Labeling in the Wild: Crowdsourcing versus Categorical Perception

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

University of Illinois

LabPhon 2014

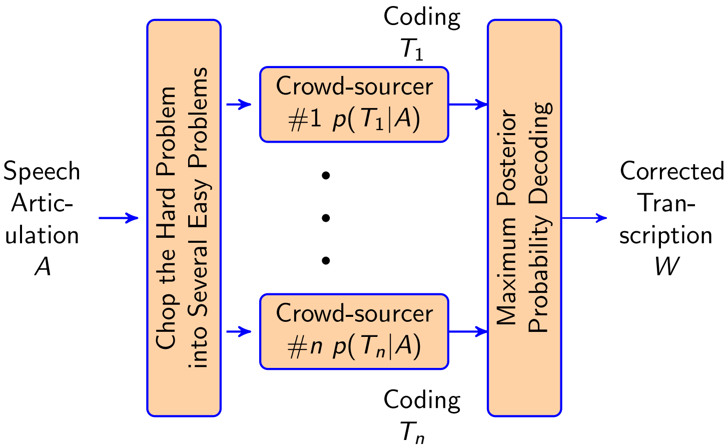
Main Points of This Talk

1. **Crowdsourcing** can give you data cheaply, but crowdsourcers make mistakes. Majority voting reduces error, but triples (or worse) your cost.
2. **Error-correcting codes:** 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.
3. **The science of easy questions:** Factoring a hard problem into easy problems allows you to find out what linguistically naïve crowdsourcers think about hard linguistic questions.
4. **Crowdsourcing versus categorical perception:** Transcription in the wrong language introduces errors. The errors can be modeled using FST models of transcriber cognition.

The Main Problem: Crowdsourcing Introduces Noise

Speech Articulation A $\rightarrow$ **Crowdsourcer is a Noisy Communication Channel $p(W|A)$** $\rightarrow$ Transcriptions W

Proposed Solution: Chop the Hard Problem into Several Easy Problems



Outline

Outline

Sample Problem

Speech recognition fails for Betelgeusians because **they have two heads,** which results in an unusual pronunciation of their vowels. To solve this problem, we would like to learn a classifier that can distinguish /i/ from /e/.

1. Both classes Gaussian w/Identity covariance.
2. Gaussian mixture models (GMM) w/Identity covariance.

A Famous Betelgeusian

(Zaphod Beeblebrox, *Hitchhikers' Guide to the Galaxy*)

Assume Random Training Data, $\mathcal{D}_0$ and $\mathcal{D}_1$

Randomly choose a training sample

$$\begin{array}{lll} \text{From} & \text{Class} & 0: \\ \mathcal{D}_0 & = & \{\vec{x}_1, \ldots, \vec{x}_n\} \\ \text{From} & \text{Class} & 1: \\ \mathcal{D}_1 & = & \{\vec{x}_{n+1}, \ldots, \vec{x}_{2n}\} \end{array}$$

Each $\vec{x}$ is a d-dimensional vector, e.g., cepstrum. Estimate the sample means

$$\hat{\mu}_0 = \frac{1}{n}\sum_{i=1}^{n}\vec{x}_i, \quad \hat{\mu}_1 = \frac{1}{n}\sum_{i=n+1}^{2n}\vec{x}_i$$

Assume a Fixed Testing Datum, $\vec{x}$

$g(\vec{x})$ is the classifier function, e.g.,

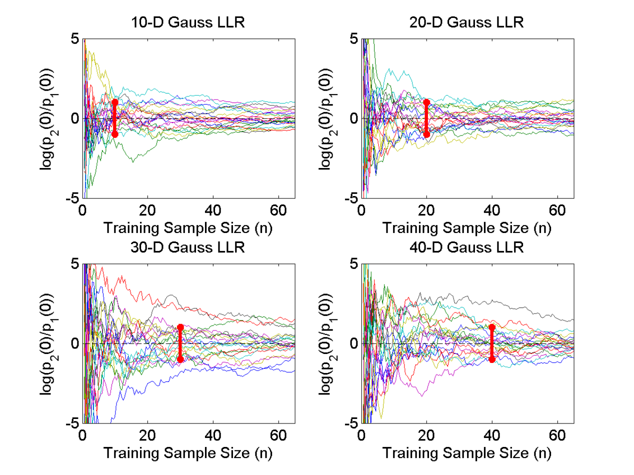
$$g(\vec{x}) = \frac{|\vec{x}-\hat{\mu}_0|^2 - |\vec{x}-\hat{\mu}_1|^2}{2}$$

$$= \vec{x}^T(\hat{\mu}_1 - \hat{\mu}_0) + \frac{|\hat{\mu}_0|^2 - |\hat{\mu}_1|^2}{2}$$

For **random** $\hat{\mu}_0$, $\hat{\mu}_1$ and **fixed** $\vec{x}$, $g(\vec{x})$ is a Gaussian plus the difference of two scaled χ^2 random variables:

$$\sigma_{g(\vec{x})} = \sigma_x\sqrt{\frac{d}{n}}\sqrt{\frac{2|\vec{x}|^2}{d} + \sigma_x^2}$$

d-Dim Classifier Converges Like d Variances



$g(\vec{0})$ as a function of n, multiple random trials. $\sigma_{g(\vec{0})} = \sigma^2_X = 1$ when $n = d$.

Unit Variance Gaussian Mixture Model (GMM)

Test Rule

$$g(\vec{x}) = \ln\left(\frac{\sum_{k=1}^{m} \mathcal{N}\left(\hat{\mu}_{1k}, I\right)}{\sum_{k=1}^{m} \mathcal{N}\left(\hat{\mu}_{0k}, I\right)}\right)$$

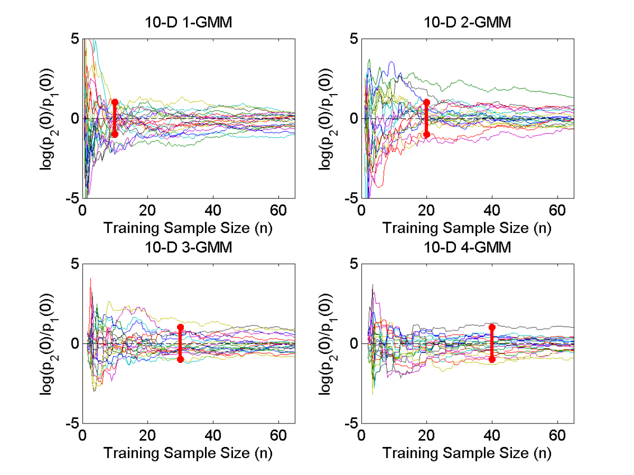
Training Rule

$$\hat{\mu}_{ck} = \frac{1}{n_{ck}} \sum_{\vec{x}_i \in \mathcal{D}_{ck}} \vec{x}_i, \quad 0 \leq c \leq 1,\ 1 \leq k \leq m,\ \sum_{k=1}^{m} n_{ck} = n$$

