Mark Hasegawa-Johnson*, Jennifer Cole, Preethi Jyothi and Lav R. Varshney

Models of dataset size, question design, and cross-language speech perception for speech crowdsourcing applications

DOI 10.1515/lp-2015-0012

Abstract: Transcribers make mistakes. Workers recruited in a crowdsourcing marketplace, because of their varying levels of commitment and education, make more mistakes than workers in a controlled laboratory setting. Methods for compensating transcriber mistakes are desirable because, with such methods available, crowdsourcing has the potential to significantly increase the scale of experiments in laboratory phonology. This paper provides a brief tutorial on statistical learning theory, introducing the relationship between dataset size and estimation error, then presents a theoretical description and preliminary results for two new methods that control labeler error in laboratory phonology experiments. First, we discuss the method of crowdsourcing over error-correcting codes. In the error-correcting-code method, each difficult labeling task is first factored, by the experimenter, into the product of several easy labeling tasks (typically binary). Factoring increases the total number of tasks, nevertheless it results in faster completion and higher accuracy, because workers unable to perform the difficult task may be able to meaningfully contribute to the solution of each easy task. Second, we discuss the use of explicit mathematical models of the errors made by a worker in the crowd. In particular, we introduce the method of mismatched crowdsourcing, in which workers transcribe a language they do not understand, and an explicit mathematical model of second-language phoneme perception is used to learn and then compensate their transcription errors. Though introduced as technologies that increase the scale of phonology experiments, both methods have implications beyond increased scale. The method of easy questions permits us to probe the perception, by untrained listeners, of complicated phonological models; examples are provided from the prosody of English and Hindi. The method of mismatched crowdsourcing permits us to probe, in more detail than ever before, the perception of phonetic categories by listeners with a different phonological system.

***Corresponding author: Mark Hasegawa-Johnson,** University of Illinois, Urbana, IL, USA, E-mail: jhasegaw@illinois.edu
Jennifer Cole: E-mail: jscole@illinois.edu, **Preethi Jyothi:** E-mail: pjyothi@illinois.edu, **Lav R. Varshney:** E-mail: varshney@illinois.edu, University of Illinois, Urbana, IL, USA

Keywords: crowdsourcing, rapid prosody transcription, second-language phonology, statistical learning theory

1 Overview

Crowdsourcing can be defined as the purchase of data (labels, speech recordings, etc.), usually on-line, from members of a large, heterogeneous, fluctuating, partially anonymous, and variably skilled labor market. The purchase of speech data and speech labels from crowdsourcing labor markets has the potential to significantly increase the scale of experiments that can be conducted in laboratory phonology, if experiments are designed to overcome the fundamental limitations of crowdsourcing. A few of the key limitations worth discussing include the severely limited control an experimenter has over the accuracy, training, and native language of workers hired to perform any given task. The goal of this article is to describe methods that can be used to compensate for crowd worker errors, for their possible lack of linguistic expertise, and for their possible inability to understand the language they are transcribing.

The experiments described in this paper were designed and conducted with frequent reference to two sources of information: the on-line tutorial guide for the crowdsourcing site used in this work, and the book *Crowdsourcing for Speech Processing: Applications to Data Collection, Transcription and Assessment* edited by Eskenazi et al. (2013). Material in the on-line tutorial provided the knowledge necessary to set up tasks, and to invite and pay workers. Useful material in Eskenazi et al. (2013) included methods for screening workers, choosing a fair rate of payment, explaining tasks to workers, and evaluating the quality of work already performed. Material in these two key references will not be repeated here, except as necessary to discuss error compensation, compensation for lack of training, and compensation for worker inability to understand the language being transcribed.

Crowdsourcing is partially anonymous, in the sense that tasks are usually connected to workers by way of an intermediate labor broker. The broker may be a company, a consortium of academics (e.g., GalaxyZoo), a non-profit foundation (e.g., SamaSource), or any other institution; it is also possible for an academic research project to solicit crowd workers directly, e.g., by advertising on appropriate mailing lists. When a broker is involved, the broker typically handles payment, so that employers and employees do not see one another's financial information.

The demographics of crowdsourcing seem to be primarily determined by the ability of brokers to pay workers, which, in turn, is determined by labor and financial regulations in the employer and employee countries of residence. Pavlick et al. (2014) tracked the IP addresses of workers responding to a task, then determined country of residence using reverse address lookup; they also asked each worker to provide a brief description of his or her working conditions. They found that workers were logged in from 86 different countries; the two most frequently reported countries were India and the United States. Workers in more-developed countries reported that they chose to work in crowdsourcing markets as a part-time job with scheduling flexibility. In less-developed countries, crowd workers reported being mostly full-timers, who treat crowdsourcing as a consulting job. Mason and Watts (2009) reported that the payment offered per task affects the speed with which tasks are performed (the number of workers who will perform the task, and the number of such tasks performed by each worker), but not the quality of the work performed. Our own experiments suggest that the result of Mason and Watts (2009) must be qualified. For example, the anonymity of the crowd labor market permits the existence of "spammers": workers who will enter data at random without any good-faith attempt to perform the assigned task. It is possible to avoid spammers by hiring only those workers with a good reputation (workers who have been assigned high scores by previous employers), but workers with high reputation are usually only willing to work for a reasonable wage.

The remainder of this paper is organized around a series of illustrative examples, which will be periodically made concrete. Suppose that you are trying to describe the sound system of the Betelgeusian language. Perceptual transcription studies, including your own, suggest that the Betelgeusian vowel inventory is characterized by formant frequencies, but also by some other yet-unknown acoustic features (possibly related to the fact that Betelgeusians have two heads; Adams 1979). We consider three approaches that might be used to identify acoustic correlates of the vowel categories in Betelgeusian. First, we consider the most controlled of the three proposed experiments, in which minimal pairs (words that differ only in the vowel) are recorded by cooperative informants. Under controlled conditions, the question of interest is dataset size: how many recorded examples are necessary in order to definitively describe inter-category differences in a certain pre-determined number of acoustic correlates? Second, we consider a somewhat less controlled experimental situation, in which words are recorded without advance knowledge of the vowel category in each word, and therefore some type of waveform labels are necessary. It is assumed that labels are solicited via crowdsourcing, but that there is no way to recruit crowd workers who speak Betelgeusian; therefore it is necessary to simplify the task so that it can be performed by workers who are not native speakers of Betelgeusian.

Finally, we consider the least controlled of the three experiments, in which short recorded phrases in the Betelgeusian language must be transcribed by crowd workers who speak no Betelgeusian. Even in the least controlled experimental situation, it is possible to recover a complete transcription, if one has side knowledge about the vocabulary of Betelgeusian and about the misperception of Betelgeusian vowels by English-speaking crowd workers.

2 Dataset size

As an illustrative example, suppose that you are trying to describe the sound system of the Betelgeusian language. You have decided to plan a trip to Betelgeuse, during which you will acquire recordings of men, women, and children producing each of the vowels in real words matching a list of target consonant contexts. Ultimately, you would like to identify the acoustic correlates that distinguish each vowel from every other. The question answered by this section is: how many examples of each vowel do you need to record?

The question of dataset size is actually two questions: (1) what measurements need to be acquired, and (2) how many training examples are necessary in order to accurately estimate the desired measurements? For example, consider the description of American English vowels by Peterson and Barney (1952). Peterson and Barney proposed that each vowel category, in American English, is characterized by its first and second formant frequencies (F1 and F2). Specifically, they proposed that each vowel category is described by average formant frequencies (μ_1 and μ_2), by the standard deviation of each of the two formants (σ_1 and σ_2), and by the correlation of the two formants (ρ). There are two ways in which a model of this type might be wrong. First, the set of measured parameters (two mean formant frequencies per vowel, two standard deviations per vowel, and one correlation coefficient per vowel) might not be sufficient to characterize the true difference between any pair of vowels. Second, even if the model is correct, the training data (the set of recorded examples from which the parameters are estimated) might be inadequate to estimate the model parameters. Barron (1994) called the first type of error "Approximation Error," and the second type "Estimation Error." He demonstrated that, for some types of models, Estimation Error is proportional to n/N, where n is the number of parameters that are being estimated, and N is the number of recorded training examples that are used to perform the estimation.

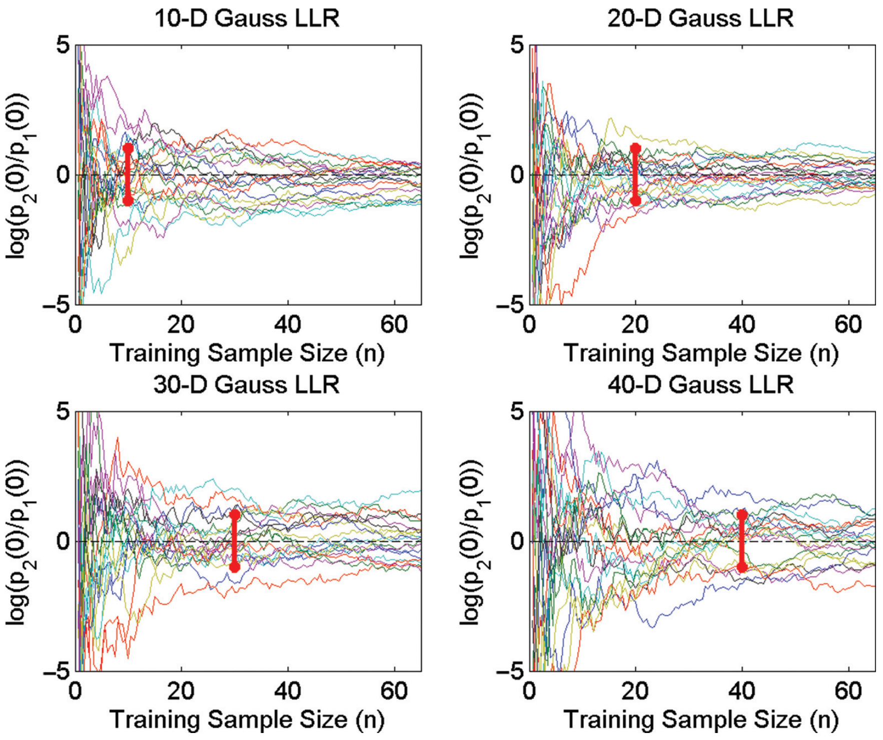


Figure 1: Output of a two-class Gaussian classifier (the kind of classifier used by Peterson and Barney 1952) whose input is a test token exactly halfway between the two classes. The classifier is normalized so that it should output "zero" in this case, but it does not, because the training database is drawn at random. The horizontal axis shows the size of the training database, N; the four subfigures show Gaussian classifiers with different acoustic measurement vector sizes, D. The red vertical bar in each plot crosses the abscissa at the value $N = D$, in order to exemplify the idea that the variability caused by randomly selected training data is controlled if one controls the ratio N/D. The "rule of 5" is a heuristic recommendation (a rule of thumb) suggesting that $N/D \geq 5$ is often a good choice.

Figure 1 shows an example of the way in which Estimation Error decreases with N. Suppose that we are trying to distinguish between two types of vowels that are distinguished by some unknown phonological distinctive feature. One of the two vowels is [+feature]; we will call this vowel +1. The other vowel is [–feature]; we will call this vowel –1. Since we do not know, in advance, what the feature is, the best thing we can do is to observe N examples of vowel +1, and compute an average spectrum μ_1. Likewise, we will observe N examples of vowel –1, and compute its average spectrum μ_{-1}. But since the training examples are recorded from a randomly selected group of people, there is some variability. In particular, suppose that we now record a vowel that is exactly halfway between +1 and −1, and we use the trained classifier to decide which of the two categories it belongs to. Since the vowel is (by assumption) halfway between the two categories, our classifier should produce the output '0', but since our classifier was trained using too few data, it produces an output that is not exactly '0'. Figure 1 shows the actual output of such a classifier, as a function of both N (the number of training examples) and n (the number of trainable parameters, which in this case is equal to D, the dimension of the spectral vector μ_1). There are 20 different curves on each set of axes, because the experiment was run 20 times per setting using random simulated training datasets. Notice that, as N gets larger, all of the different random experiments converge to the ideal solution, '0.' The vertical line segment in the middle of each subfigure is drawn when $N = n$, that is, when the number of training examples is equal to the number of trainable parameters: notice that the experimental conditions have been normalized so that when $n = N$, the Estimation Error of the classifier is exactly $n/N = 1$. Further details of the equations underlying Figure 1 are provided in Appendix A.

So how many training examples do you need? The answer is: it depends on how much Estimation Error you are willing to tolerate. Figure 1 is normalized so that the Estimation Error is exactly n/N, but there is usually some multiplier involved; for example, the normalized Gaussian classifier shown in Figure 1 will actually misclassify about 16% of all test tokens when $n/N = 1$, but only about 2% when $n/N = 1/5$. There are so many different types of classifiers for which $N \geq 5n$ is adequate that this ratio has been given a name: the "Rule of 5." The Rule of 5 says, simply, that in order to train a classifier with n trainable parameters, it is usually adequate to acquire $N \geq 5n$ training examples.

A Gaussian mixture model (GMM) is a model in which each vowel is allowed to have M different modal spectra, as shown in Figure 2. These different modes might represent different allophones of the same phoneme, or they might represent much smaller sub-categorical differences. For example, in English, the vowel /ɪ/ tends to take on the lip rounding features of the consonants surrounding it: the segment /ɪ/ extracted from the word *will* sounds (if removed from context) more like /ʊ/ than /ɪ/, yet when perceived in context, it is unquestionably an /ɪ/. In a GMM, each modal production is represented by its own Gaussian, with its own average formant frequency vector (or any other set of features, e.g., Davis and Mermelstein 1980 or Hermansky 1990), as shown in Figure 2. If each Gaussian mean is represented by D frequency samples, and if there are M modes per vowel, then there are a total of $n = MD$ trainable parameters per vowel. As shown in Figure 3, the

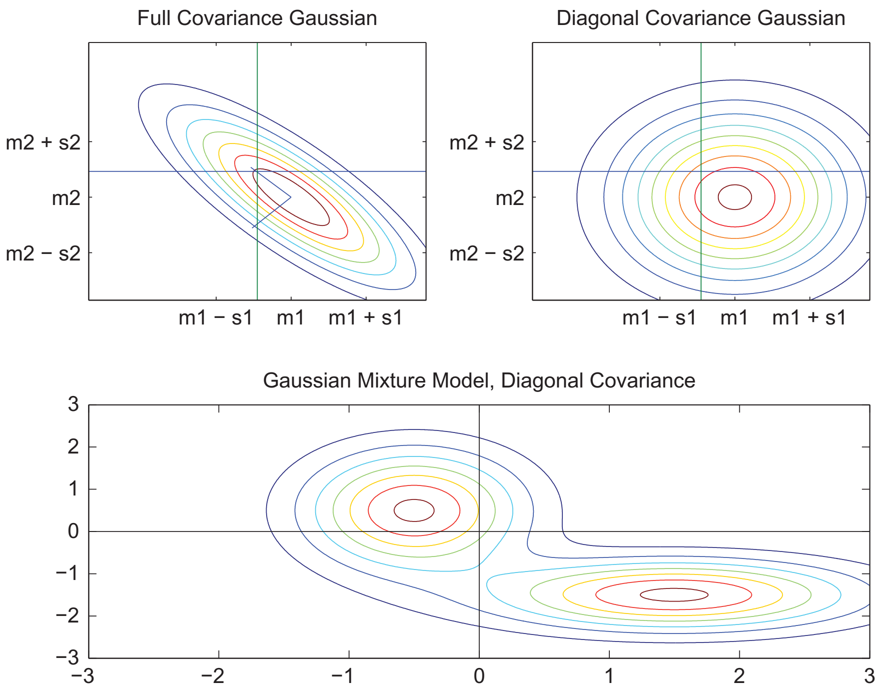


Figure 2: Contour plots of Gaussian and Gaussian mixture model (GMM) probability distributions. The horizontal and vertical axes, in each figure, are the first and second measurement dimension (e.g., F1 and F2). The contour plots show the probability distribution of tokens associated with some particular type, e.g., these might show the distribution of tokens associated with the vowel /i/. Top left: a Gaussian vowel category is characterized by the mean and variance of each feature, and by their covariance (correlation). Top right: a "diagonal covariance" Gaussian is one that ignores the covariance (assumes it equal to zero); this trick is commonly used to reduce the number of trainable parameters (decreasing Estimation Error), at the expense of reduced fidelity (increased Approximation Error). Bottom: a GMM is a distribution in which there are several different modal productions of the vowel, e.g., perhaps because the vowel tends to take on lip spreading vs. rounding from its surrounding consonants, as does /ɪ/ in English; the result is a vowel category that has several different modal productions, as shown here.

