. On an isolated word recovery task, we obtain more than 85% accuracy using only the mismatched transcriptions. We also present an information-theoretic analysis of this problem, including a mechanism for estimating the amount of information lost in a mismatched transcription channel as a function of the number of transcriptions sought per utterance. There are many directions for future work, including extending our results to continuous speech recovery, experimenting with other languages and obtaining improved performance by estimating reliability statistics for the crowd workers.

We intend to scale our approach to cost-effectively create training data for developing speech technology in minority languages, using active learning algorithms (using partially trained recognition systems to determine which speech data need transcripts) and semi-supervised learning techniques (training a recognition system with an initial set of transcribed data and using it to acquire more labels).

7 Acknowledgements

This research was supported in part by a Beckman Postdoctoral Fellowship for the first author. The second author's contributions were supported by QNRF NPRP 09-410-1-069. The first author thanks Manoj Prabhakaran for useful discussions on the information-theoretic analysis part. The authors thank Lav Varshney for helpful comments on an earlier draft. The authors also gratefully acknowledge all the anonymous reviewers for their insightful suggestions and comments.

References

Akahane-Yamada, R. 1996. Learning non-native speech contrasts: What laboratory training studies tell us. In *Proceedings of ASA/ASJ*, 953–958.

Allauzen, C.; Riley, M.; Schalkwyk, J.; Skut, W.; and Mohri, M. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata (CIAA)*.

Baker, P.; Hardie, A.; McEnery, T.; Cunningham, H.; and Gaizauskas, R. J. 2002. EMILLE, A 67-million word corpus of Indic languages: Data collection, mark-up and harmonisation. In *Proceedings of LREC*.

Callison-Burch, C., and Dredze, M. 2010. Creating speech and language data with Amazon's Mechanical Turk. In *Proceedings of NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Chomsky, N., and Halle, M. 1968. *The sound pattern of English*. MIT Press.

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

Eskenazi, M.; Levow, G.-A.; Meng, H.; Parent, G.; and Suendermann, D. 2013. *Crowdsourcing for Speech Processing: Applications to Data Collection, Transcription and Assessment*. John Wiley & Sons.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *Proceedings of NIPS*.

Karger, D. R.; Oh, S.; and Shah, D. 2013. Efficient crowdsourcing for multi-class labeling. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and modeling of computer systems*.

Knight, K., and Graehl, J. 1998. Machine transliteration. *Computational Linguistics* 24(4):599–612.

Livescu, K. 2005. *Feature-based pronunciation modeling for automatic speech recognition*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Mohri, M.; Pereira, F.; and Riley, M. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech & Language* 16(1):69–88.

Novotney, S., and Callison-Burch, C. 2010. Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In *Proceedings of NAACL HLT*.

Pisoni, D. B., and Lively, S. E. 1995. Variability and invariance in speech perception: A new look at some old problems in perceptual learning. In Strange, W., ed., *Speech perception and linguistic experience: Issues in cross-language speech research*. Timonium, MD: York Press. 433–462.

Shannon, C. E. 1948. A mathematical theory of communication. *Bell System Technical Journal* 27:379–423, 623–656.

Vempaty, A.; Varshney, L.; and Varshney, P. 2014. Reliable crowdsourcing for multi-class labeling using coding theory. *IEEE Journal of Selected Topics in Signal Processing* 8(4):667–679.

Weide, R. 2007. The CMU pronouncing dictionary, release 0.7a.

Yamada, R. A., and Tohkura, Y. 1992. The effects of experimental variables on the perception of American English /r/and/l/by Japanese listeners. *Perception & psychophysics* 52(4):376–392.