If $\vec{x}$ is fixed but $\mathcal{D}_{ck}$ are random, then $g(\vec{x})$ is random

$$\frac{g(\vec{x})}{2m\sigma_x^2/n} \sim \chi^2\left(d\right), \qquad \sigma_{g(\vec{x})} \approx \sigma_x^2 \sqrt{\frac{md}{n}}$$

m-GMM Converges like m Gaussians



$g(\vec{0})$ as a function of n, multiple random trials. $\sigma_{g(\vec{x})} = \sigma_x^2 = 1$ when $n = md$.

How Many Labeled Data Are Needed?

To learn a d-dimensional m-GMM, with a classifier function $g(\vec{x})$ that has standard error at most $\epsilon\sigma^2$, we need

$$n \geq \frac{md}{\epsilon^2}$$

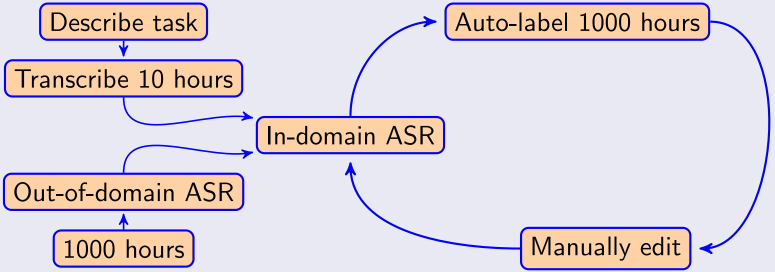
For example, to train a 6-GMM for 40-dimensional cepstra so that $\sigma_{g(\vec{0})} \leq 0.1\sigma_X^2$ requires

$$n \geq 24,000$$

example cepstra (4 minutes of speech) per phone.

- If we have 40 phones represented by exactly 4 minutes of speech per phone, that's 160 minutes (2.67 hours of speech).
- If we have 5000 context-dependent triphones, we need 200,000 minutes (3500 hours).

The Speech Technology Development Cycle



Assumptions

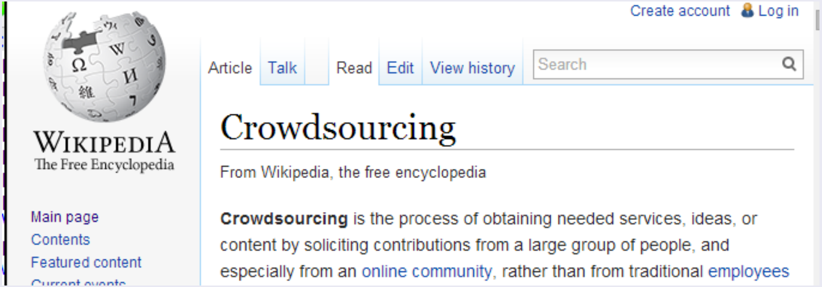
1. Speech is perceived in terms of discrete phonological categories
2. Labelers perceive those categories consistently, as long as. . .
3. Labelers must be drawn from a homogenous linguistic community.

Who are the Labelers?

Source	Motivation	Speed @ Wage
Academic	High	$20 \frac{\text{transcriber hours}}{\text{speech hour}}$ @ \$25/hour
Professional	High	$6 \frac{\text{trainscriber hours}}{\text{speech hour}}$ @ \$30/hour
Crowd	Variable	$600 \frac{\text{hits}}{\text{speech hour}}$ @ \$0.1/hit

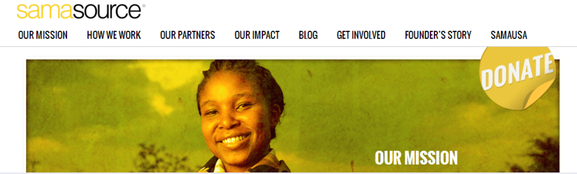
- Cieri et al., "The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text," LREC 2004
- Eskenazi et al., *Crowdsourcing for Speech Processing*, 2013

Crowdsourcing

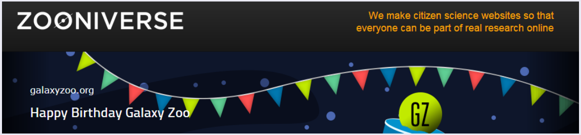


Crowdsourcing sites include big companies. . .

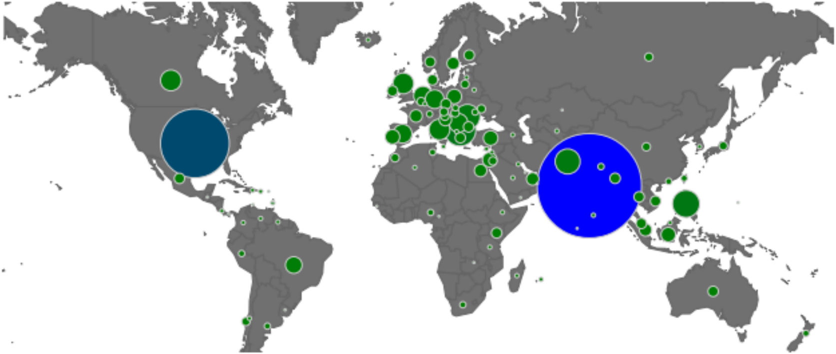
. . . international development organizations. . .



. . . and scientific consortia.



The Language Demographics of Mechanical Turk (Pavlick, Post, Irvine, Kachaev and Callison-Burch, 2013)



Number of workers per country, based on geolocating the IP addresses of 4983 workers. India: 1998, US: 866, Philippines: 142, Egypt: 25, Russia: 10, Sri Lanka: 4. (Pavlick et al., 2013).

Cost, Speed and Quality (Mason and Watts, 2009)

- Payment affects **quantity** of work performed (and **speed**)
- Unexpectedly, payment doesn't affect **quality** of work performed.