Estimation Error of a GMM is therefore proportional to $n/N = MD/N$, which is M times larger than the Estimation Error of a simple Gaussian model. The advantage of a GMM is its flexibility: by using M modes per vowel, it is possible to represent up to M subtly different modal productions of the vowel. The disadvantage is increased Estimation Error. If adequate training data are available, then a GMM is a better model; if the training database is too small, then a Gaussian is a better model.

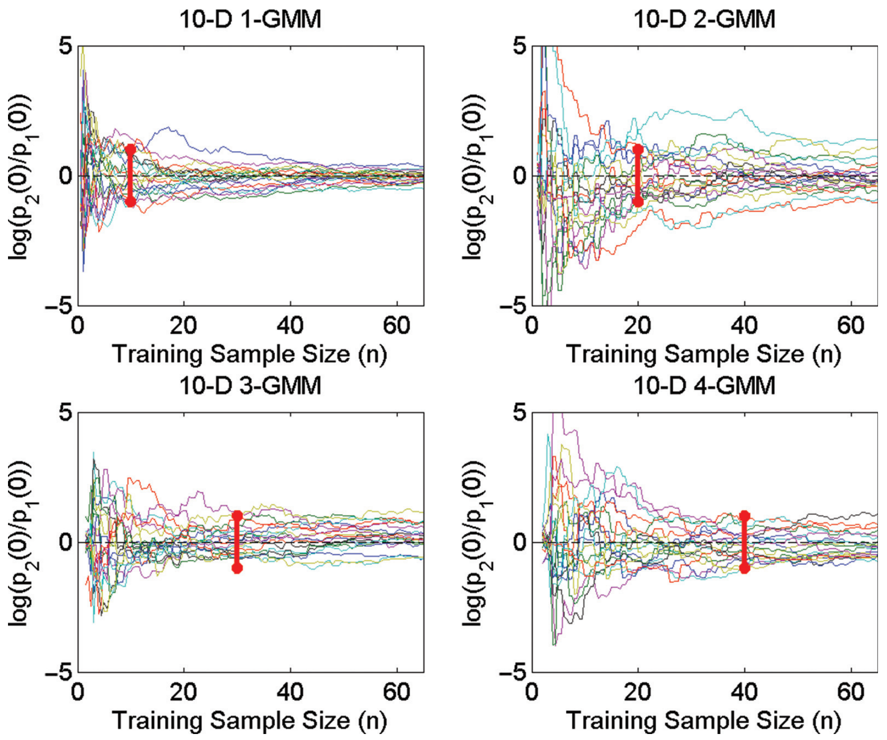


Figure 3: $h_\theta(0)$ as a function of N, 20 random trials per parameter setting, for a GMM classifier with M Gaussians per class. For a GMM, the total number of trainable parameters per vowel type is no longer just D (the measurement vector size): now it is MD (the measurement vector size times the number of Gaussians per class). As shown in the figure, randomness caused by random training data can be controlled if you make sure that N/MD is larger than a fixed constant. (The value $N/MD = 1$ is shown by the red bar in each figure, but $N/MD \geq 5$ is the value recommended in the "rule of 5.").

2.1 Active learning

Suppose you are studying a language with high front and high mid unrounded vowels – at least, that is what they sound like to you. You have singleton recorded examples of these two vowels as produced by one informant in /bVb/ context; they exhibit similar F1, but F2 of 2,200 Hz and 1,300 Hz, respectively. Suppose you decide to run a perceptual test, in which you will synthesize vowels on a continuum between these two exemplars, and ask your informant to label them, in order to estimate the location of the category boundary, and suppose that for some reason you need to know the boundary with a precision of 10 Hz. You could solve this problem by synthesizing 91 examples, $D = \{x_1 \ldots, x_N\}$, with $F_2(x_i) = 1300 + 10(i - 1)$, and asking your informant to label all 91. On the other hand, you could synthesize one example with $F_2 = \frac{1{,}300+2{,}200}{2} = 1,750$Hz, and ask your informant to label that one example; if she labels it as high front, then your next example will have $F_2 = \frac{1{,}300+1{,}750}{2} = 1,525$Hz, and so on. Using this quick-search algorithm, you can find the category boundary with 10 Hz precision by asking your informant at most 10 questions.

In the mathematical learning literature, the second algorithm is called "active learning" because the learner (you, and your computer that performs the synthesis for you) takes an active role in its own education. In most situations that are of interest in the real world, a supervised learning problem that would require on the order of N reference labels without interaction can be converted, by carefully designed active learning, into a problem that requires only on the order of $\log_2 N$ labels. A balanced binary tree with N terminal nodes has on the order of $\log_2 N$ levels; instead of labeling every terminal, active learning labels only the nonterminals on the path between the root of the tree and the terminal closest to the category boundary.

Active learning is often framed as an improvement over semi-supervised learning (SSL). Suppose that we have N labeled examples $((x_1, y_1), \ldots, (x_N, y_N))$ and U unlabeled examples (x_{N+1},..., x_{N+U}). Cohn et al. (1994) proposed training a classifier using N labeled data, using the same classifier to label the remaining U data, and estimating the confidence in each of the labeling decisions. The unlabeled datum with lowest confidence is then given to a human teacher, the teacher provides a label, and the classifier is re-trained. Beygelzimer et al. (2009) demonstrated that the order-log N label complexity of an active learner can be maintained even if the unlabeled data are not all available at the outset. In their algorithm, which they call "importance-weighted active learning," the unlabeled data are observed one at a time, or a few at a time, e.g., we might extract only a few seconds of data each day from a daily minority-language radio broadcast downloaded over the internet. Each unlabeled datum is assigned an importance weight between 0 and 1, based on the confidence with which the classifier is able to give it a label. A reasonable thing to do, at this point, would be to simply threshold the importance function, and ask humans to label every token with an importance greater than, say, 0.5. The problem with that approach is that it does not allow the importance-weighting function to make mistakes: if the importance-weighting function says that all vowels with low F1 and low F2 are obviously /ɪ/, and that such tokens have low importance because they are so obviously /ɪ/, then when examples of /ɑ/ somehow get mistakenly mixed into the database, there is no way we can detect them. In order to avoid this type of failure mode, Beygelzimer et al. (2009) recommend using a random number generator, which generates a random number between 0 and 1: if the random number is greater than the datum's estimated importance, then a human teacher is asked to provide a label.

Dasgupta (2011) more carefully outlined the limitations of active learning, by more strongly linking it to semi-supervised learning. The strong connection between active learning and order-log N binary search is only guaranteed if category labels are associated with compact, connected regions in feature space that have relatively smooth category boundaries. If any given category label claims discontinuous regions in feature space, then these discontinuous regions can only be detected by an active learning algorithm if it includes a semi-supervised learning component. As in SSL, the active learner must be able to predict the locations of category boundaries by observing structure in the evidence distribution of unlabeled data. In situations where every discontinuous category boundary is matched with at least one structural feature of the unlabeled data distribution, Dasgupta's integration of semi-supervised and active learning retains the order-log N label complexity of active learning.

Active learning has been used with great effectiveness in a large number of natural language processing applications (Olsson 2009); speech processing applications are less frequently reported (e.g., Douglas 2003). One example of the use of active learning is the "How May I Help You?" call routing application of Tür et al. (2005). In that work, a classifier was given a text transcript of the words with which a user responded to the question "How May I Help You?" The goal of the classifier was to correctly sort the phone calls into one of 49 available call categories. A classifier trained without active learning achieved linear error rate reductions: the error rate E scaled with the number of training data N as $E = 0.25 + 40/N$, achieving an optimum of $E_{min} = 0.26$ with $N = 40{,}000$ labeled data. A system trained using active learning achieved exponential error rate reductions (logarithmic label complexity): $E = 0.31e^{-N/7{,}400}$, achieving an optimum of $E_{min} = 0.249$ with $N = 20{,}000$ labeled data.

2.2 Crowdsourcing: labels for less

Crowdsourcing is a method for acquiring labels more cheaply by acquiring them from a large, heterogeneous, fluctuating, and variably skilled labor market. Theories of supervised, semi-supervised, and active learning apply *ceteris* *parabus* to crowdsourcing, except that crowd workers make mistakes. Indeed, reference transcriptions in linguistic tasks have always been known to contain mistakes, but most of the mathematical learning literature in the twentieth century chose to maintain the convenient fiction that human labelers are infallible. Crowdsourcing errors are more frequent, and therefore explicit models of label noise are a necessary part of any crowd-based methodology for science or technology development. Indeed, Novotney and Callison-Burch (2010) found that they could improve the accuracy of a speech recognizer by doubling the size of the training corpus, even if doubling the size also resulted in twice the transcriber error rate; apparently the benefits of extra data can sometimes outweigh the costs of extra error.

The speech technology development cycle is built on several assumptions. First, we assume that speech is perceived in terms of discrete phonological categories. We assume that labelers perceive those categories consistently, as long as labelers are drawn from a homogenous linguistic community. The requirement that labelers be drawn from a homogenous linguistic community results in labeling costs of at least 6 labeler hours per hour of transcribed speech (Cieri et al. 2004). Crowdsourcing methodologies eliminate most of the training and linguistic homogeneity requirements, thereby typically reducing the cost of labeling speech by a factor of three (Eskenazi et al. 2013) (see Table 1).

Table 1: For many decades, speech science and technology relied on transcriptions produced by academic experts (e.g., Zue et al. 1990). During roughly the years 2000–2009 it became typical, instead, to outsource labeling to a specialist consultant or consulting firm (Cieri et al. 2004). Crowdsourcing methodologies substantially reduce cost, at the expense of increased error (Eskenazi et al. 2013).

Source	**Motivation**	**Speed @ Wage**
Academic	High	$20\frac{\textit{transcriber hours}}{\textit{speech hour}}$ @ \$35/hour
Professional	High	$6\frac{\textit{transcriber hours}}{\textit{speech hour}}$ @ \$30/hour (Cieri et al. 2004)
Crowd	Variable	$600\frac{\textit{segments}}{\textit{speech hour}}$ @ \$0.1/segment (Eskenazi et al. 2013)

Quality of crowdsourced projects can be controlled at several stages: before, during, and after completion of the task (Parent 2013). Before data acquisition, manual quality control includes, e.g., choosing only workers with good reputation, whereas automatic quality control methods include, e.g., asking a gold standard question, and allowing to continue only those who pass. Quality control during data acquisition includes, e.g., majority voting. Quality control after data acquisition includes human intervention, e.g., asking other crowdsourcers to validate questionable input, or automatic methods, e.g., getting many responses to same question, comparing similarity using string edit distance, and eliminating outliers. It turns out that quality can be dramatically improved by anonymously pairing crowd workers, and by making payment dependent on criteria that encourage either explicit cooperation or explicit competition between paired workers (Varshney 2014).

Crowdsourcing can provide data cheaply, but crowdsourcers make mistakes. Majority voting reduces error, but triples (or worse) the cost. Majority voting is a simple process: assign the same task to *k* different crowdsourcers, and label the datum with the majority opinion. If each crowdsourcer is correct with probability *p*, then the probability that majority voting fails is less than or equal to the probability that no more than *k*/2 of the workers are correct. For example, Figure 4 shows probability of error as a function of *p*, for a three-person majority voting scheme. As shown, even with only three crowd workers per question, the probability that a majority voting scheme makes mistakes can be quite a bit lower than the probability of error of an individual crowd worker.

If each task is given to more than three crowd workers, then significantly reduced error rates can be achieved using weighted majority voting (e.g., Karger

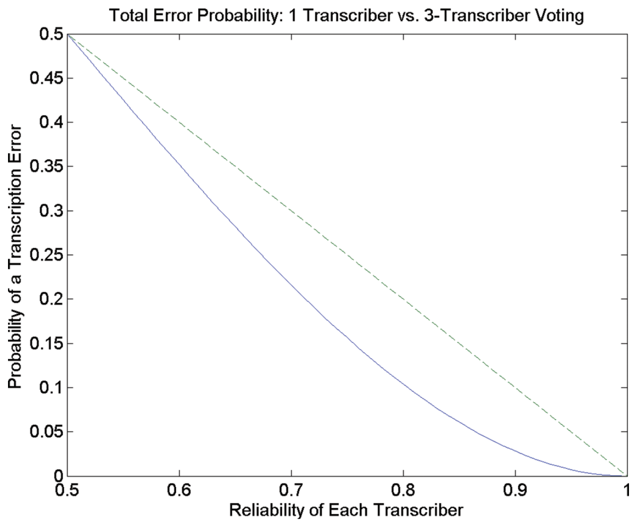


Figure 4: Probability of error versus the reliability of each coder, for a single coder (dashed line) and for a three-coder majority voting scheme (solid curve).

et al. 2011). In a weighted majority voting scheme, crowd workers return responses to a series of yes-no questions. Many different crowd workers answer each question. Since the true answer to each question is not known in advance, weighted majority voting computes an expected true answer, which is a real number between +1 (yes) and −1 (no). Each crowd worker's reliability is estimated by computing the average, across all questions, of the degree to which her answers match the expected true answer. The expected true answer, in turn, is computed as a weighted average of crowd worker answers, weighted by the reliability of each crowd worker. By iteratively estimating the true answers, then the worker reliabilities, then re-estimating the true answers, and so on, this algorithm can converge to a set of answers with significantly fewer errors than an unweighted majority voting scheme (Karger et al. 2011): there is no significant difference in accuracy with fewer than 5 workers answering each question, but the weighted voting scheme outperforms unweighted voting by an order of magnitude if at least 15 workers answer each question.

Majority voting and weighted majority voting significantly improve the probability of getting a correct transcription, but at extremely high cost: the cost is proportional to the number of workers who perform each task. Is majority voting worth the cost? Novotney and Callison-Burch (2010) found that training a speech recognizer using crowdsourced transcriptions degrades word error rate (WER) by 2.5%. Three-crowdsourcer majority voting results in transcriptions as accurate as professional transcriptions. Although the extra accuracy was helpful, it was not as helpful as having three times as much data: a speech recognizer trained with 20,000 words of crowdsourced transcriptions outperformed a system trained with 10,000 words of professional transcriptions, despite the significantly higher error rate of the crowdsourced transcriptions (similar results were found comparing 20k to 40k words, 40k to 80k, and 80k to 160k). Thus the benefit of extra data outweighed the cost of increased error.

3 Crowdsourcing with binary error correcting codes

There are, essentially, three ways to get transcribed speech data. First, one can prepare a list of words containing the speech sounds of interest, and ask cooperative informants each to produce an example of each word. Solicited productions can provide enough data to estimate the parameters of a reasonably simple statistical model, e.g., a model with no more than a few dozen trainable parameters (and therefore requiring no more than a few hundred solicited productions). In order to estimate a model with more parameters, it is usually necessary to record spontaneous speech (e.g., news broadcasts, storytelling, interviews, and conversations), and to transcribe it after the fact. The second standard method of acquiring transcriptions is by soliciting them from native speakers of the language being transcribed. Transcription by native speakers is possible if the language has an orthography, and if there are native speakers with computers who are willing to use the orthography to perform transcription. Most of the world's 7,000 languages do not have a standard orthography, and/or do not have a sufficiently large pool of internet-connected users who are willing and able to perform native-language transcription. When native-language transcription is not possible, all previously published research concludes that transcription is impossible, and that solicited productions are the only available method of inquiry. This paper proposes a method called "mismatched crowdsourcing," in which transcribers who don't understand the language are, nevertheless, asked to write what they hear. There are two important obstacles to mismatched crowdsourcing: the transcribers lack discrete phoneme categories matching those of the language they are transcribing, and they lack correct perceptual category boundaries. Incorrect perceptual maps are a problem similar to the problem of second language acquisition, and will be discussed in more detail in a later section. Lack of discrete phoneme categories, on the other hand, can be usefully compared to the lack of expertise in a scientific categorization ontology: category ontologies exist in many areas of science, and most of them are too complicated to be effectively used by non-expert transcribers. This section explores the problem of soliciting partial transcriptions from crowd workers who lack the expertise necessary to provide a full transcription.

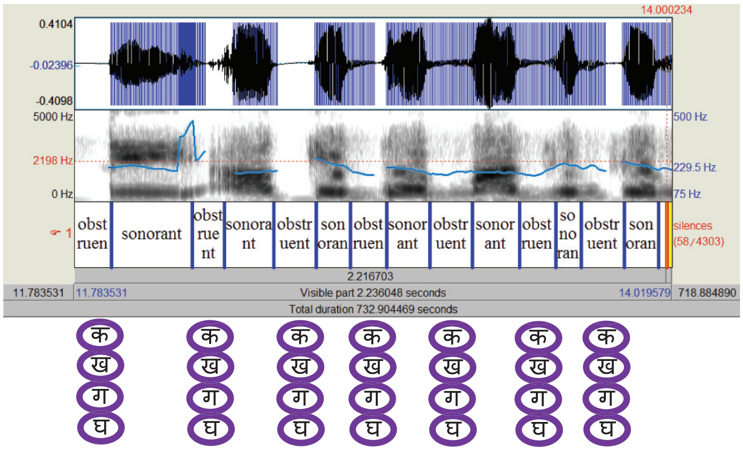
For linguistic tasks, the distinction between "easy" and "hard" tasks varies substantially depending on the background and training of the worker. Distinctions learned in elementary school are often considered "easy", while those learned in graduate school might be considered "hard". In many situations, age of acquisition has been demonstrated to be a surprisingly useful metric for estimating the difficulty that will be presented by any linguistic task to people who have no formal linguistic training; for example, Kim et al. (2010) found that average age of acquisition was the best predictor of production error rates in dysarthria. In order to obtain useful transcriptions of a language from people who do not speak the language, therefore, it may be useful to factor each vowel or consonant labeling task into several disctinctive feature labeling tasks, and to assign each distinctive feature labeling task only to workers whose native language includes a comparable distinctive feature.

Naturally, distinctive feature notation is unfamiliar to most non-linguists; therefore the tasks distributed to non-experts should be mapped into a notation that they have used since elementary school: the standard orthography of their native language.

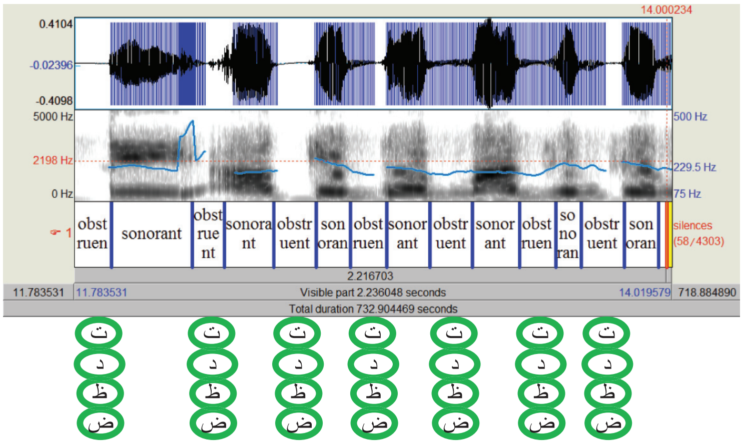
For example, suppose you have established that the Betelgeusian consonant inventory includes oral plosives with up to eight distinct categories at each place of articulation, apparently categorized as voiced versus unvoiced ([g] versus [k]), unaspirated versus aspirated/breathy ([k,g] versus [k^h,g^h]), and lax versus pharyngealized ([k] versus [$k^ʕ$]).

Hindi (and other Indic languages) includes a four-way categorization at each place of articulation, among the glottal features voiced versus unvoiced ([g] versus [k]) and unaspirated versus breathy ([k] versus [k^h]); every native speaker of Hindi learns to correctly label these four categories in kindergarten, using the four Devanagari symbols shown in Figure 5(a). Arabic, on the other hand, includes a four-way categorization marked by the features voiced versus unvoiced ([d] vs. [t]) and plain versus pharyngealized ([t] versus [$t^ʕ$]), and every native speaker learns to perform this labeling task in kindergarten using the Arabic symbols shown in Figure 5(b). In order to transcribe Betelgeusian, therefore, it might be wise to divide the transcription task into several sub-tasks, two of which are shown in Figure 5. Native speakers of Hindi can be solicited to label the glottal features of all plosives, using the Devanagari symbols for ‘ka’, ‘kha’, ‘ga’, and ‘gha’; the experimenter can then interpret each of these labels as a pair of binary distinctive feature labels transcribing the voicing and aspiration of each plosive. Native speakers of Arabic can be solicited to label the voicing and pharyngealization of all plosives, using the Arabic symbols for ‘ta’, ‘da’, ‘$t^ʕ$a’, and ‘$d^ʕ$a’; the experimenter can then interpret each label as a pair of binary distinctive feature labels transcribing voicing and pharyngealization. The final voicing label of each plosive is computed by a majority vote including all transcribers, both Hindi-speaking and Arabic-speaking. The final aspiration label is computed by a majority vote among Hindi-speaking transcribers, while the final pharyngealization label is computed by a majority vote among Arabic-speaking transcribers.

Factoring transcription into sub-tasks is likely to lead to faster, more reliable results, for two reasons. First, we have opened up the labor market for our task: rather than requiring native speakers of Betelgeusian, we are now free to recruit native speakers of Hindi and Arabic, and as shown by Pavlick et al. (2014), the crowd labor market includes many people with native proficiency in at least one of these languages. Second, and equally important, we have converted a difficult labeling task (one that is non-native for all available transcribers) into a series of easy labeling tasks. Each crowd worker listening to the sentence has



(a) Interface allows native speakers of an Indic language to click to label each automatically detected obstruent as unaspirated (क), aspirated (ख), voiced (ग),or voiced aspirated (घ).



(b) Interface allows native speakers of Arabic to click to label each automatically detected obstruent as unvoiced (ت), voiced (د), emphatic unvoiced (ظ), or emphatic voiced (ض).

Figure 5: Phonetic transcription of an unknown language is hard, but can be simplified by asking each transcriber to label only the distinctions that exist in his native language: (a) Hindi speakers easily label aspiration and voicing of stops; (b) Arabic speakers easily label voicing and pharyngealization (schematic only; the depicted user interface does not yet exist).

access to an auditory percept representing the complete transcription, but (if not a native speaker of Betelgeusian) he does not have the expertise to correctly label it. Instead, he has only the expertise to correctly label a sequence of several "easy questions": binary distinctive features, whose values his transcription can correctly label as either $a_j = +1$ or $a_j = -1$.

Errors are introduced because (i) transcriber attention occasionally wanders, and more importantly, because (ii) no two languages use identical implementations of any given distinctive feature. Variability is introduced when two distinctive features appear independently in Betelgeusian, but not in either transcriber language. For example, neither Arabic nor Hindi distinguishes the Betelgeusian phoneme pair [g^{h}] versus [g^{ʕh}].

Preliminary experiments suggest that a phoneme that does not exist in the transcriber's language is misperceived (mapped to symbols in his language) according to a probability distribution that is neither perfectly predictable (with zero error) nor uniformly unpredictable (with maximum error), but somewhere in between. It is therefore possible to talk about the error rate of the *j*th crowd worker's phone transcription: he labels binary distinctive features as though he believes any given phone should be given the *m*th possible phoneme label. Suppose that he has probability $1 - p$ of guessing the wrong phone label, where $1 - p$ is presumably rather high, because he is labeling speech in a language he does not know. Instead of asking the *j*th transcriber to provide a phoneme label, however, suppose that we interpret his transcription as if it provided only one binary distinctive feature label, a_j, which is either $a_j = +1$ or $a_j = -1$. The distinctive feature is wrong only if the *j*th transcriber has picked a phone label with the wrong value of a_j. The probability of this happening is the probability of error, $1 - p$, multiplied by the fraction of all errors that are wrong about this particular distinctive feature: thus the probability of any given distinctive feature is less than $1 - p$.

Partially correct transcriptions can be accumulated from many different crowd workers by letting c_{mj} represent the answer the *j*th worker should have given if hypothesis *m* were correct ($c_j = +1$ if the *m*th possible phoneme label is [+feature$_j$], $c_\mathrm{j} = -1$ if the *m*th possible phoneme label is [−feature$_j$], where feature$_j$ is the distinctive feature we extract from the *j*th transcription). The best phoneme transcription is then the transcription whose error-correcting code, c_{mj}, best matches the distinctive feature labels that were actually provided by the labelers. Redundancy in this way permits us to acquire more accurate transcriptions, because even a crowd worker who is wrong about every single phoneme is often, nevertheless, right about many of the distinctive features (Vempaty et al. 2014).

4 Binary coding answers scientific questions

The method of asking easy questions is useful not only because it improves accuracy, but also because it permits the experimenter to ask questions that could not otherwise be asked.

For example, suppose you would like to evaluate the prosodic characteristics of an utterance in a language for which the prosodic system has not yet been studied by linguists. This is a challenging task for two reasons. First, phrase-level prosodic features marking prominences and phrase boundaries are not consistently denoted in standard orthography for most languages, which means that transcribers from any native language background will not have a familiar vocabulary or symbol set with which to identify prosody in a given utterance, regardless of whether the target language is one they speak. A second problem is that the acoustic parameters that encode the prosodic features of a word (e.g., F_O, duration, intensity) vary as a function of many factors other than prosody, including the sex, age, and physiological state of the speaker, the speaker-selected style and rate of speech, and the local phonological and discourse context of the word (Cole 2015). These factors interact with the expression of prosodic features marking prominences and phrase boundaries, with the result that the acoustic cues to prosody are highly variable across utterances, both within and across speakers, making prosodic transcription challenging even for trained transcribers who are native speakers of the language being transcribed. In short, it can be difficult to obtain a prosodic transcription of an utterance using discrete prosodic features when there is imperfect or incomplete knowledge of the feature set and/or of the mapping between prosodic features and acoustic cues.

Linguistic analyses of prosody typically rely on prosodic transcriptions performed by trained experts using a transcription system grounded in phonological analysis. For instance, the Tones and Break Indices (ToBI) transcription system allows for each language or dialect an inventory of tones and break indices as categorical prosodic features (Beckman and Elam 1994; see Figure 6). The ToBI system is based on the Autosegmental-Metrical theory (Pierrehumbert 1981), which proposes discrete prosodic features that encode the syntactic, semantic, and pragmatic properties of the prosodically marked work. While experiments on prosody perception and production can probe the prosodic categories specified in a ToBI transcription for a given language, it is not possible to simply ask crowd workers to transcribe the tones and break indices they hear in speech audio; it takes considerable auditory training, and training in the interpretation of acoustic cues from visual displays of the speech waveform, spectrogram, and pitch track to produce reliable and consistent prosodic transcription. To our knowledge, prosodic transcription of this type is always carried

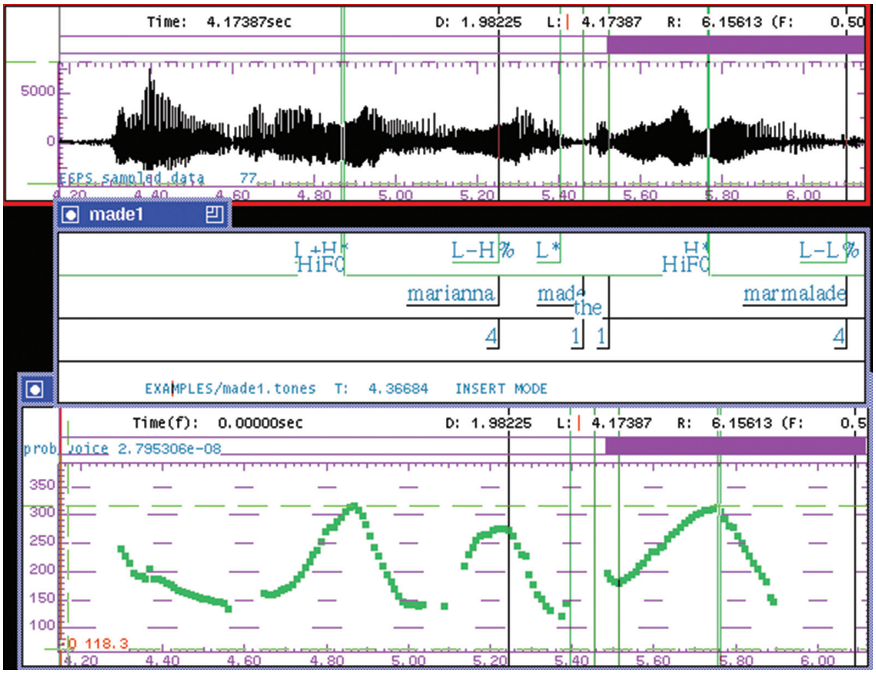
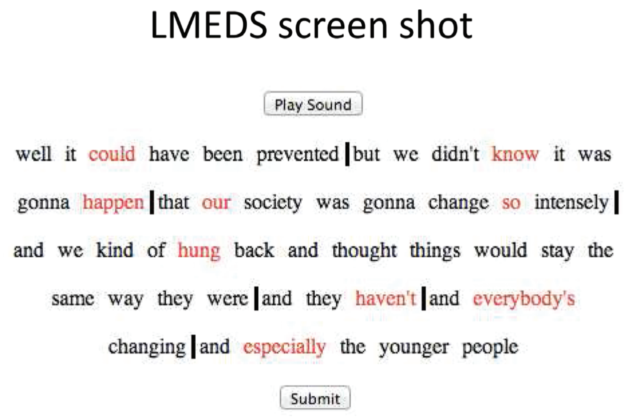


Figure 6: Prosodic transcription in the tones and break indices (ToBI) system, using a language-specific inventory of categorical tone features and break indices to mark words as prosodically prominent or at a prosodic phrase boundary, exemplified in the top labeling tier of this example from Beckman and Elam (1994).

out by transcribers who have native or near-native fluency in the target language. In addition to the requirement of training and native-like fluency, another bottleneck for obtaining prosodic transcription using ToBI or similar systems is transcription time, which can take anywhere from 10 to 100 times the duration of the audio file.