Who Turks? (Pavlick et al., 2013)

- USA: mostly people who want a part-time job with scheduling flexibility
- India: mostly full-timers, treat it as a consulting job

Quality Control Methods (Parent, 2011)

1. Before Data Acquisition

   Manual, e.g., choose only workers with good reputation.

   Automatic, e.g., ask a gold standard question, and allow to continue only those who pass.

2. During Data Acquisition (e.g., majority voting)
3. After Data Acquisition

   Manual, e.g., ask other crowdsourcers to validate questionable input.

   Automatic, e.g., get many responses to same question, compare similarity using string edit distance, eliminate outliers

Outline

Main Points of This Talk

1. **Crowdsourcing** can give you data cheaply, but crowdsourcers make mistakes. Majority voting reduces error, but triples (or worse) your cost.
2. **Error-correcting codes:** 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.
3. **The science of easy questions:** Factoring a hard problem into easy problems allows you to find out what linguistically naïve crowdsourcers think about hard linguistic questions.
4. **Crowdsourcing versus categorical perception:** Transcription in the wrong language introduces errors. The errors can be modeled using FST models of transcriber cognition.

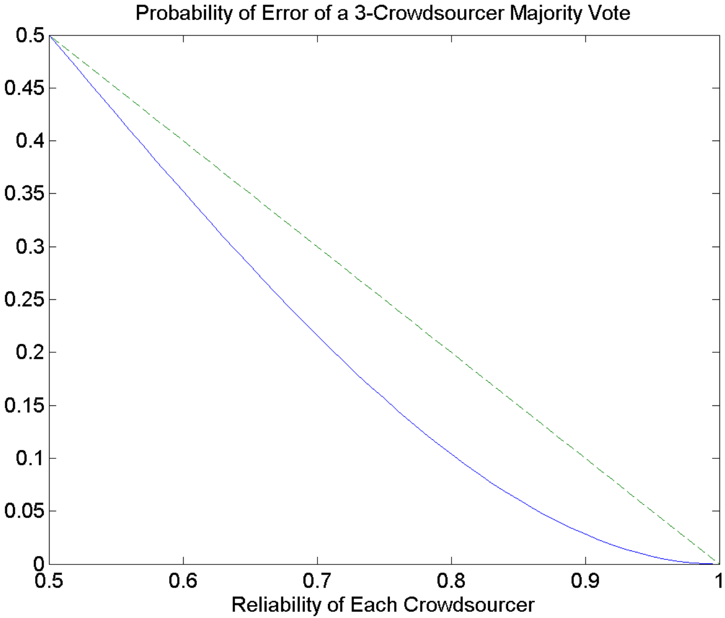
Majority Voting

- Majority voting: assign the same task to ℓ different crowdsourcers. Label the datum with the majority opinion.
- System fails if the majority is wrong. If each crowdsourcer is correct with probability p, then the probability of error is

$$\mathbb{P}_{\text{Error}} = \sum_{k=1}^{\ell/2} \left(\frac{\ell!}{k!(\ell-k)!} \right) p^k (1-p)^{\ell-k}$$

- For example, with $\ell = 3$,

$$\mathbb{P}_{\text{Error}} = 3p(1-p)^2 + (1-p)^3$$


Probability of Error of a 3-Crowdsourcer Majority Vote
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.5
0.6
0.7
0.8
0.9
1
Reliability of Each Crowdsourcer

Weighted Majority Voting (e.g., Karger, Oh & Shah 2011)

- a_{ij} = answer that i^{th} crowdsourcer gave in response to j^{th} question. (Binary: $a_{ij} \in \{-1, 1\}$)
- $p_{ij} = \Pr\{\ i^{\text{th}}$ crowdsourcer is correct about the j^{th} question$\}$. $(0 \leq p_{ij} \leq 1)$
- r_{ij} = "reference opinion" used to determine whether or not a_{ij} is correct. $(-1 \leq r_{ij} \leq 1)$

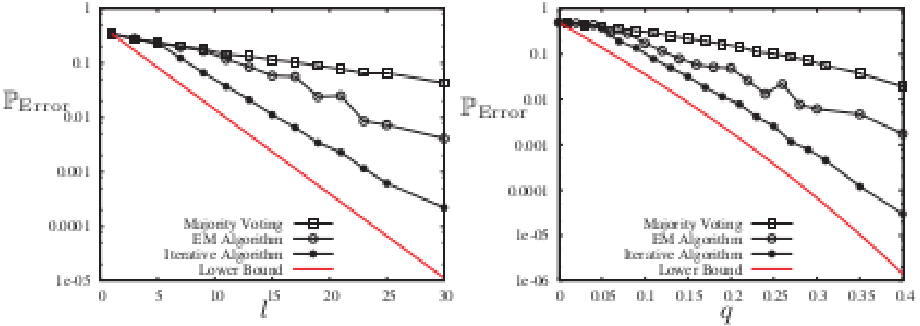
$$r_{ij} \longleftarrow \sum_{k \neq i} a_{kj} p_{kj}$$

$$p_{ij} \longleftarrow \sum_{\ell \neq j} a_{i\ell} r_{i\ell}$$

Iterate until convergence, then compute $r_j = \text{sign}\left(\sum_i a_{ij} \hat{p}_{ij}\right)$, the answer to the j^{th} question.

Weighted Majority Voting is better than Majority Voting (Karger, Oh & Shah, 2011)

- **Theoretical result:** $\mathbb{P}_{\text{Error}} \le e^{-\ell q/\rho^2}$ for
  - $\ell = \#$ crowdsourcers per question
  - $q = E[2p_{ij} - 1]$ =average crowdsourcer reliability
  - $\rho \approx 3$ is a constant term.
- **Empirical result:**



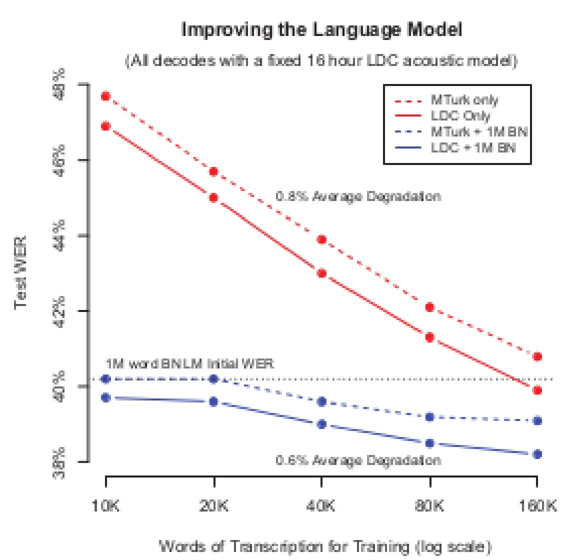
Is Majority Voting Worth the Cost?