The system of Rapid Prosody Transcription (RPT; Cole et al. 2010a, b) was developed explicitly for the purpose of soliciting judgments of perceived prosodic features from linguistically untrained subjects. RPT expresses the hypothesis that every language user produces and perceives at least two prosodic distinctions: the distinction between prominent versus non-prominent words, and the distinction between phrase-boundary and non-boundary word junctures. In initial experiments, over 100 University of Illinois at Urbana-Champaign undergraduates without significant linguistic training were asked to perform prosodic transcription based only on their auditory impression, without reference to any visual display of the acoustic speech signal (Cole et al. 2010a, b). Transcription was intentionally


LMEDS screen shot

well it could have been prevented | but we didn't know it was gonna happen | that our society was gonna change so intensely | and we kind of hung back and thought things would stay the same way they were | and they haven't | and everybody's changing | and especially the younger people

Figure 7: Rapid prosody transcription example using the LMEDS interface (Language Markup and Experimental Design Software): Vertical bars indicate how the speaker breaks up the text into chunks (boundary). Red indicates words that are emphasized or stand out relative to other words (prominence).

coarse-grained: transcribers were given only simple definitions of prominence and boundary, and were instructed to mark words where they heard prominence or boundary (Figure 7).

The method is intentionally fast: Transcription is done in real time, with two listening passes per excerpt, based only on auditory impression. Because definitions of prominence and boundary are simplified, the RPT system can be used to solicit prosodic transcriptions from crowd workers. The Language Markup and Experimental Design software (LMEDS) was developed for this purpose, and has been used in recent experiments (Mahrt 2013).

RPT compensates for ambiguity in the definitions of prominence and boundary (because the words *prominence* and *boundary* are not understood as precise terms by most language users) through strength in numbers: groups of 15–22 subjects transcribe prosody for the same speech excerpts. The labels assigned by multiple transcribers are aggregated to assign each word in the transcript a Prominence Score (P-Score) or Boundary Score (B-Score) (Figure 8). The B-Score and P-score are each fractions between 0 and 1. The B-score is defined as the proportion of transcribers who marked a boundary following the word. Similarly, each word receives a prominence score (P-score) indicating the proportion of transcribers who marked the word as prominent.

Because the questions asked by RPT can be answered by untrained crowd workers, it is possible for RPT to ask questions of scientific interest that could not otherwise be asked. For example, the question RPT was initially developed

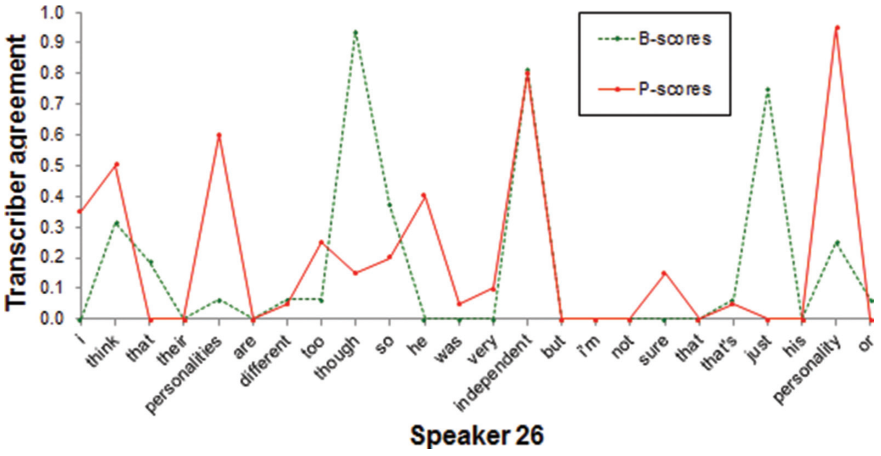


Figure 8: P-scores and B-scores: the fraction of crowd workers who label each word to have "prominence" or "boundary," respectively.

to ask: Can untrained transcribers produce reliable and consistent categorical transcriptions of prosody in conversational, spontaneous speech? Substantial evidence now indicates that the answer is yes; categorical prosodic features are perceived in ordinary conversational speech, even by language users with no formal linguistic training (Mahrt et al. 2011, Mahrt et al. 2012; Cole et al. 2014). Having established the validity of prosodic labels produced by untrained listeners, RPT has then been applied in order to probe the acoustic and textual correlates of perceived prosodic prominence and boundary, as heard by untrained listeners (examples of these findings for correlates are given in papers including Cole et al. 2010a, Cole et al. 2010b, Cole et al. 2014; Mahrt et al. 2011, Mahrt et al. 2012).

As another example of the types of scientific questions that can be answered using binary-encoded crowdsourcing, consider the problem of determining whether or not Hindi phrasal stress (prominence of one word within each prosodic phrase) exists, and if so, whether or not it is cued by any intonational correlates. The previously published literature expresses considerable disagreement about this question. While traditional pedagogical references on Hindi grammar (Kellogg 1938) already considered prosody, there have been several more recent studies exclusively focusing on specific aspects of prosody (Moore 1965; Ohala 1983, Ohala 1986; Harnsberger 1994; Nair 2001; Dyrud 2001; Patil et al. 2008; Genzel and Kügler 2010; Féry 2010; Puri 2013). Some of these later studies on Hindi intonation borrow insights from extensive work on the intonational phonology of Bengali (a closely related South-Asian language) (Hayes and Lahiri 1991, Hayes and Lahiri 1992; Fitzpatrick-Cole and Lahiri 1997). These works on Hindi intonation have found that there are systematic prosodic phrasing effects in Hindi. They have also uncovered specific consistent features, like the presence of a rising contour associated with every non-final content word. On the other hand, there is no consensus on several other aspects of Hindi prosody. For instance, though it is widely agreed that there is lexical stress in Hindi (Moore 1965; Ohala 1986; Harnsberger 1994; Nair 2001; Dyrud 2001), there is less agreement on the phonetic correlates of stress and the perceived placement of stress at the word level. As another instance, contradictory theories regarding the relation between prosodic prominence and pitch accents have appeared in prior work (Patil et al. 2008; Féry 2010; Féry and Kentner 2010; Genzel and Kügler 2010). In particular, based on the observation that Hindi speakers do not produce consistent pitch contours on stressed syllables, Féry and colleagues (Patil et al. 2008; Féry 2010; Féry and Kentner 2010) propose that Hindi uses phrasal tones to prosodically structure an utterance and does not have prominence-lending pitch accents.

The question of prominence and boundary perception in Hindi was studied in Jyothi et al. (2014) by recruiting 10 adult speakers of Hindi, and playing to them 10 narrative excerpts in the Hindi language (about 25 seconds each, from the OGI Multi-language Telephone Speech Corpus). Listeners were asked to mark (a) how the speaker breaks up the text into chunks (boundary), and (b) words that are emphasized or stand out relative to other words (prominence). Each listener coded (condition 1) half of the utterances with audio, and (condition 2) half without audio, in order to determine the extent to which responses depended on the text versus the audio (no punctuation was provided in either condition). RPT responses in Hindi were compared to two "reference" ToBI transcriptions. Since there is no standard ToBI transcription system for Hindi, both reference transcriptions were mismatched, but in slightly different ways. Condition 3 consisted of a ToBI transcription performed by a professional linguist, trained in the ToBI transcription of English, who is also a native speaker of Hindi. She assigned prosodic phrase boundary labels based on her knowledge of the acoustic correlates of phrase boundary in Hindi, but because there is no consensus about the status of phrasal prominence in Hindi, she did not similarly seek correlates of phrasal prominence. Instead, she used the "pitch accent" label L+H* to label any acoustically evident pitch rise that she perceived as a phonological gesture, including the pitch rises that are characteristic of most content words in standard Hindi production. Her definition of pitch accent was therefore, by design, incommensurate with the instructions given to subjects in the rapid prosody transcription (RPT) study. Condition 4 was incommensurate with the RPT labels in a different and more severe way: ToBI labels were generated for the same speech data using AuToBI (Rosenberg 2010), a program that is designed to automatically generate ToBI labels for English-language speech.

Agreement of the 10 RPT transcribers with one another, and with each of the two reference transcriptions, was measured using Fleiss' kappa (Figure 9). Agreements about the coding of phrase boundaries were moderate, supporting the claim that phrase boundary is perceptible in Hindi. Responses to audio stimuli showed moderate agreement with responses to text stimuli, supporting the claim that phrase boundaries can be accurately predicted from text. Responses of the 10 listeners without linguistic training show moderate agreement with ToBI labels produced by the expert linguist, suggesting that the acoustic cues described in the linguistic literature correlate well with the perceptions of untrained listeners. All human Hindi speakers (including both the 10 listeners without linguistic training, and the 1 listener with linguistic training) show moderate agreement with the scores produced by the English-language AuToBI system (which has no information about the Hindi text), suggesting that the acoustic correlates of phrase boundary in Hindi are similar to those used in English.

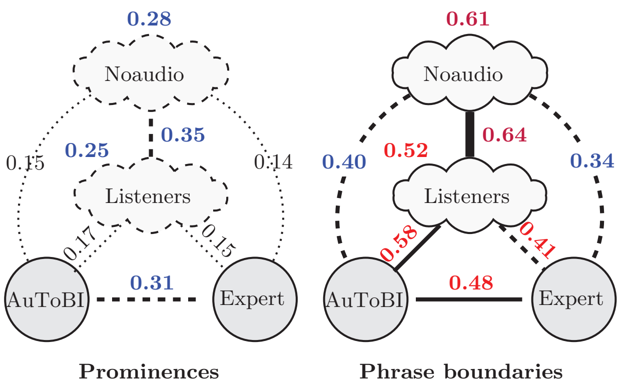


Figure 9: Kappa-score results: prominence and boundary detection in Hindi. The kappa scores 0.28 and 0.61 represent the levels of agreement among subjects with no audio, about transcription of prominence and boundary, respectively. The scores 0.25 and 0.52 represent levels of agreement among subjects with audio, about transcription of prominence and boundary, respectively. All other kappa scores shown in the figure represent pair-wise levels of agreement between the transcriber groups at either end of a link shown in the figure.

Kappa scores computed based on descriptions of prominence show a quite different pattern from the boundary scores. Listeners without linguistic training show fair agreement with one another, providing some support for the claim that prominence is perceptible in Hindi. Listener responses to audio stimuli show fair agreement with the responses to text stimuli, supporting the claim that, whatever the Hindi listeners are perceiving, it can be predicted from text without audio (as was also true of phrase boundaries). It is not yet clear, from these results, whether the labels produced by non-expert listeners have any acoustic correlates, or whether they have only textual correlates. Unlike B-scores, P-scores produced by untrained listeners show only slight agreement with the labels produced by the trained linguist, indicating that the instructions given to untrained labelers in the RPT study were effective in achieving the desired outcome: the expert linguist labeled the pitch accent on nearly every content word, but the listeners without linguistic training labeled only a subset of these words. Finally, the English-language AuToBI system showed fair agreement with the expert linguist's pitch accent labels, and very little (slight) agreement with the non-expert labels, suggesting that the acoustic correlates of pitch accent in English can be used to predict presence of a content word in Hindi, but not phrasal prominence.

5 Crowdsourcing under conditions of language mismatch

In the age of globalization, the act of listening to a language you do not understand is the source of frequent amusement. Transcribing what it sounds like (using words in your own language) is called, in Japanese, *soramimi* (literally 'empty ear'), after the "Soramimi Hour" in the popular TV program "Tamori Club". In English, transcribing speech using words of the wrong language is sometimes called "buffalaxing," after the screen name of the author of the popular "Benny Lava" video. Buffalax (Mike Sutton) listened to a Tamil love song, "Kalluri vaanil kaayndha nilaavo" (lit. 'the moon that scorched the college campus,' danced by Prabhu Deva and Jaya Seal in 2000), and heard the words "My loony bun is fine Benny Lava." He proceeded to add subtitles to the entire video, providing English lyrics that were phonetically similar to the original Tamil lyrics, but absurd and often outrageous (Phan 2007).

It is important to keep in mind that the Buffalax lyrics represent a single perception: this is the transcription produced by one listener, listening to sung speech with background music, under unknown listening conditions. He is explicitly seeking to map unintelligible Tamil phone sequences into real English words: more than that, he is explicitly searching for an English word sequence that will be funny. Despite its limitations, as a real-world speech perceptual experiment, the Benny Lava video is enlightening. The word “kalluri,” for example, was heard as “my loony,” perhaps in part because the flapped /n/ of “loony” is one of the English phones acoustically most similar to the flapped /r/ of “kalluri.” More enlightening is the title of the spoof: the phrase “fine Benny Lava” is derived from the Tamil words “kayndha nilaavo.” If second-language speech perception tends toward mistakes that minimize distinctive feature substitutions, as proposed by many standard second language (L2) acquisition models (Strange 1995), then the aspirated /dh/ in “kayndha” should have been misperceived as an English /d/, and the title of the spoof should have been “Danny Lava.” Instead, for some reason, the stop was perceived as a /b/. In further experiments described in the remainder of this section, we find that the minimum-feature-distance substitution pattern described by standard L2 acquisition models is approximately sustained, but that L2 phones with no exact first language (L1) match are subject to considerably more variable interpretation than phones with an exact L1 counterpart.

The “Benny Lava” experiment exemplifies a number of the problems that have been addressed in the literature on second language speech perception. Two of the most influential theoretical models of second language speech perception are Flege’s Speech Learning Model (SLM; Flege 1987, Flege 1995, Flege 2007; Flege et al. 2003) and Best’s Perceptual Assimilation Model (PAM) (Best et al. 1988; Best 1994, Best 1995). Flege demonstrated that learners of a second language (L2) create new phonetic categories for the L2 before they are able to reliably distinguish novel phonemes, and that any given L2 category may therefore lump together more than one phoneme in the L2. For example, an American beginning to learn French may create a new French /u/ category that differs from either the American /u/ or the French /u/, because the learner has mistakenly grouped together exemplars of the distinct French vowels /u/ and /y/ (Flege 1987). Best studied L2 phoneme perception during first exposure to the L2, and defined six possible relationships between L2 phonemes, depending on the way in which the L2 phonetic distinction interacts with the L1 (first language) phonological system. A pair of L2 phonemes both mapped to the same L1 phoneme (as in Flege’s /u,y/ example) possess a “Single Category” relationship, which Best demonstrated leads to the greatest difficulty in perceptual learning. In some cases, one of the L2 phonemes may be considered a very good example of the corresponding L1 category, while the other is considered a very poor example; Best defined this relationship to be a “Category Goodness” relationship, and demonstrated that it leads to substantially improved perceptual learning. Two L2 phonemes mapped to different L1 categories possess a "Two Category" relationship, and can be distinguished immediately. Finally, Best defined three different relationships that may occur when one or both of the L2 phonemes are "Uncategorizable," or for other reasons cannot be mapped to any L1 phonetic category, in which case their distinguishability therefore depends on auditory perceptual acuity rather than phonetic dimensions. The mapping of the Tamil word sequence "kayndha ni laavo" to "fine Benny lava," for example, may indicate (if it is not simply random noise: remember that this is a single perceptual trial) that the Tamil voiced breathy plosive /dh/ was perceived as "uncategorizable," rather than being mapped as a poor exemplar of either the English phonemes /d/ or /t^h/.

In creating the "Benny Lava" video, Buffalax explicitly sought English word sequences that were phonetically similar to the Tamil lyrics. Less explicit effects of L1 vocabulary have been demonstrated in every L2 learning situation, even in situations where learners seek to hear nonsense words. Ganong (1980) showed that phonetic category boundaries between L1 phonemes (e.g., voice onset time boundaries between /d/ and /t^h/ in English) shift in a direction that increases the probability of hearing a known English word. Norris et al. (1997) showed that L1 phonotactics also play a role: nonsense phonetic content is perceived in a manner such that every perceived phoneme is part of a phonotactically acceptable L1 "word" (they call this the "possible word constraint").