Novotney & Callison-Burch (2010) found that

- Training a speech recognizer using crowdsourced transcriptions degrades word error rate (WER) by 2.5%.
- 3-crowdsourcer majority voting results in transcriptions as accurate as LDC, however...
- It's better to have 3× as much data.
- Benefit of extra data outweighs the cost of increased error.

Is Majority Voting Worth the Cost? Example

WER with varying amounts of language model training data, fixed acoustic model (Novotney & Callison-Burch, 2010, Fig. 2).



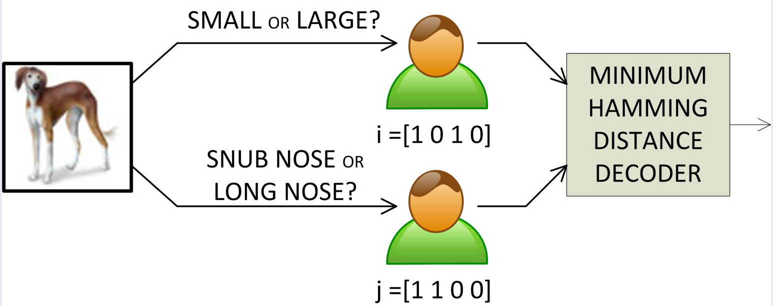
Outline

Main Points of This Talk

1. **Crowdsourcing** can give you data cheaply, but crowdsourcers make mistakes. Majority voting reduces error, but triples (or worse) your cost.
2. **Error-correcting codes:** 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.
3. **The science of easy questions:** Factoring a hard problem into easy problems allows you to find out what linguistically naïve crowdsourcers think about hard linguistic questions.
4. **Crowdsourcing versus categorical perception:** Transcription in the wrong language introduces errors. The errors can be modeled using FST models of transcriber cognition.

Error Control: Hard Question → Easy Questions (Vempaty, Varshney and Varshney, 2014)

Consider the task of classifying a dog image into one of $M = 4$ breeds: H_0 =Pekingese, H_1 =Mastiff, H_2 =Maltese, or H_3 =Saluki. Crowdsourcers may not be canine experts, but can answer simpler questions.



Easy Questions as a form of Error-Correcting Code

- The "hard question" has M possible answers: $1 \leq m \leq M$. Each is equally likely *a priori*: probability $= \frac{1}{M}$
- "Easy questions" are asked of up to ℓ different crowdsourcers, and they give their answers: $a_j =$ answer given by j^{th} crowdsourcer to whatever question he was asked ($a_j \in \{1, -1\}$)
- $c_{mj} =$ answer he should have given if hypothesis m were correct ("code bit" $c_{mj} \in \{1, -1\}$)

Decoding Rule: Choose $\hat{m}$ for

$$\hat{m} = \arg \min_{1 \leq m \leq M} \sum_{j=1}^{\ell} |a_j - c_{mj}|$$

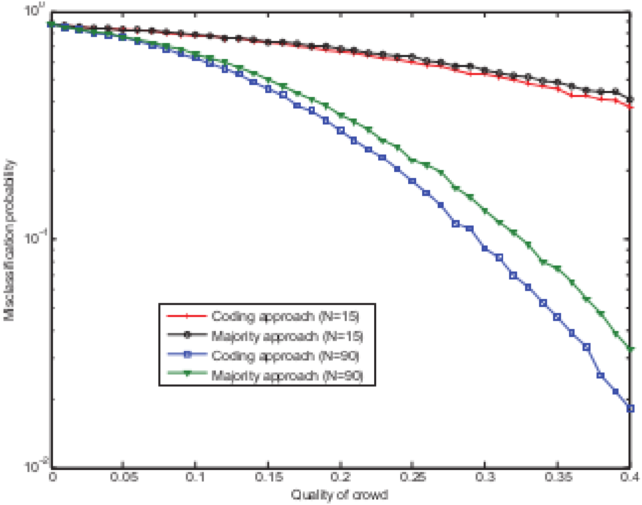
Error-Correcting Code Beats Majority Voting because Even a Wrong Crowdsourcer is Right About Some Things

- Each crowdsourcer answers *easy questions* as though he believes m is the answer to the *hard question*.
- Let $p = \Pr\{\text{crowdsourcer is right about the hard question}\}$
- Let $\frac{1-p}{M-1} = \Pr\{\text{crowdsourcer chooses any particular wrong answer } i \neq m,\ 1 \leq i \leq M\}$

$$1 - \mathbb{P}_{Error} =$$

$$\sum_{m=1}^{M} \frac{1}{M} \sum_{\vec{a}:\hat{m}(\vec{a})=m} \left(\prod_{j=1}^{\ell} \frac{1}{2} \left(1 + a_j \left(p c_{mj} + \frac{1-p}{M-1} \sum_{k \neq m} c_{kj} \right) \right) \right)$$

Error-Correcting Code Beats Majority Voting because Even a Wrong Crowdsourcer is Right About Some Things

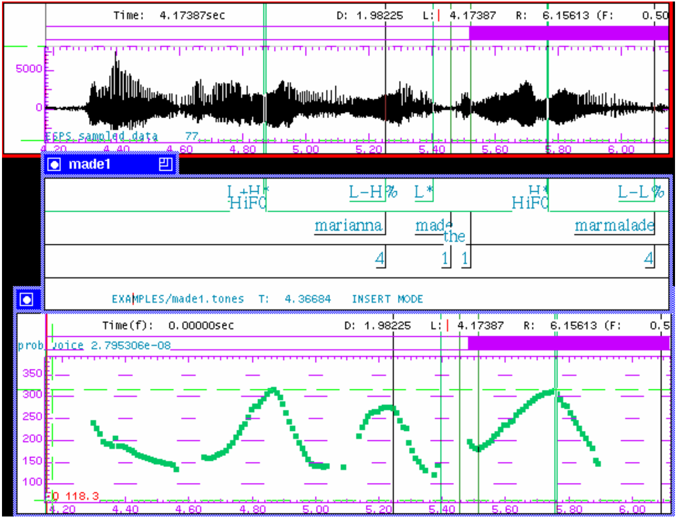


Outline

Main Points of This Talk

1. **Crowdsourcing** can give you data cheaply, but crowdsourcers make mistakes. Majority voting reduces error, but triples (or worse) your cost.
2. **Error-correcting codes:** 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.
3. **The science of easy questions:** Factoring a hard problem into easy problems allows you to find out what linguistically naïve crowdsourcers think about hard linguistic questions.
4. **Crowdsourcing versus categorical perception:** Transcription in the wrong language introduces errors. The errors can be modeled using FST models of transcriber cognition.