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. We propose a methodology that bypasses the need for native language transcription. Our methodology, which we call "mismatched crowdsourcing," is essentially formalized *soramimi*: we propose that speech in a target language should be transcribed by crowd workers who have no knowledge of the target language, and that explicit mathematical models of second language phonetic perception can be used to recover an equivalent transcription in the language of the speaker (Figure 10).

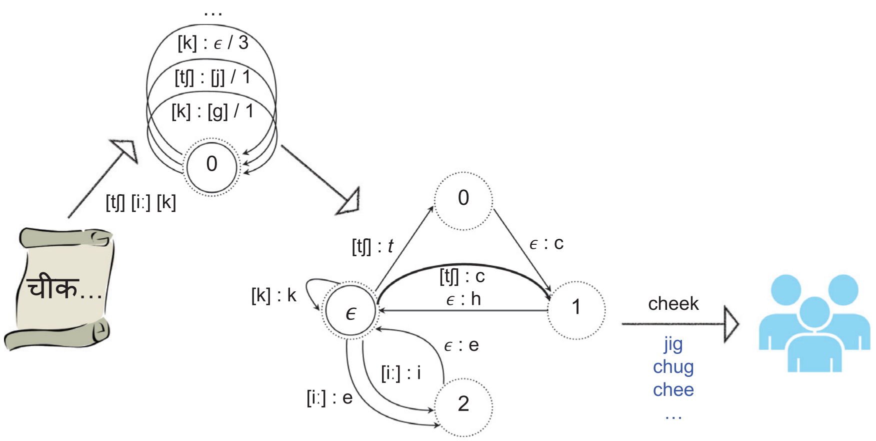


Figure 10: A finite state transducer model of mismatched crowdsourcing: the talker's target phone string is mapped to an implemented phone string by way of a phonetic reduction transducer, then to a perceived phone string by way of an explicit model of second language speech perception. To the left of each colon is a Hindi phonetic segment label, drawn from transcriptions by a Hindi-speaking phonetician. To the right of each colon is a letter that was used, by some subset of the crowdsourcers, as a transcription for the specified segment. An ε is a null symbol, representing letter insertion (ε to the left of the colon) or phone deletion (ε to the right of the colon).

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 spoken language to transcribed language can therefore be modeled as a noisy channel represented by the composition of two FSTs (Figure 10): a pronunciation model and a mismatch model. The pronunciation model is an FST representing processes that distort the canonical phoneme string during speech production, including processes of reduction and coarticulation. The mismatch model represents the mapping between the spoken phone string (in symbols matching the phone set of the spoken language) and the transcribed phone string (in symbols matching the orthographic set of the transcribed language).

Of these two FSTs, only the mismatch FST is not a component in any current standard speech technology. The mismatch FST is similar to phone substitution models that are used in computer-assisted language learning (CALL) software; e.g., we have learned CALL models in previous research by initializing weights according to substitution models reported in the second language learning literature, factoring the weights according to a distinctive feature based representation of each phone, and then applying machine learning methods to refine the classifier

(Yoon et al. 2009). In research reported in this paper, two different mismatch transducers were tested. The first was an FST implementation of phonetic segment string edit distance, with substitution weights set proportional to the number of phonological distinctive features separating the spoken from the perceived phoneme. The second transducer was identical to the first, but with substitution weights learned from the data resulting from a mismatched crowdsourcing experiment. In order to train the mismatch FST, training data were created by asking crowd workers to transcribe Hindi speech using English-orthography nonsense syllables (producing an English-orthography nonsense transcription of each utterance, i.e., a sequence $\Psi = [\psi_1, \psi_2, \ldots]$ of English letters ψ_i), and by then re-transcribing each utterance with a fine phonetic transcription produced by a Hindi-speaking linguist, $A = [a_1, a_2, \ldots]$. Using these transcriptions, the FST substitution costs, deletion costs, and insertion costs can be learned in order to minimize the total alignment cost between mismatched transcriptions and native language or expert transcriptions of a small sub-corpus.

Preliminary experiments in mismatched crowdsourcing were carried out by Jyothi and Hasegawa-Johnson (2015) using Hindi speech excerpts extracted from Special Broadcasting Service (SBS, Australia) radio podcasts. Approximately one hour of speech was extracted from the podcasts (about 10,000 word tokens in total) and phonetically transcribed by a Hindi speaker. The data were then segmented into very short speech clips (1–2 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 (Ψ) was aligned with the Hindi phone transcripts (A) using the mismatch FST illustrated in Figure 11. 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 using the expectation maximization algorithm (EM) (Dempster et al. 1977).

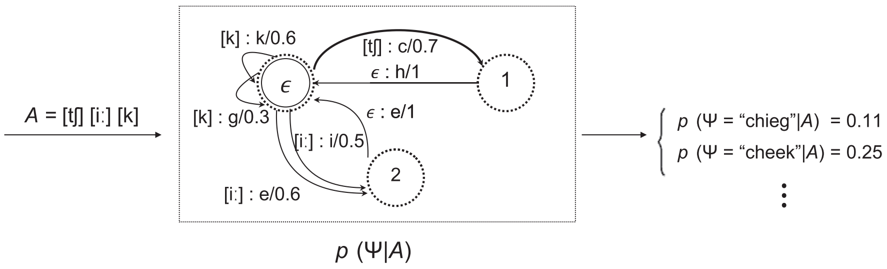


Figure 11: Mismatch FST model of Hindi transcribed as English. Hindi phones (to the left of each colon) were replaced, by crowd workers, with English letters (to the right of each colon). The number following each replacement is the probability of the replacement shown.

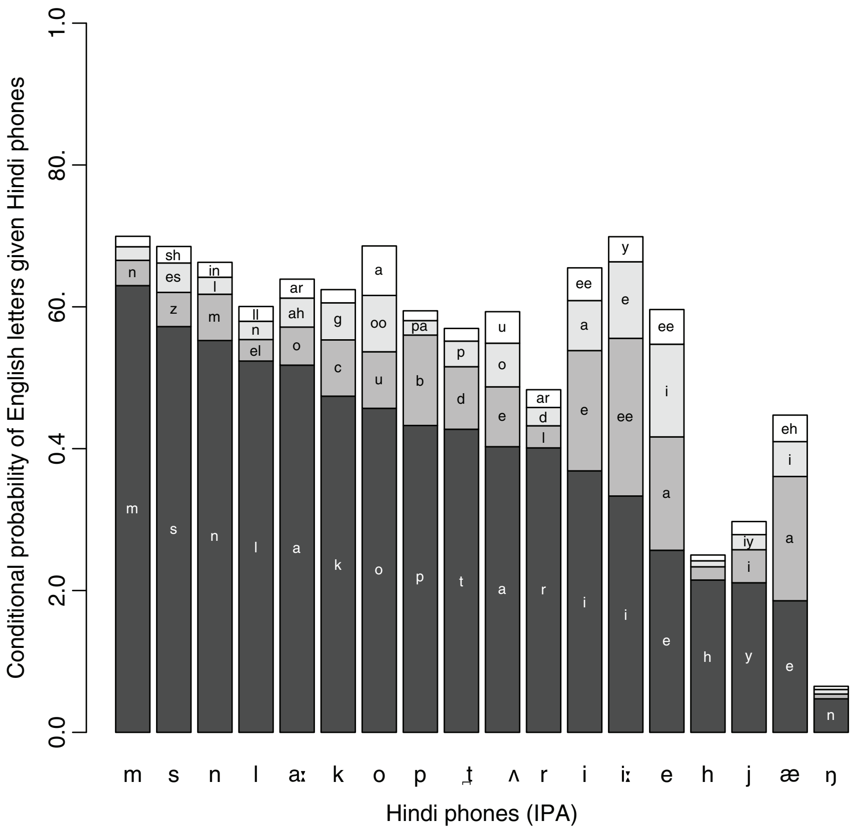


Figure 12: Hindi sounds (phones labeled using IPA symbols) with probabilistic mappings to English letter sequences. Size of each bar indicates how often each Hindi phone was transcribed using each English letter or two-letter sequence. Any English letter sequence used by fewer than two crowd workers (for any given Hindi phone) is not shown in the figure, which is the reason that the bars do not total 100%.

Figure 12 (from Jyothi and Hasegawa-Johnson 2015) visualizes the main probabilistic mappings from Hindi phones (labeled using IPA) to English letter sequences, as learned by EM; the arc costs were initialized using uniform probabilities for all Hindi phone to English letter transitions. Only Hindi phones with 1,000 or more occurrences in the training data are displayed. The plot indicates that the crowd workers predict many of the Hindi sounds correctly, and in cases where they do not, the plot reveals some fairly systematic patterns of mismatch. For example, unaspirated voiceless stops in Hindi such as /p/ and /k/ were sometimes mistaken as their voiced counterparts, /b/ and /g/, respectively. This could be because voiceless stops in Hindi are unaspirated even in word-initial syllables, unlike in English, which causes them to be confused for their voiced counterparts when transcribed by speakers unfamiliar with Hindi.

It is useful to further quantify the variety of transcribed English letters matching each Hindi phone. For example, consider the hypothesis that Hindi phones that also exist in English (phones whose transcriptions, by linguists studying Hindi and/or English speech, commonly use the same IPA symbol) are perceived less variably than phones that do not. This hypothesis can be tested by measuring the equivocation of the English letter transcription (Appendix B), conditioned on the true Hindi phone label. Equivocation computed using standard formulas (Shannon and Weaver 1949; see also Appendix B) is shown in Table 2. The phone class in the last row of Table 2 refers to consonants in Hindi that are commonly transcribed using IPA symbols that do not appear in standard English phonetic transcriptions. The equivocation was lowest for the class of consonants that appeared both in Hindi and English, suggesting that crowd workers were more certain about transcribing these sounds. Conversely, the class of Hindi consonants not appearing in English had the highest equivocation.

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

Phone classes (in Hindi)	Conditional equivocation (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

Having trained the mismatch FST, we now have a complete and invertible model of the process by which Hindi words are transcribed into English orthography. The language model represents $p(W)$, the probability of the Hindi word string $W = [w_1, w_2, \ldots]$. The pronunciation model represents $p(A|W)$, the conditional probability of a Hindi phone string given a Hindi word string. The mismatch transducer shown in Figure 11 represents $p(\Psi|A)$, the probability of the English letters given the Hindi phones. By composing and searching these FSTs, it is possible to estimate $p(W|\Psi)$, the posterior probability of any possible Hindi word string given the available English-orthography transcriptions, and to find the most probable such word string. More than half the time it turns out that the correct Hindi word string is not the one with maximum posterior probability. Instead, these preliminary results suggest that the correct word string is one of the top eight, and that the size of this list can be typically reduced to four by combining the transcriptions from multiple crowd workers. In order to use mismatched crowdsourcing at scale, it will be necessary to apply other methods in order to reduce the size of the N-best list.

Methods derived from communication theory may be applicable. Traditional information theory (Shannon and Weaver 1949) is often concerned with the scenario where the decoder (from the mismatched crowdsourced labels) makes a single "hard" decision about which symbol was produced, but when we have (costly) access to an expert labeler, it is useful for the decoder to produce more than just a single estimate alone (Figure 13). If the decoder puts out more than one estimate, the result is called an N-best list, and the expert can then reduce the remaining equivocation in a feedback loop. We suggest that future work will be able to exploit the typical set in order to permit one informant, who understands the spoken language, to very quickly select the correct spoken-language transcription of each spoken utterance. Thus, for example, utterances might be given to many mismatched crowdsourcers, reducing the size of the N-best list to fewer than 32 words. These 32 words might then be listed to a Hindi-speaking linguist, who works as a sort of high-efficiency "correcting device" (Figure 13). The prompt screen lists $N(W|\Psi) + 1$ options: The $N(W|\Psi)$ Hindi words that are most probable given the English transcription, and 1 option that says "OTHER" and allows the linguist to type something different.

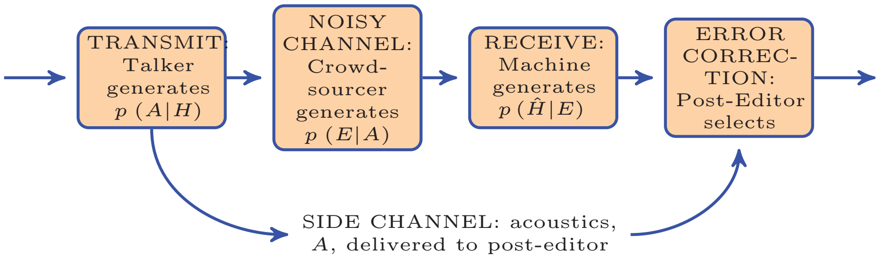

TRANSMIT: Talker generates $p(A|H)$
NOISY CHANNEL: Crowd-sourcer generates $p(E|A)$
RECEIVE: Machine generates $p(\hat{H}|E)$
ERROR CORREC-TION: Post-Editor selects
SIDE CHANNEL: acoustics, A, delivered to post-editor

Figure 13: Noisy channel correction model.

In order to scale these methods, it will also be necessary to acquire mismatched transcriptions from a large number of internet users, typically users who are interested in language, but who do not speak the language they are transcribing. By setting up mismatched transcription as a code-breaking task, we believe that it can be made enjoyable, increasing the number of internet users who are willing to

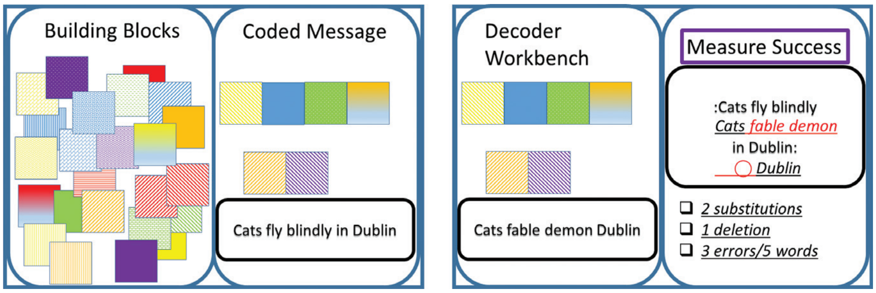


Figure 14: Conceptual mockup of the Secret Agent Game. Left: One player, the "coder", sequences blocks of Hindi audio so that they sound like an English sentence. Right: The other player, the "decoder", tries to guess the transcription of the sentence constructed by the coder.

help us with this task. Figure 14 shows the mockup of an on-line game in which one player (the "coder") selects and sequences blocks of non-English audio so that they sound like an English language sentence. The second player (the "decoder") listens to the sequenced audio clips, and tries to guess the English language sentence that was transcribed by the coder. Both players receive points in proportion to the overlap between their transcriptions. A prototype of this game was deployed at the 2015 Beckman Open House at the University of Illinois.

6 Conclusions and future work

In order to model the phonetic correlates of phonology in any language, it is necessary to first (1) define the form of the model, (2) collect relevant training data, and (3) use the data in order to learn the parameters of the model. The number of parameters that can be learned from a dataset is, in general, directly proportional to the number of data. Methods for more precisely estimating the best fit to a dataset include structural risk minimization and cross-validation.

Crowdsourcing is the set of methods whereby labels are solicited from a large temporary labor market, often on line. Crowdsourcing provides labels at much lower cost than traditional professional data transcription, but crowdsourced labels also tend to have higher rates of transcription error. Majority voting reduces error, but triples (or worse) your cost.