A Hard Problem: Prosodic Phonology (ToBI Guidelines: Beckman & Ayers, 1994)



A Hard Problem: Prosodic Phonology

Different ways in which ToBI has been simplified, in order to simplify the training of automatic prosody detection algorithms. From (Escudero-Mancebo, Gónzalez-Ferreras, Vivaracho-Pascual and Cardeñoso-Payo, 2013)

Classification							
Mapping	H*	H*	H*	H*	high	high	high
	L+H*	L+H*	L+H*	L+H*	high	high	high
	!H*	!H*	H*	!H*	downstepped	downstepped	downstepped
	H+!H*	H+!H*	H+!H*	ignored	high	high	high
	L+!H*	L+!H*	L+H*	ignored	downstepped	downstepped	downstepped
	L*	L*	L*	L*	low	low	low
	L*+H	L*+H	L*+H	ignored	low	low	low
	no label	none	ignored	ignored	unaccented	unaccented	unaccented
	#Classes	8	5	4	4	4	4
	Reference	[7]	[8]	[9]	[10]	[11]	[12]
	Level	word	word	word	syllable	syllable	syllable
	#Words/Syllables	27,767	29,578	28,300	14,599	14,599	14,377
	#Speakers	6	6	6	1	1	1
	Accuracy	**70.8%**	**63.99%**	**56.4%**	**80.17%**	**81.3%**	**87.17%**

[7] González-Ferreras et al. (2012) ; [8] Rosenberg (2010) ; [9] Ananthakrishnan and Narayanan (2008b) ; [10] Ross and Ostendorf (1996) ; [11] Levow (2005) ; [12] Sun (2002)

An Easy Problem: Rapid Prosody Transcription (RPT: Cole, Mo & Hasegawa-Johnson, 2010)

Naïve transcribers: Over 100 UIUC undergraduates, non-experts, performed auditory prosody transcription.

Coarse-grain transcription: Transcribers were given only simple definitions of prominence and boundary, and were instructed to mark words where they heard prominence or boundary.

Strength in numbers: Groups of 15-22 subjects transcribe prosody for the same speech excerpts.

Speed: Transcription is done in real-time, with two listening passes per excerpt, based only on auditory impression.

Rapid Prosody Transcription Example

Vertical bars indicate how the speaker breaks up the text into chunks (boundary)

Underline indicates words that are emphasized or stand out relative to other words (prominence)

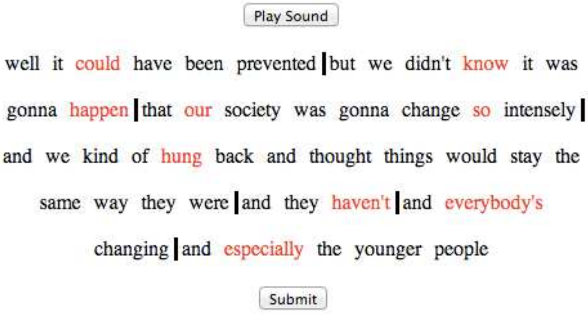
yeah he's not getting that | I dont think he's getting that | learning | he's | he's more his | that's his grandmother | yknow | watching him. . .

yeah he's not getting that I dont think he's getting that learning he's he's more his that's his grandmother yknow watching him. . .

Audio

RPT Example Using LMEDS (LMEDS: Language Markup and Experimental Design Software, Mahrt 2013)

LMEDS screen shot



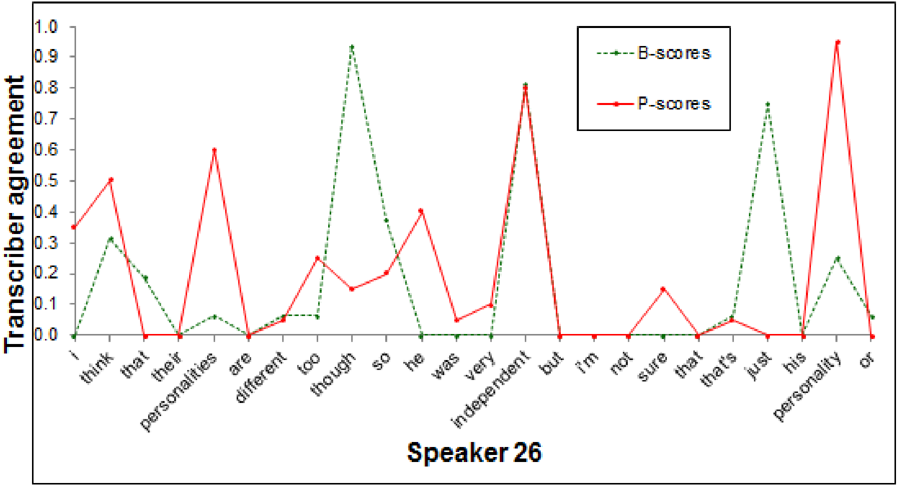
Prosody 'scores'

Each word receives a boundary score (B-score) and a prominence score (P-score).

$$\text{B-score} = T_b / N$$

- $T_b = \#$ of transcribers who marked a boundary following that word
- $N = \text{total } \#$ transcribers
- Similarly, each word receives a prominence score (p-score) indicating how many transcribers marked the word as prominent.