Error-correcting codes can be used to reduce the error rate of crowdsourcing: if you factor each hard question into several easy (binary) questions, you can improve accuracy more cheaply, because each crowdsourcer only needs to be partially correct. Error-correcting codes can also be used to divide a task with a relatively constrained labor market (e.g., transcription of a rare language) into many smaller tasks, each of which can be released into a larger labor market (e.g., transcription of binary distinctive features, each of which is easily perceptible by the native speakers of one or more widely-spoken languages).

Besides its economic benefits, factoring a hard problem into many easy questions also has substantial scientific benefits. There are many questions that, phrased precisely, are comprehensible only to people with formal linguistic training. It is often possible to factor the hard question into many easy questions, and to phrase each of the easy questions in a way that is comprehensible to people without formal linguistic training. In this way, the science of easy questions permits us to explore the cognitive representations that are applied to the task of speech perception by people with no formal linguistic training.

Another method for acquiring transcriptions of a rare language is mismatched crowdsourcing, which we define to be the transcription of speech by those who have never learned the language being spoken. Mismatched crowdsourcing generates errors that are biased by the non-native perception of the listener. Biases of this kind have been heavily studied in the second-language learning literature; mismatched crowdsourcing provides a new method for studying these biases, and a new motivation for characterizing them precisely. In particular, by representing second-language speech perception as a noisy channel (a probabilistic finite state transducer), it becomes possible to compute the most likely source-language message based on error-filled mismatched transcriptions.

Future work will include further analysis of the mismatched crowdsourcing model (e.g., using ideas from the communication theoretic literature on guessing with side information; Sundaresan 2006). Some experimental validation of the mismatched crowdsourcing model has already been performed (Jyothi and Hasegawa-Johnson 2015), and more is under way.

The economic goal of this research is to scale the mismatched crowdsourcing approach, using methods including semi-supervised learning (train an automatic speech recognizer when only a few labels are available, and then use it to help acquire more labels) and active learning (use the half-trained speech recognizer to determine which data need human labels), in order to develop speech technology in languages for which the societal benefits of speech technology outweigh the commercial benefits. For example, consider the problem of helping geographically constrained people in remote communities (e.g., mothers with children) to buy and sell on the internet. There are more cell phones than humans in the world today; many people are able to access the voice network who have no reliable access to any other telecommunications modality. People in remote locations are less likely to speak a majority language, however, so if a woman wishes to sell her products on-line, she needs to rely on an educated intermediary to set up the website for her. There is no money to be made by creating speech technology in her language. Of the 6,000 languages spoken in the world today, fewer than 100 are spoken by more than eight million speakers each. Porting speech technology to a language with fewer than one million speakers is unlikely to yield great financial incentives, but if it can be accomplished automatically and almost for free, then we can use this technology to provide freedom and power to those who most need it.

Acknowledgments: The authors wish to thank Kikuo Maekawa, Natasha Warner, and one anonymous reviewer for extraordinarily detailed and well-informed feedback that significantly improved the quality of the manuscript. Thanks to the organizers and participants in the 2014 Conference on Laboratory Phonology for ideas that became part of this manuscript, and thanks to NINJAL (the National Institute for the Japanese Language) for supporting the conference. Mark Hasegawa-Johnson's research reported here is supported by QNRF NPRP 7-766-1-140. Jennifer Cole's research reported here is supported by NSF BCS 12–51343. All results and opinions are those of the authors and are not endorsed by sponsors of the research.

Appendix A: The mathematical theory of learning

The goal of this appendix is to review key results from the Mathematical Theory of Learning, as it has been developed by many authors during the twentieth century and the first part of the twenty-first. Mathematicians have studied the rate at which statistical estimates converge to true parameter values since at least the 1850s (Bienaymé 1853; Tchebichef 1867), but the first general theorem with an exponential rate of convergence was proved by Chernoff in the 1950s (Chernoff 1952). Vapnik and Chervonenkis (1971) showed that a classification function is an example of such a convergence. The name "Theory of the Learnable" was coined by Valiant (1984). This section reviews key results for the tasks of supervised learning, active learning, and crowdsourcing.

A.1 Type of classification function

The Mathematical Theory of Learning (Vapnik and Chervonenkis 1971; Valiant 1984) assumes that there is a particular functional mapping that we wish to learn, i.e., we wish to learn a function $h(\cdot)$ that generates the correct label, $y = h(x)$, for every possible observation x. For example, x might be a spectrum, and y might be the corresponding label. If x is a spectrum extracted from an instance of the vowel /i/, then y should have the value $y =$/i/; if x is a spectrum extracted from an instance of /ɔ/, then y should have the value $y =$/ɔ/.

A "classifier" is a computer program that implements the function $y = h(x)$: it takes some type of observation as input (a spectrum, say), and generates some type of label as output (a phoneme label, say). A "machine learning algorithm" is a computer program that is able to rewrite other programs so that they work better. In particular, a machine learning algorithm accepts two inputs: the code specifying how the classifier should work, and a "training database" of examples. Given these inputs, the machine learning algorithm generates one output: a revised classifier program.

A machine learning algorithm doesn't usually re-write the java, or whatever programming language the classifier is written in. Instead, the machine learning algorithm tweaks certain numerical parameters that govern the behavior of the classifier: these parameters are read from disk by the machine learner at the beginning of the learning process; then the improved parameters are written to disk at the end of learning.

Machine learning is called "supervised learning" if the machine learning algorithm has access to a labeled training database, that is, a database, D, that contains N different examples of spectra that the classifier should observe, and the labels that the classifier should generate when it observes each spectrum. For example, suppose that the database contains $N = 20$ example spectra, numbered x_1 through x_{20}. Suppose that the first 10 spectra are extracted from instances of /i/, and the last 10 spectra are extracted from instances of /ɔ/: then the corresponding training labels should be $y_1 = y_2 = \ldots = y_{10} =$ /i/, and $y_{11} = \ldots = y_{20} =$ /ɔ/.

The size of the labeled training database is important: in almost all cases, the larger N is, the better chance the machine learning algorithm has of correctly learning the function $h(x)$. In the Betelgeusian example, a training database might be created by eliciting from our Betelgeusian speakers productions of words containing each of the vowel phonemes of interest. Correctness of the labels might be further confirmed by excising each word, and playing it back, and asking a second group of Betelgeusian speakers to choose that word from a list of pictures of near-homonyms. Almost all of the papers published in the theoretical machine learning literature address this situation. If the characteristics of the function you want to learn are well known, then tight upper bounds exist that can tell you how many labeled data are necessary to learn $h(x)$ with any desired level of accuracy. If the characteristics of $h(x)$ are not known in advance (a much more reasonable assumption), then theoretical bounds can nevertheless be used to give some guidance about the level of accuracy you might expect.

Learning requires constraints and data. For example, consider the following constraint: let us assume that the function $h(x)$ comes from some parameterized function family H which is parameterized by some parameter vector θ, and that the only thing we need to learn is the parameter vector θ. In other words, the classifier is a set of pre-programmed code, whose behavior depends on the numerical values of the parameters in the vector θ. The machine learner knows exactly the way in which $h(x)$ depends on θ, so the machine learner is able to find the values of θ that will optimally match the examples given in the training database D.

One of the simplest classifiers we could learn is a Gaussian classifier. A Gaussian classifier assumes that the tokens x that correspond to each vowel label y (remember that particular vowels have the labels $y = /\text{i}/$, $y = /\text{ɔ}/$, and so on) are distributed according to a Gaussian distribution (a "Gaussian" is another name for a normal distribution). For example, this is exactly the way in which Peterson and Barney (1952) described the vowels of English. They proposed that each vowel token is described by a two-dimensional vector containing the first two formants, $x = [F_1, F_2]$. They proposed that each vowel label, y, is described by a mean vector and a covariance matrix. The mean vector, for the vowel labeled y, is $\mu_y = \left[\bar{F}_1, \bar{F}_2\right]$, where $\bar{F}_1$ is the average F_1 for all tokens of that vowel, and $\bar{F}_2$ the average F_2. The covariance matrix is a way of writing the variance of each of the formants, and their covariance (the covariance between two measurements is a scaled version of their correlation). Most of the machine learning literature abuses notation in the following way: we say that $\theta = [\mu_1, \Sigma_1, \mu_2, \Sigma_2, \ldots]$, that is, we say that the "vector" of learnable parameters, θ, includes, as its components, the spectral mean vectors μ_1 and μ_2 (and so on, if there are more than two vowels), and the covariance matrices Σ_1 and Σ_2 (and so on).

The number of learnable parameters is also very important. A Gaussian classifier using two formant frequency vectors has only five learnable parameters per vowel type (two mean formant frequencies, two variances, and one covariance). What if, instead of measuring formant frequencies, we use the whole log-magnitude spectrum as our measurement? A log-magnitude spectrum might be a 512-dimensional measurement; characterizing it would require $d = 512$ average spectral measurements, $d = 512$ variances, and something like $d^2 = 512^2$ covariances (one for every pair of spectral measurements), for a total of $n = 512 + 512 + 512^2 =$ some awfully big number. Most machine learning papers ignore the covariance, at this point, and only model the means and the variances, which allows us to model only $n = 512 + 512 = 1024$ trainable parameters per vowel type – still a big number, but not quite as big. Most machine learning papers also use a shortened summary of the spectrum: we can make n smaller by choosing a smaller d. In the linguistics literature, d is often compressed to just two dimensions: the two formant frequencies. The problem with formants is that they require a lot of work to extract (even if you use an automatic formant tracker, you have to check what it produces for errors), so automatic speech recognition studies often use some type of cepstrum instead. A cepstrum is a relatively low-dimensional summary of the shape of the entire spectrum, which has the benefit that it can be computed completely automatically, with no need for human error checking. For example, it is typical that a log-magnitude FFT vector has $d = 512$ dimensions, but that a cepstrum might have only $d = 13$ dimensions (not quite as good as $d = 2$, but a lot better than $d = 512$). The two types of cepstra most commonly used in automatic speech recognition are the mel-frequency cepstral coefficients (MFCC; Davis and Mermelstein 1980) and the perceptual linear predictive coefficients (PLP; Hermansky 1990); open source code is available on line that will produce either of these representations automatically from a speech signal.

In the Betelgeusian example, we impose this constraint by choosing, in advance, the type of vowel classifier we will learn. For example, we might compute a 13-dimensional PLP feature vector from each centisecond of each vowel, and then train a classifier to distinguish the vowels.

In order to learn θ, suppose that we have a training dataset D that contains N labeled training examples, $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, selected i.i.d. (independent and identically distributed) from some unknown true probability distribution $P(x, y)$. The i.i.d. assumption requires that we have recorded from a group of informants and contexts that is sufficiently variable to be representative of the language. Usually, the best way to meet this requirement is to redefine "representative of the language", e.g., by restricting ourselves to consider talkers of only one gender, in only one pre-specified consonant context (e.g., Peterson and Barney 1952). Having definitively learned the vowel space under such constraints, it is then possible to consider a wider variety of contexts.

A.2 Performance guarantees

In order to evaluate a learner, we need some metric of success. Suppose we have defined some loss function $\ell(y, h(x))$; for example, $\ell(y, h(x))$ might be a binary function that is equal to one if the automatically generated vowel label ($h(x)$) is not equal to the correct label ($h(x) \neq y$), and equal to zero if no error occurs ($y = h(x)$). Our goal is to minimize the Risk, which we define to be the expected loss.

In order to define "expected loss," we need to define what we mean by "expected." In mathematics, the word "expected" always implies that the data are distributed according to some probability distribution, $P(x, y)$. The distribution $P(x, y)$ tells us the probability of x and y both occurring at any given instant. For example, suppose that you turned on your radio right now. $P(x, y)$ tells you the probability that the radio announcer is producing the phoneme y at exactly the moment you turn on the radio, *and* that the spectrum he is using to produce that phoneme is x. Obviously, $P(x, y)$ depends on what language the announcer is talking. Also obviously, if y is a vowel that doesn't exist in the language being spoken by the announcer, then $P(x, y) = 0$. In fact, if x is a spectrum that could never, ever be produced while the announcer is trying to say y, then $P(x, y) = 0$. Conversely, if a particular spectrum, x, is three times as likely to occur with vowel label $y1$ than with vowel label $y2$, then we would represent that by saying that $P(x, y1) = 3P(x, y2)$.

The definition "risk is equal to expected loss" can be written mathematically as follows:

$$R(h) = \sum_{y} \int P(x, y)\ell(y, h(x))dx \qquad [1]$$

which means that we find the risk by multiplying the loss for any particular observation/label pair ($\ell(y, h(x))$) by the probability of drawing that particular pair ($P(x, y)$) and then integrating. The integration computes a weighted average of $\ell(y, h(x))$, where the weights are given by the probabilities $P(x, y)$.

If we were able to listen to English-language radio broadcasts forever, then we could build up a very good model of the probability distribution $P(x, y)$ for English. Remember, computers represent the spectrum x as a short list of ones and zeros. That means that every spectrum that the computer can possibly represent will *eventually* occur, if we listen to the radio long enough. If we listen even longer, then we will hear every possible x spoken as a token of every possible phoneme, y. You would probably never hear a fricative spectrum, x, uttered as an instance of the vowel $y = /\mathrm{i}/$, but that is just because $P(x, y) = 0$ for this particular combination: every combination for which $P(x, y) > 0$ will eventually occur. Then, by measuring the frequency of these occurrences, we would get an accurate lookup table for the probability distribution $P(x, y)$. Unfortunately, having such a lookup table for English will not help us at all if we want to model French; in order to estimate $P(x, y)$ for French, we have to start all over again.

If we knew the true probability distribution $P(x, y)$, then we could compute the Risk associated with every possible classifier, and we could just choose the best one. Unfortunately, we can never perfectly know the true value of $P(x, y)$, because to know it, we would first have to measure an infinite amount of training data. Instead, all we have available is a randomly selected training database, D. Remember that D is a database containing N labeled training examples, $D = \{x_1, y_1, \ldots, x_N, y_N\}$. We can use D to compute an empirically estimated Risk, as follows:

$$\hat{R}(h; D) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, h(x_i)) \qquad [2]$$

where the notation $\widehat{R}$ means "estimated value of R," and the notation $\hat{R}(h; D)$ is a way of explicitly stating that this particular estimate depends on the particular set of training examples, D. In most speech applications, $\ell(y, h(x))$ is just an error counter, so it makes sense to call $\hat{R}(h; D)$ the "training corpus error," and to call $R(h)$ the "expected test corpus error".

The difference between the training corpus error and the expected test corpus error is governed by the size of the training dataset, D, and the size of the hypothesis class, H. Here we are using the term "hypothesis" to describe a particular labeling function $y = h(x)$, and the set of parameters θ that tell us how to do the labeling – so the "hypothesis class", H, is the set of all of the different labeling functions you might use. The idea that H has a "size" is counter-intuitive to most scientists; most scientists assume that there are an infinite number of hypotheses available to them, and the task is to choose the best one. When the computer is doing the learning for you, though, that will not work. The training dataset, D, is finite (remember that it has only N labeled examples in it), so the set of hypotheses that you allow the computer to consider must be either finite or severely constrained. A "finite hypothesis class" is a finite list of labeling functions $H = \{h_1, \ldots, h_m\}$: if the hypothesis class is finite, then the job of the machine learning algorithm is simply to try all m of them, for all N of the training data, compare the generated labels to the true labels for every training token and for every possible hypothesis, and choose the hypothesis that has the smallest error rate. Valiant (1984) showed that, under this circumstance, the training error rates of every hypothesis in the hypothesis class converge to their expected test corpus error rates. He quantified this convergence by defining small threshold variables ε and δ, and by proving that, for every possible δ in the range $0 < \delta < 1$, there is a corresponding ε in the range $0 < \varepsilon < 1$ such that the following is true:

$$Pr\left\{\max_{h\in H}\left|\hat{R}(h;D) - R(h)\right| > \varepsilon\right\} \le \delta \qquad [3]$$

Equation [3] defines ε to be *probably* the Estimation Error – the difference between the training corpus error and the expected test corpus error $\hat{\mathrm{R}}(\mathrm{h};\mathrm{D}) - \mathrm{R}(\mathrm{h})$. Equation [3] says that the Estimation Error is *probably* no larger than ε; the probability that the ε threshold gets violated is δ. Equation [3] is the first published example of the type of theoretic guarantee called a "Probably Approximately Correct" (PAC) learning guarantee. The idea that the Estimation Error is only *probably approximately* zero is disturbing, until you realize that we haven't put any constraints, at all, on either the training corpus or the test corpus. If the training corpus and the test corpus are both taken from the same larger set of data (the same underlying $P(x,y)$), then eq. [3] holds regardless of what type of data they are.