P-scores and B-scores: Example



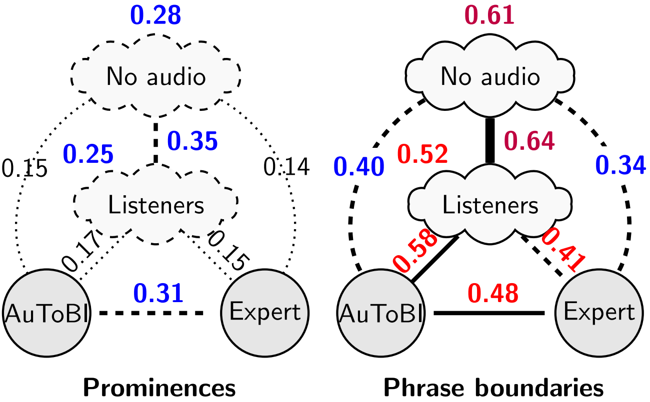
Questions that RPT can ask, but ToBI can't

- Can untrained transcribers label "prosody?"
  Answer: Yes (Cole, Mahrt & Hualde, 2014)
- What are the acoustic and textual correlates of prosodic prominence and boundary, as heard by untrained listeners?
  Some answers: (Cole, Mo & Hasegawa-Johnson, 2010; Cole, Mo & Baek, 2010; Mahrt et al., 2011, 2012)
- Hindi has an F0 movement on each content word, thus English-language models of prominence are largely irrelevant. Does that mean that there is no such thing as prominence in Hindi?
  Results suggest the question is too simple to have a yes/no answer: (Jyothi, Cole & Hasegawa-Johnson, 2014)

An Investigation of Prosody in Hindi Narrative Speech (Jyothi, Cole & Hasegawa-Johnson, 2014)

- Speech data: 10 narrative excerpts in Hindi, about 25 seconds each, from the OGI Multi-language Telephone Speech Corpus
- Transcriptions:
  - RPT with audio: 10 adult speakers of Hindi were asked to mark
    1. how the speaker breaks up the text into chunks (boundary)
    2. words that are emphasized or stand out relative to other words (prominence)
  - RPT without audio
  - ToBI: 1 linguist Ph.D., native speaker of Hindi, ToBI-trained in the USA
  - AuToBI software (Rosenberg, 2010) trained using English-language data

Kappa-score results: Prominence and Boundary



0.1-0.2: Slight agreement
0.2-0.4: Fair agreement
0.4-0.6: Moderate agreement
0.6-0.8: Good agreement

Outline

Main Points of This Talk

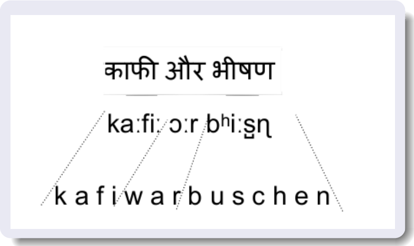
1. **Crowdsourcing** can give you data cheaply, but crowdsourcers make mistakes. Majority voting reduces error, but triples (or worse) your cost.
2. **Error-correcting codes:** If you factor each hard question into several easy (binary) questions, you can improve accuracy more cheaply, because each crowdsourcer only needs to be partially correct.
3. **The science of easy questions:** Factoring a hard problem into easy problems allows you to find out what linguistically naïve crowdsourcers think about hard linguistic questions.
4. **Crowdsourcing versus categorical perception:** Transcription in the wrong language introduces errors. The errors can be modeled using FST models of transcriber cognition.

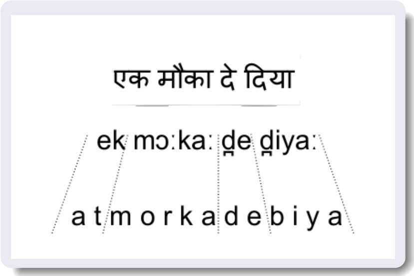
Mismatched Crowdsourcing: Non-Native Transcription

Kalluri vaanil kaayndha nilaavo. . . (Prabhu Deva and Jaya Seal, 2000, as heard by Buffalax=Mike Sutton in 2007)

Examples: Mismatched Transcriptions

Experimental Data, Hindi transcribed as English





Finite State Transducer Models (I)

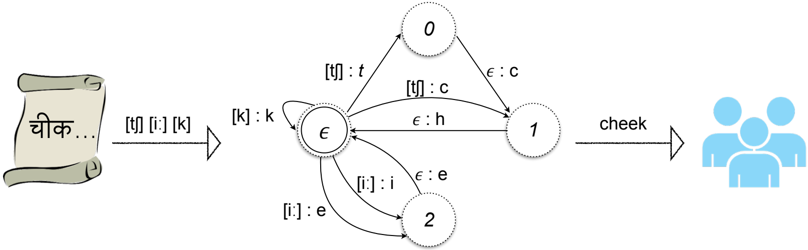
H: Hindi words → Surface phonology WFST $p(A|H)$ → Language mismatch WFST $p(\Psi|A)$ → English dictionary WFST $p(E|\Psi)$ → E: English words

A =Phones articulated

Ψ =Characters perceived

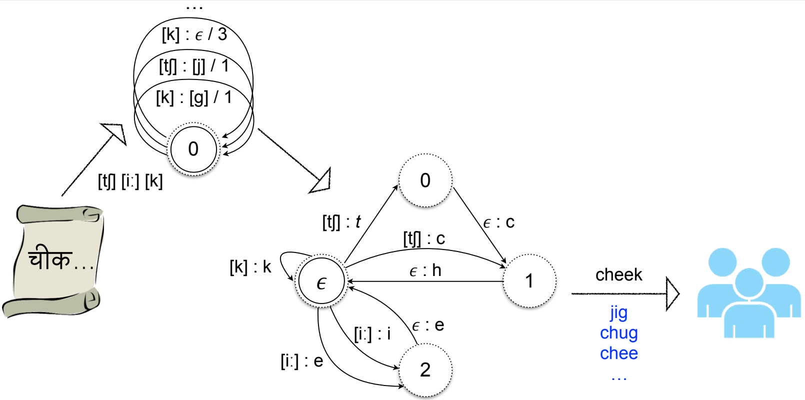
Finite State Transducer Models (II)

Mapping Hindi Words to English Letters



Finite State Transducer Models (III)

Including Hindi Surface Phonology WFST



Estimating the Mismatch FST

In order to estimate the mismatch FST, we need training data.

- A: Fine phonetic transcription by a Hindi-speaking linguist

$$A = [a_1, a_2, \ldots]$$

- Ψ: Ask crowdsourcers to write nonsense syllables instead of English words.

$$\Psi = [\psi_1, \psi_2, \ldots]$$

Mismatched Crowdsourcing: Task Description

Speech materials: Interviews in Hindi from Special Broadcasting Service (SBS, Australia) radio podcasts (mostly spontaneous, formal speech).