Regardless of what type of data you are studying, the probability that you randomly choose a training dataset that causes large Estimation Error (larger than ε) is very small (less than δ). For any given value of δ, Valiant showed that you can achieve the following value of ε (for some constants a_1 and a_2 that, unfortunately, depend on the particular problem you're studying):

$$\varepsilon \le \frac{a_1\log(m) - a_2\log(\delta)}{N} \qquad [4]$$

Equation [4] suggests the following learning algorithm. Create a list of possible classifiers, $H = \{h_1, ..., h_m\}$. Acquire some training data, $D = \{x_1, y_1, \ldots, x_N, y_N\}$. Figure out which of your classifiers is best on the training data. Then eq. [4] says that the difference between training corpus error and testing corpus error is no larger than $a_1\log(m)/N$. We don't know what a_1 is, so the numerator of this bound is not very useful – but the denominator is very useful. The denominator says that if you want to halve your Estimation Error, all you have to do is double the amount of your training data.

Machine learning algorithms that choose from a finite list ($H = \{h_1, \ldots, h_n\}$) are actually not very useful. A much more useful machine learning algorithm is one that finds, for some classifier function, the best possible set of real-valued classifier parameters – remember that a few paragraphs ago, we called this parameter vector θ. So, for example, the classifier designed by Peterson and Barney (1952) was parameterized by average and standard deviation of each formant frequency, for each vowel type. Formant frequencies are real numbers, e.g., if the average F2 of the vowel /ɔ/ were 1,229.58 Hz, that might actually define a slightly different set of vowels than some other vowel type whose average F2 was 1,230 Hz. In most cases, the equivalent "size" of a continuous hypothesis space is, roughly speaking, $m = 2^n$, where n is the number of trainable parameters; for example, Barron (1993) showed that this is true for two-layer neural networks, while Baum and Haussler (1989) demonstrated the same result for neural nets of arbitrary depth. Thus, with probability at least $1 - \delta$, the difference between training corpus and testing corpus is no worse than the Estimation Error ε, which is probably

$$\varepsilon \leq \frac{b_1 n - b_2 \log(\delta)}{N} \tag{5}$$

for some constants b_1 and b_2. Again, the constants b_1 and b_2 depend on exactly what type of data you are working with, so it is hard to know them in advance. The only part of eq. [5] that is really useful is the part saying that $\varepsilon \leq b_1 n/N$: in other words, the Estimation Error is proportional to the number of trainable parameters divided by the number of training examples. So, for example, if you want to characterize each vowel with twice as many measurements (e.g., four formant frequencies instead of two), and you want to do it without any increase in Estimation Error, then you need to double the number of training tokens per type.

Barron (1994) pointed out that the randomness of the training data is not the only source of error. He observed that the ability of a neural network to learn any desired classifier is limited by the number of hidden nodes. If a neural network has just one hidden node, it can only learn a linear classifier; if it has n hidden nodes, it can learn a roughly piece-wise linear classifier with n pieces. Notice that a network with more hidden nodes also has more trainable parameters, thus there is a tradeoff: neural nets with more hidden nodes are able to learn better classification functions, but they require more training data. Barron showed that for neural networks, with probability $1 - \delta$, the error rate of the learned classifier is

$$\varepsilon \leq \frac{c_1 n}{N} + \frac{c_2}{n} + c_3 \tag{6}$$

where c_1, c_2, and c_3 are constants that depend on δ, on $P(x, y)$, and on the structure of the neural net. In words, eq. [6] says that there are three different sources of error. The term c_3 is sometimes called the Bayes Risk: it is the smallest error rate that could be achieved by any classifier. In speech applications, Bayes Risk is often nonzero, because we often ask the classifier to perform without adequate context. Even human listeners are unable to correctly classify phone segments with 100% accuracy if the segments are taken out of context (Cole et al. 1994). The term c_2/n is called, by Barron (1994), the Approximation Error, while Geman et al. (1994) call it Bias; it represents the smallest error that could be achieved by any classifier drawn from a particular hypothesis class – for example, this might be the lowest number of times the vowel /i/ would be misclassified as some other vowel if the classifier is of a particular type. The term $c_1 n/N$ is called, by Barron (1994), the Estimation Error, while Geman et al. (1994) call it Variance: it represents the difference between the training corpus error and the expected test corpus error.

Equation [6] is sometimes called the Fundamental Theorem of Mathematical Learning, and the conundrum it expresses – the tradeoff between n and N – could be called the fundamental problem of mathematical learning. Basically, the conundrum is this: if your model has too few parameters (n too small), then it underfits your training data, meaning that the model is not able to learn about the data. If it has too many parameters, however (n too large), then it overfits the training data: it learns a model that represents the training data, but that does not generalize well to unseen test data. Thus, for example, imagine trying to classify vowels by drawing a straight line across a two-dimensional formant frequency space. Each such straight line can be represented by $n = 2$ trainable parameters: the intercept and slope of the line. Since $n = 2$ is a very small number of trainable parameters, the equation for the line can be estimated with very small Estimation Error, but unfortunately, the Approximation Error is huge: most vowels are not well represented by a linear boundary in F_1–F_2 space. A quadratic boundary might give lower Approximation Error, at the cost of increased Estimation Error, because parameterizing a quadratic category boundary requires $n = 3$ trainable parameters. As you increase the number of training data (N), it is possible for you to effectively train a model that has more parameters: so, for example, the "rule of 5" suggests that you should use a linear boundary between vowel types ($n = 2$) if you have $10 \leq N < 15$ tokens per vowel type, but that if $N \geq 15$, you have enough data to try a quadratic boundary ($n = 3$). Finding a model that fits, without overfitting, can be done in a number of ways. For example, the structural risk minimization principle states that the ideal learned model should provide a balance between empirical risk and model complexity; learning algorithms that optimize a weighted combination of empirical risk and model complexity, such as support vector machines (Vapnik 1998), are commonly used for speech.

Another method that can be used to find the best model size is cross validation. In a cross-validation experiment, we hold out some fraction of the data as "test data." For example, we might use 20% of the data as a test set, in order to evaluate classifiers trained on the remaining 80%. A series of different classifiers are trained on the 80%, and tested on the 20%, and the error rates of all of the classifiers are tabulated. The experiment is then repeated four more times, providing a total of five different estimates of the error rate of each classifier. The classifier with the lowest average error rate is judged to be the best one for these data, and is re-trained, one more time, using the full training dataset.

A.3 Examples: Gaussian and GMM classifiers

Suppose, again, that we are trying to learn the average PLP spectra of Betelgeusian vowels. Remember that, for the goal of characterizing the acoustic difference between two hypothesized vowel phonemes of Betelgeusian, the method is to build a classifier that learns to classify vowels based on differences in their average PLP spectra. Suppose that we have two vowels, in particular, whose formant frequencies are almost identical, so we want to determine whether or not these two vowels can be distinguished based on their PLP feature vectors: perhaps one of the two vowels is $y = 1$, which is a code for $y =$/i/, and the other vowel ($y = 2$ in code) is a high-fronted vowel with a constricted pharynx whose classification is not quite clear, but it sounds something like $y =$/ɪ/ or $y =$/ɨ/. We have recorded N examples of vowel 1, and their PLP vectors have been collected into a dataset called $D_1 = \{\vec{x}_1, \ldots, \vec{x}_N\}$. Likewise, we have N examples of vowel 2, collected into a dataset $D_1 = \{\vec{x}_{N+1}, \ldots, \vec{x}_{2N}\}$. Each x is an n-dimensional PLP vector. We want to know: is N large enough to learn a Gaussian classifier separating these two vowels?

For purposes of exposition, let's simplify the problem: suppose that we want to learn a Gaussian classifier, but we're not going to allow the two classes to have different standard deviations. Instead, we'll only learn the average PLP vectors μ_1 and μ_2; we will assume that both vowel types have exactly the same standard deviation. Thus the only parameters we need to learn are the average PLP vector of vowel 1, and the average PLP vector of vowel 2. Thus we could say that the learnable parameter set is $\theta = \{\mu_1, \mu_2\}$, where these mean vectors can be estimated by averaging all of the training tokens for each vowel. Now we need to know whether N is large enough so that we can trust our estimates μ_1 and μ_2. The problem addressed by the Theory of Learning is that the training datasets, D_1 and D_2, were chosen randomly (we do not have perfect control over the choice of talkers who recorded data for us, nor were we able to control the speaker's emotional state, the speech style, speaking rate, or other factors influencing the acoustic form of each vowel); therefore θ is itself a random vector. Since θ is random, therefore any classifier function $h_\theta(x)$ that depends on θ is, because of its dependence on θ, random! Let us be more precise. For any particular classifier, the function $h_\theta(x)$ is some explicit function of the test datum being classified (x) and of the model parameters (θ). Suppose that we want to analyze the dependence of $h_\theta(x)$ on the randomness of the training data. One way to accomplish this is by assuming a particular fixed, non-random test datum (x), and by writing $h_\theta(x)$ explicitly as a function of x and θ. Any function of a random variable is also a random variable; since θ is random, $h_\theta(x)$ is random. For example, a Gaussian classifier distinguishing between the vowels /i/ and /ɪ/ might label any particular test token as y = /ɪ/ if the following function is positive, and y = /i/ otherwise, where the test function is $h\theta(x) = \log\frac{p_2(x)}{p_1(x)}$. Here $p_2(x)$ is a Gaussian probability density model centered at μ_2, and $p_1(x)$ is a Gaussian model centered at μ_1. Expanding this, we get an explicit function of both the test feature vector, x, and the trained model parameters, μ_2 and μ_1. It is normal to view the test datum as random, and the trained model parameters as constant, but eq. [6] also allows us to do the opposite. Suppose that we view the test datum as fixed, and suppose we view the training corpus as random. For example, suppose that we know, in advance, that we are going to try to classify a vowel that is exactly halfway between the two vowel categories /i/ and /ɪ/. In this case, the classifier *should* output $h_\theta(x) = 0$, since the test vowel is exactly halfway between the two vowel types (for convenience, let us also suppose that $x = [0, 0, \ldots, 0]$, perhaps because we are using normalized feature measurements). Since we know the test token in advance, the only thing that is still random is the training set: we will create a training set by choosing N examples of each vowel, at random, from some much larger database. In this case, $h_\theta(x)$ is random because the training database is random: the ideal value of $h_\theta(x)$ is zero, and any deviation from zero is entirely caused by Estimation Error. Well, knowing the form of $h_\theta(x)$ allows us to calculate exactly its residual randomness (its variance), and as it turns out, that residual randomness is

$$Var(h_\theta(x)) = \sigma_x^2 \frac{n}{N} \quad [7]$$

By comparing eq. [7] with eq. [6], the reader may verify that the variance of a Gaussian classifier (its variability, in response to a previously known test token, if the training database is chosen at random) has the same form as the general form of Estimation Error: it is proportional to the number of trainable parameters (remember that we assume an n-dimensional feature vector, so the total number of trainable parameters is $2n$), and inversely proportional to the size of the training database (remember that we assume N training tokens per vowel type, so the total size of the training database is also $2N$). In order to halve the Estimation Error, all we need to do is to double the size of the training database.

Similar convergence rules apply to more complicated machine learning models. Consider a Gaussian mixture model (GMM), for example. A Gaussian mixture model is a model in which we suppose that $p_y(x)$, the probability distribution of the yth vowel, is the weighted average of M different Gaussian distributions, as shown in Figure 2 (the figure shows the case $M = 4$). By assuming a GMM instead of a simple Gaussian, we get some benefits: for example, it is possible to represent a vowel that has several different modal varieties. The disadvantage of a GMM, as we will see, is that it requires us to learn a larger number of parameters. For example, consider a somewhat simplified GMM, in which every Gaussian has exactly the same standard deviation, so that only the Gaussian mean vectors have to be learned. Suppose that we are trying to distinguish between two vowel categories: the vowel y = /i/, which is represented by the GMM distribution $p_1(x)$, and the vowel y = /ɪ/, which is represented by the GMM distribution $p_2(x)$. In English (not necessarily in other languages, but let us suppose that Betelgeusian is similar), /ɪ/ tends to take on lip rounding features from the consonants surrounding it, so that an /ɪ/ extracted from the word *will* sounds exactly like /ʊ/ (when you listen to it without context). In order to represent this type of behavior, the vowel /ɪ/ needs a GMM with at least $M = 2$ Gaussians; the vowel /i/ might really only need $M = 1$ Gaussian, but just to be general, let us assume that there are the same number of Gaussians in each vowel, so that /ɪ/ uses M Gaussians, and /i/ also uses M Gaussians. As before, the classifier function $h_\theta(x)$ is the logarithm of the ratio between $p_2(x)$, the probability of y =/ɪ/, and $p_1(x)$, the probability of /i/. Suppose that the test vector x is fixed to exactly halfway between the two vowel categories, as before, so that the ideal output of the classifier is $h_\theta(x) = 0$. Since the training data are drawn at random, however, the actual value of $h_\theta(x)$ achieved by any particular training database is a random variable. In fact, for any fixed test vector x, the GMM classifier function $h_\theta(x)$ is approximately a χ^2 random variable, as was the Gaussian classifier. Unlike the Gaussian classifier, however, the GMM classifier function is a χ^2 random variable with only about $n_{yk} \approx \frac{N}{M}$ degrees of freedom, so to get equivalent small estimation error, we need M times as much training data:

$$Var(h_\theta(x)) = \sigma_x^2 \frac{nM}{N} \qquad [8]$$

The decrease in variance of the GMM classifier function $h_\theta(x)$, with increasing training dataset size, is exemplified in Figure 3. The key point to notice, in both the figure and in eq. [8], is that Estimation Error in a GMM is proportional to the total number of parameters Mn, not just to the number of acoustic feature dimensions n: so in order to limit Estimation Error, we need to choose enough training data so that Mn/N is a large enough number. For example, statisticians often use the "rule of 5," which says that the number of training examples needs to be five times larger than the number of trainable parameters: in this case, the "rule of 5" can be written as $Mn/N \geq 5$.