Data set: ≈ 52 minutes of data excised from speech of 5 interviewers totaling ≈ 10K words. Transcribed with phonetic labels by a Hindi expert.

Provided to Mechanical Turk workers: Total of 2074 speech excerpts (≈ 2 secs each) with overlapping 0.5 sec segments. Workers asked to transcribe what they hear using nonsense English syllables.

MTurk worker statistics: Total of 68 workers. 40/68 familiar with English only. Other languages familiar to workers mainly included Spanish, Japanese and Chinese.

Mismatched Crowdsourcing: Task Description

Speech materials: Interviews in Hindi from Special Broadcasting Service (SBS, Australia) radio podcasts (mostly spontaneous, formal speech).

Data set: $\approx$ 52 minutes of data excised from speech of 5 interviewers totaling $\approx$ 10K words. Transcribed with phonetic labels by a Hindi expert.

Provided to Mechanical Turk workers: Total of 2074 speech excerpts ($\approx$ 2 secs each) with overlapping 0.5 sec segments. Workers asked to transcribe what they hear using nonsense English syllables.

MTurk worker statistics: Total of 68 workers. 40/68 familiar with English only. Other languages familiar to workers mainly included Spanish, Japanese and Chinese.

Mismatched Crowdsourcing: Task Description

Speech materials: Interviews in Hindi from Special Broadcasting Service (SBS, Australia) radio podcasts (mostly spontaneous, formal speech).

Data set: $\approx$ 52 minutes of data excised from speech of 5 interviewers totaling $\approx$ 10K words. Transcribed with phonetic labels by a Hindi expert.

Provided to Mechanical Turk workers: Total of 2074 speech excerpts ($\approx$ 2 secs each) with overlapping 0.5 sec segments. Workers asked to transcribe what they hear using nonsense English syllables.

MTurk worker statistics: Total of 68 workers. 40/68 familiar with English only. Other languages familiar to workers mainly included Spanish, Japanese and Chinese.

Mismatched Crowdsourcing: Task Description

Speech materials: Interviews in Hindi from Special Broadcasting Service (SBS, Australia) radio podcasts (mostly spontaneous, formal speech).

Data set: $\approx$ 52 minutes of data excised from speech of 5 interviewers totaling $\approx$ 10K words. Transcribed with phonetic labels by a Hindi expert.

Provided to Mechanical Turk workers: Total of 2074 speech excerpts ($\approx$ 2 secs each) with overlapping 0.5 sec segments. Workers asked to transcribe what they hear using nonsense English syllables.

MTurk worker statistics: Total of 68 workers. 40/68 familiar with English only. Other languages familiar to workers mainly included Spanish, Japanese and Chinese.

Structure of the Mismatch FST

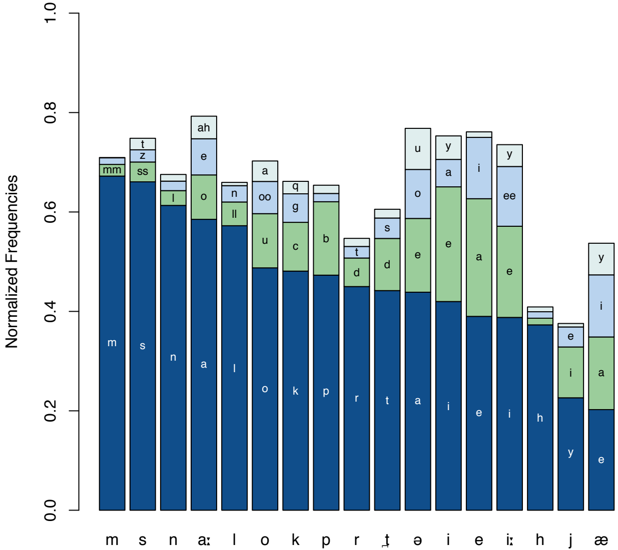
The mismatch FST can be represented as one of these:

- **Distinctive-Feature Weighted Levenshtein Distance:** $-\log p(\Psi|A)$ given by # distinctive feature insertions, deletions, & substitutions from articulated phone string $A = [\ldots, a_t, \ldots]$ to perceived character string $\Psi = [\ldots, \psi_\tau, \ldots]$
- **Learned Levenshtein:** Minimum string-edit distance phone alignment, with substitution costs $\text{SCOST}(a, \psi)$, deletion costs $\text{DCOST}(a)$, and insertion costs $\text{ICOST}(\psi)$ learned from data:

$$-\log p(\Psi|A) \sim \sum_a \sum_\psi \text{SCOST}(a,\psi)\text{NSUBS}(a,\psi)$$

$$+ \sum_a \text{DCOST}(a)\text{NDEL}(a) + \sum_\psi \text{ICOST}(\psi)\text{NINS}(\psi)$$

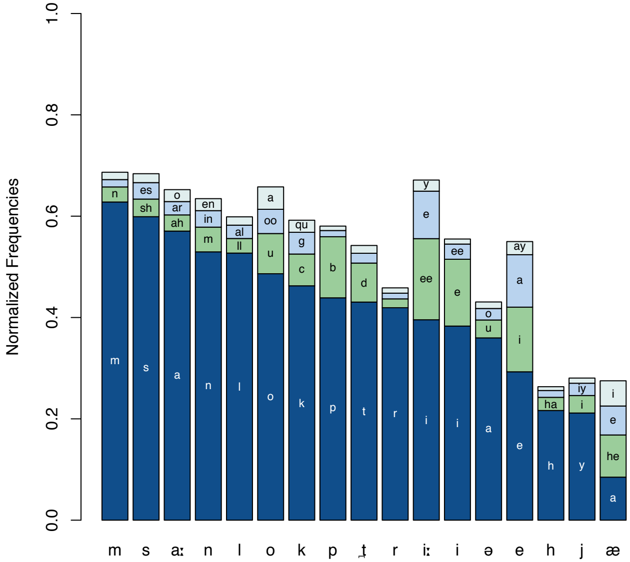
Hindi Sounds Perceived as English Letters (I)



Hindi phones with $\geq$1000 occurrences in the training data

Distinctive Feature-Weighted Levenshtein Mismatch FST: Costs are not learned from data.

Hindi Sounds Perceived as English Letters (II)



Levenshtein Mismatch FST with learned edit costs using the EM algorithm.

Learned Levenshtein Aligns A with Ψ, Allowing us to Compute $p(H|E)$. So what?