Appendix B: Conditional entropy

The variability of the transcribed English letters for Hindi phones can be quantitatively analyzed using equivocation for different phone classes as shown in Table 2. This table shows the equivocation as a function of phone class. Let a phone class be denoted A, and let a and $\hat{a}$ be particular phones drawn from this class, and let ψ be one or two English-language orthographic symbols (letters) written by a transcriber when he or she hears the phone a. Denote the equivocation of ψ given a, averaged over phones limited to phone class A, as $\eta(\psi|a, a \in A)$. This is defined as:

$$\eta(\psi|a, a \in A) = \sum_{\hat{a} \in A} \eta(\psi|a = \hat{a}) p(a = \hat{a}|a \in A) \qquad [9]$$

where $\eta(\psi|a = \hat{a}) = -\sum_{\hat{a} \in A} p(\psi, a = \hat{a}) \log p(\psi|a = \hat{a})$, and $p(\ )$ denotes probability of an event. The quantity $\eta(\psi|a = \hat{a})$ can be easily computed from our mismatch model (that directly gives us $\log p(\psi|a = \hat{a})$ for each ψ and $\hat{a}$). The prior probability $p(a = \hat{a}|a \in A) = p(a = \hat{a})/p(a \in A)$ for each $\hat{a} \in A$ is derived from the empirical counts of the phones appearing in our phonetically transcribed corpus of Hindi speech.

The transformation from Hindi words to English orthography can be viewed as a noisy channel. The equivocation of the Hindi word string W given the English orthographic string $\Psi = [\psi_1, \ldots]$ is

$$\eta(W|\Psi) = -\sum_{W} \sum_{\Psi} p(W|\Psi) \log p(W|\Psi) \qquad [10]$$

Shannon (1949) proved an extremely useful fact about the number of different Hindi word strings, W, that we need to consider as possibilities given knowledge of the English orthographic string Ψ. Let $N(W|\Psi)$ denote the perplexity of W given Ψ, loosely defined as the number of typical inputs given a particular input. More precisely, if we notice that a "uniform input language" containing exactly $1/p(W|\Psi)$ equally probably strings would assign the probability $p(W|\Psi)$ to each of them, then we can say that $N(W|\Psi) = 2^{\eta(W|\Psi)}$ is the exponential of the average of the log of the number of strings per uniform input language. Shannon (1949), in his Theorem 3, proved that as the length of the input string, W, approaches infinity, all languages approach uniform input: the set of $N(W|\Psi)$ most probable strings each approach equal probability ($p(W|\Psi) \to 1/N(W|\Psi)$ for strings in the typical set), and all other strings approach zero probability ($p(W|\Psi) \to 0$ for strings outside the typical set). In other words, in the limit as sentences become infinitely long, only the first $N(W|\Psi)$ most likely Hindi sentences need to be considered as possibilities; all others have zero probability. The $N(W|\Psi)$ most likely sentences are called, in communication theory, the "typical set" of Hindi sentences given the mismatched English transcription.

References

Adams, Douglas. 1979. *The hitchhiker's guide to the galaxy*. London: Pan Books.

Barron, Andrew R. 1993. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory* 39(3). 930–945.

Barron, Andrew R. 1994. Approximation and estimation bounds for artificial neural networks. *Machine Learning* 14(1). 115–133.

Baume, Eric & David Haussler. 1989. What size net gives valid generalization? *Neural Computation* 1(1). 151–160.

Beckman, Mary E. & Gayle Ayers Elam. 1994. Guidelines for ToBI labelling. Technical report, Ohio State University. http://www.ling.ohio-state.edu/research/phonetics/E_ToBI/singer_tobi.html (accessed 17 September 2015).

Best, Catherine T. 1994. The emergence of native-language phonological influences in infants: A perceptual assimilation model. In J. C. Goodman & H. C. Nusbaum (eds.), *The development of speech perception: The transition from speech sounds to spoken words*, 167–224. Cambridge, MA: MIT Press

Best, Catherine T. 1995. A direct realist view of cross-language speech perception. In W. Strange (ed.), *Speech perception and linguistic experience: Issues in cross-language research*, 171–204. Timonium, MD: York Press.

Best, Catherine T., Gerald W. McRoberts & Nomathemba M. Sithole. 1988. Examination of perceptual reorganization for nonnative speech contrasts: Zulu click discrimination by English-speaking adults and infants. *Journal of Experimental Psychology: Human Perception and Performance* 14(3). 345.

Beygelzimer, Alina, Sanjoy Dasgupta & John Langford. 2009. Importance weighted active learning. In *Proceedings of the 26th International Conference on Machine Learning*. 49–56.

Bienaymé, I.-J. 1853. Considérations à l'appui de la découverte de laplace. *Comptes Rendus de l'Académie des Sciences* 37. 309–324.

Chernoff, Herman. 1952. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics* 23(4). 493–507.

Cieri, Christopher, David Miller & Kevin Walker. 2004. The Fisher corpus: A resource for the next generations of speech-to-text. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*. 69–71.

Cohn, David, Les Atlas & Richard Ladner. 1994. Improving generalization with active learning. *Machine Learning* 15(2). 201–221.

Cole, Jennifer. 2015. Prosody in context: A review. *Language, Cognition and Neuroscience* 30(1–2). 1–31. doi: 10.1080/23273798.2014.963130

Cole, Jennifer, Tim Mahrt & José I. Hualde. 2014. Listening for sound, listening for meaning: Task effects on prosodic transcription. In *Proceedings of Speech Prosody 7*, Dublin.

Cole, Jennifer, Yoonsook Mo & Soondo Baek. 2010a. The role of syntactic structure in guiding prosody perception with ordinary listeners and everyday speech. *Language and Cognitive Processes* 25(7). 1141–1177.

Cole, Jennifer, Yoonsook Mo & Mark Hasegawa-Johnson. 2010b. Signal-based and expectation-based factors in the perception of prosodic prominence. *Journal of Laboratory Phonology* 1(2). 425–452.

Cole, Ronald, Beatrice T. Oshika, Mike Noel, Terri Lander & Mark Fanty. 1994. Labeler agreement in phonetic labeling of continuous speech. In *Proceedings of the International Conference on Spoken Language Processing*, Yokohama, Japan, 2131–2134.

Dasgupta, Sanjoy. 2011. Two faces of active learning. *Theoretical Computer Science* 412(19). 1767–1781.

Davis, Steven & Paul Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28(4). 357–366.

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

Douglas, Shona. 2003. Active learning for classifying phone sequences from unsupervised phonotactic models. In *Proceedings of Human Language Technology Conference North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL 2003)*, 19–21, Edmonton, Alberta, Canada.

Dyrud, Lars O. 2001. *Hindi-Urdu: Stress accent or non-stress accent?* Grand Forks, ND: University of North Dakota Ph.D. thesis.

Eskenazi, Maxine, Gina-Anne Levow, Helen Meng, Gabriel Parent & David Suendermann (eds.). 2013. *Crowdsourcing for speech processing: Applications to data collection, transcription and assessment*. New York: Wiley.

Féry, Caroline. 2010. Indian languages as intonational 'phrase languages'. In S. Imtiaz Hasnain & Shreesh Chaudhury (eds.), *Problematizing language studies: Cultural, theoretical and applied perspectives – Festschrift to honour Ramakant Agnihotri*, 288–312. Delhi: Aakar Books.

Féry, Caroline & Gerrit Kentner. 2010 The prosody of embedded coordinations in German and Hindi. In *Proceedings of Speech Prosody*, 2010.

Fitzpatrick-Cole, Jennifer & Aditi Lahiri. 1997. Focus, intonation and phrasing in Bengali and English. In Georgios Kouroupetroglou, Antonis Botinis & George Carayiannis (eds.), *Intonation: Theory, models and applications. Proceedings of the ESCA Workshop*, 119–122.

Flege, James E. 1987. The production of "new" and "similar" phones in a foreign language: Evidence for the effect of equivalence classification. *Journal of Phonetics* 15(1). 47–65.

Flege, James E. 1995. Second language speech learning: Theory, findings, and problems. In Winifred Strange (ed.), *Speech perception and linguistic experience: Issues in cross-language research*, 233–277. Timonium, MD: York Press.

Flege, James E. 2007. Language contact in bilingualism: Phonetic system interactions. *Laboratory Phonology* 9. 353–382.

Flege, James E., Carlo Schirru & Ian R. A. MacKay. 2003. Interaction between the native and second language phonetic subsystems. *Speech Communication* 40(4). 467–491.

Ganong, William F., III. 1980. Phonetic categorization in auditory word perception. *Journal of Experimental Psychology: Human Perception & Performance* 6(1). 110–125.

Geman, Stuart, Elie Bienenstock & René Doursat. 1994. Neural networks and the bias/variance dilemma. *Neural Computation* 4(1). 1–58.

Genzel, Susanne & Frank Kügler. 2010. The prosodic expression of contrast in Hindi. In *Proceedings of Speech Prosody*, 2010.

Harnsberger, James D. 1994. *Towards an intonational phonology of Hindi*. Gainesville, FL: University of Florida manuscript.

Hayes, Bruce & Aditi Lahiri. 1991. Bengali intonational phonology. *Natural Language & Linguistic Theory* 9(1). 47–96.

Hayes, Bruce & Aditi Lahiri. 1992. Durationally specified intonation in English and Bengali. In R. Carlson, L. Nord & J. Sundberg (eds.), *Proceedings of the 1990 Wenner-Gren Center Conference on Music, Language, Speech and the Brain*, 78–91. Stockholm: Macmillan.

Hermansky, Hynek. 1990. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America* 87(4). 1738–1752.

Jyothi, Preethi, Jennifer Cole, Mark Hasegawa-Johnson & Vandana Puri. 2014. An investigation of prosody in Hindi narrative speech. In *Proceedings of Speech Prosody 2014*.

Jyothi, Preethi & Mark Hasegawa-Johnson. 2015. Acquiring speech transcriptions using mismatched crowdsourcing. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 1263–1269.

Karger, David R., Sewoong Oh & Devavrat Shah. 2011. Iterative learning for reliable crowd-sourcing systems. *Proceedings of Neural Information Processing Systems*, 1953–1961.

Kellogg, Samuel Henry. 1938. *A grammar of the Hindi language*, 3rd edn. London: Lowe and Brydone.

Kim, Heejin, Katie Martin, Mark Hasegawa-Johnson & Adrienne Perlman. 2010. Frequency of consonant articulation errors in dysarthric speech. *Clinical Linguistics & Phonetics* 24(10). 759–770.

Mahrt, Tim. 2013. LMEDS: Language markup and experimental design software. http://prosody.beckman.illinois.edu/lmeds.html (accessed 17 September 2015).

Mahrt, Tim, Jennifer S. Cole, Margaret Fleck & Mark Hasegawa-Johnson. 2012. Modeling speaker variation in cues to prominence using the Bayesian information criterion. In *Proceedings of Speech Prosody 6*, Shanghai.

Mahrt, Tim, Jui-Ting Huang, Yoonsook Mo, Margaret Fleck, Mark Hasegawa-Johnson & Jennifer S. Cole. 2011. Optimal models of prosodic prominence using the Bayesian information criterion. In *Proceedings of Interspeech*, Florence, Italy.

Mason, Winter & Duncan J. Watts. 2009. Financial incentives and the “performance of crowds”. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 77–85. New York: ACM.

Moore, P. R. 1965. *A study of Hindi intonation*. Ann Arbor: University of Michigan Ph.D. thesis.

Nair, Rami. 2001. Acoustic correlates of lexical stress in Hindi. In Anvita Abbi, R. S. Gupta & Ayesha Kidwai (eds.), *Linguistic structure and language dynamics in South Asia – papers from the Proceedings of SALA XVIII Roundtable*, 123–143. Delhi: Motilal Banarsidass.

Norris, Dennis, James M. McQueen, Anne Cutler & Sally Butterfield. 1997. The possible word constraint in the segmentation of continuous speech. *Cognitive Psychology* 34(3). 191–243.

Novotney, Scott & Chris Callison-Burch. 2010. Cheap, fast, and good enough: Automatic speech recognition with non-expert transcription. In *Human Language Technologies: The 2010*

Annual Conference of the North American Chapter of the ACL, 207–215. Los Angeles, CA: Association for Computational Linguistics.

Ohala, Manjari. 1983. *Aspects of Hindi phonology*, vol. 2. Delhi: Motilal Banarsidass.

Ohala, Manjari. 1986. A search for the phonetic correlates of Hindi stress. In Bhadriraju Krishnamurti, Colin Masica & Anjani Sinha (eds.), *South Asian languages: Structure, convergence, and diglossia*, 81–92. Delhi: Motilal Banarsidass.

Olsson, Fredrik. 2009. A literature survey of active machine learning in the context of natural language processing. Technical Report SICS T2009:06. Kista, Sweden: Swedish Institute of Computer Science.

Parent, Gabriel. 2013 Crowdsourcing for speech transcription. In Maxine Eskenazi, Gina-Anne Levow, Helen Meng, Gabriel Parent & David Suendermann (eds.), *Crowdsourcing for speech processing: Applications to data collection, transcription and assessment*. New York: Wiley.

Patil, Umesh, Gerrit Kentner, Anja Gollrad, Frank Kügler, Caroline Féry & Shravan Vasishth. 2008. Focus, word order and intonation in Hindi. *Journal of South Asian Linguistics* 1. 53–70.

Pavlick, Ellie, Matt Post, Ann Irvine, Dmitry Kachaev & Christopher Callison-Burch. 2014. The language demographics of Amazon Mechanical Turk. *Transactions of the Association for Computational Linguistics* 2. 79–92.

Peterson, Gordon E. & Harold L. Barney. 1952. Control methods used in a study of vowels. *Journal of the Acoustical Society of America* 24(2). 175–184.

Phan, Monty. 2007. Buffalax mines twisted translation for YouTube yuks. *Wired Magazine*. http://www.wired.com/entertainment/theweb/news/2007/11/buffalax (accessed 17 September 2015).

Pierrehumbert, Janet. 1981. *The phonology and phonetics of English intonation*. Cambridge, MA: MIT PhD thesis.

Puri, V. 2013. *Intonation in Indian English and Hindi late and simultaneous bilinguals*. Urbana-Champaign, IL: University of Illinois Ph.D. thesis.

Rosenberg, Andrew. 2010. AuToBI: A tool for automatic ToBI annotation. In *Proceedings of Interspeech*.

Shannon, Claude & Warren Weaver. 1949. *The mathematical theory of communication*. Urbana, IL: University of Illinois Press.

Strange, Winifred. 1995. *Speech perception & linguistic experience: Issues in cross-language research*. Timonium, MD: York Press.

Sundaresan, Rajesh. 2006. Guessing under source uncertainty with side information. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, 2434–2440.

Tchebichef, P. 1867. Des valeurs moyennes. *Journal de Mathematiques Pures et Appliquees* 2(12). 177–184.

Tür, Gokhan, Dilek Hakkani-Tür & Robert Schapire. 2005. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication* 45(2). 171–186.

Valiant, L. G. 1984. A theory of the learnable. *Communications of the ACM* 27(11). 1134–1142.

Vapnik, Vladimir. 1998. *Statistical learning theory*. New York: Wiley.

Vapnik, Vladimir N. & Alexey Ya. Chervonenkis. 1971. On the convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications* 16(2). 264–280.

Varshney, Lav R. 2014. Assuring privacy and reliability in crowdsourcing with coding. In *Proceedings of the Information Theory and Applications Workshop (ITA)*, 1–6. doi: 10.1109/ITA.2014.6804213

Vempaty, Aditya, Lav R. Varshney & Pramod K. Varshney. 2014. Reliable crowdsourcing for multi-class labeling using coding theory. *IEEE Journal on Special Topics in Signal Processing* 8(4). 667–679. doi: 10.1109/JSTSP.2014.2316116

Yoon, Su-Youn, Mark Hasegawa-Johnson & Richard Sproat. 2009. Automated pronunciation scoring using confidence scoring and landmark-based SVM. In *Proceedings of Interspeech 2009*.

Zue, V. W., S. Seneff & J. Glass. 1990. Speech database development at MIT: TIMIT and beyond. *Speech Communication* 9. 351–356.