Mathematical Theory of Communication (Shannon, 1948)

- Entropy of $H|E$

$$\eta(H|E) = \sum_H \sum_E p(H,E) \log p(H|E)$$

- Perplexity = number of typical inputs given a particular input

$$N(H|E) = 2^{\eta(H|E)}$$

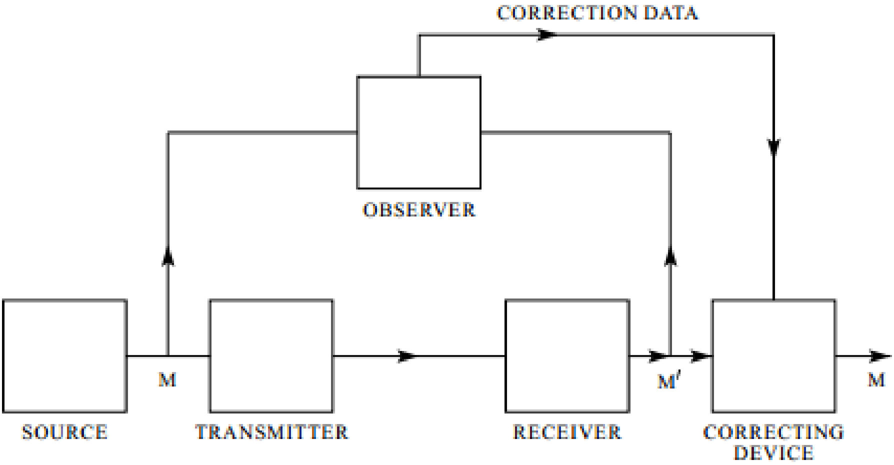
- Shannon, 1948, Theorem 3: As length$(H) \rightarrow \infty$,

$$p\,(H|E) \rightarrow \begin{cases} \frac{1}{N(H|E)} & H \text{ ``typical'' given } E \\ 0 & \text{otherwise} \end{cases}$$

Future Work: Post-Editing of Mismatched Crowdsourcing

- Post-editing by a Hindi-speaking linguist
- Prompt screen lists $N(H|E)+1$ options:
  - $N(H|E)$ Hindi sentences that are most probable given the English transcription
  - 1 option that says "OTHER:" allows linguist to type something different
- Scalability via active learning: editor sees only the transcripts with maximum $\eta(H|E)$

Noisy Channel Correction Model (Shannon, 1948, Fig. 8)



Theoretical Result: Bit Rate of the Side Channel

- Hindi language model gives $p(H)$, from which we calculate Entropy $\eta(H|E)$:

$$\eta(H|E) = \sum_{H,E} p(H,E) \ln p(H|E)$$

$$p(H,E) = \sum_{A} \sum_{\Psi} p(E|\Psi) p(\Psi|A) p(A|H) p(H)$$

- Channel capacity of the side channel is

$$C = \log_2 (\# \text{ Correction Options})$$

- Shannon, 1948, Theorem 11 (The "Fundamental Theorem of Communication"):

$$\text{If } C \geq H(H|E) \text{ then } P(\text{ERROR}) \underset{\text{length}(H)\rightarrow\infty}{\longrightarrow} 0$$

Outline

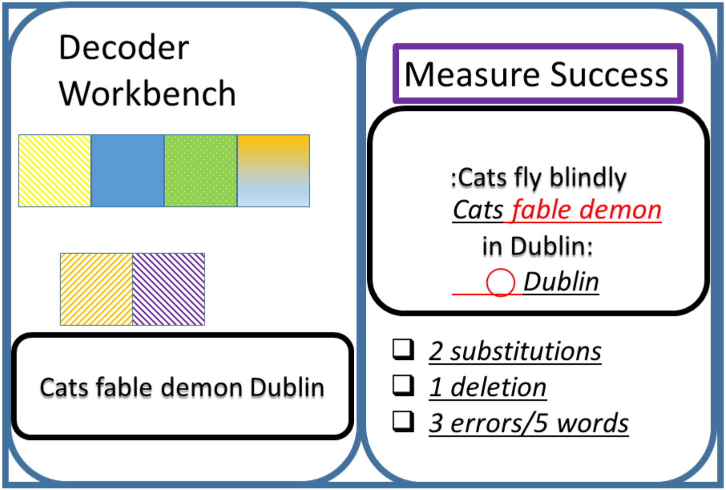
Conclusions

1. **Crowdsourcing** can give you data cheaply, but crowdsourcers make mistakes. Majority voting reduces error, but triples (or worse) your cost.
2. **Error-correcting codes:** 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.
3. **The science of easy questions:** Factoring a hard problem into easy problems allows you to find out what linguistically naïve crowdsourcers think about hard linguistic questions.
4. **Crowdsourcing versus categorical perception:** Transcription in the wrong language introduces errors. The errors can be modeled using FST models of transcriber cognition.

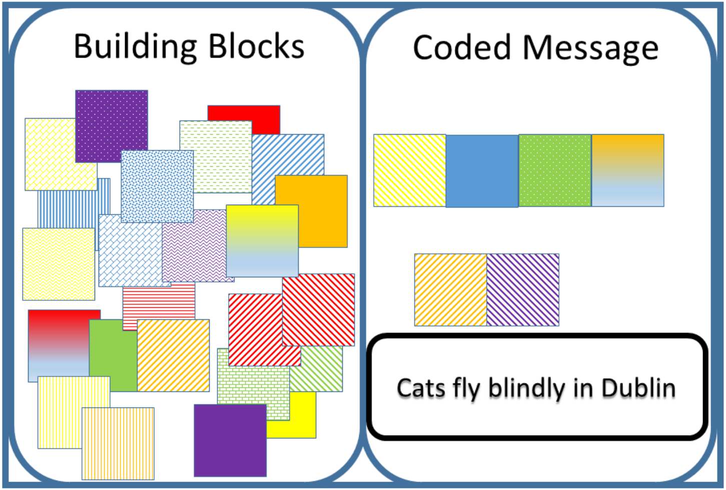
Future Work

- **Further analysis** of the mismatched crowdsourcing model (e.g., "Guessing with side information")
- **Validate** the mismatched crowdsourcing model
- **Scale** using active learning
- **Exploit** mismatched crowdsourcing to build ASR in lots of languages
- **Gamesource** these tasks: write games that bored students will want to play while waiting for the bus.

Example: Secret Agent Game, Decoder Screen



Example: Secret Agent Game, Coder Screen



Thank you!