INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

On Employing a Highly Mismatched Crowd for Speech Transcription

Purushotam Radadia, Rahul Kumar, Kanika Kalra, Shirish Karande and Sachin Lodha

Tata Research Development and Design Center, TCS-Innovation Labs-Pune
(purushotam.radadia, Rahul.14, kanika.kalra, shirish.karande, sachin.lodha)@tcs.com

Abstract

Crowd sourcing provides a cheap and fast way to obtain speech transcriptions. The crowd size available for a task is inversely proportional to the skill requirements. Hence, there has been recent interest in studying the utility of mismatched crowd workers, who provide transcriptions even without knowing the source language. Nevertheless, these studies have required that the worker be capable of providing a transcription in Roman script. We believe that if the script constraint is removed, then countries like India can provide significantly larger crowd base. With this as a motivation, in this paper, we consider transcription of spoken Russian words by a rural Indian crowd that is unfamiliar with Russian and has very limited knowledge of English. The crowd we employ knew Gujarati, Marathi, Telugu and used the scripts of these languages to provide their transcriptions. We utilized an insertion-deletion-substitution channel to model the transcription errors. With a parallel channel model we can easily combine the crowd inputs. We show that the 4 transcriptions in Indic scripts (2 Gujarati, 1 Marathi, 1 Telugu) provide an accuracy of 73.77 (vs. 47% for ROVER algorithm) and a 4-best accuracy of 86.48%, even without employing any worker filtering.

Index Terms: speech transcription, mismatched crowdsourcing, insertion-deletion-substitution channel model

1. Introduction

Manual transcriptions still remain in high demand, either to generate corpus for speech engines or because humans transcribers are resilient to poor speech quality. However, manual transcriptions can be expensive and slow. Hence there is a significant recent interest [1, 2, 3, 4] in using a crowd (non-experts) for such tasks. Typically the crowd consists only of workers who know the source language. This can clearly limit the addressable crowd size. Hence, it is worth investigating whether a mismatched crowd which is unfamiliar to the source language can also be efficiently used for transcription. Jyothi et al. [5] showed this to be the case in a recent study, where a mismatched crowd was used for transcribing Hindi words using Roman script (English).

The population of India is second highest in the world, and an increasing number of Indians are being provided access to the Internet and smart phones. Hence, a huge demographic dividend can be reaped if workers from countries like India can be used for transcription of languages from around the world. Nevertheless, a large percentage of Indians cannot write in English nor any other non-Indian scripts. A script may not be able to code all the phonemes perceived by the worker and may introduce its own biases. Thus it needs to be established whether a highly mismatched crowd that transcribes the foreign language in an Indic script can provide quality transcriptions.

We requested a rural Indian crowd which transcribes Russian words in Gujarati, Marathi and Telugu scripts and has very limited knowledge of English. Our focus in this paper is to combine the multiple noisy transcriptions obtained in different scripts to decode the spoken Russian word. We propose a phoneme level insertion-deletion-substitution (IDS) channel that models the transcription errors. We compare the performance of our method with a pure Finite State Transducer (FST) based approach employed in [5] and a meagre edit-distance based approach reported in our work-in-progress [6]. We observed consistent improvement for various combinations. Particularly an accuracy of 73.77% for combining 4 transcriptions (Guj-Guj-Tel-Mar) and 79.1% for 6 transcriptions (Eng-Eng-Guj-Guj-Tel-Mar) is observed.

The remainder of the paper is organized as follows: In Section 2, we summarize the related work. We describe transcription collection in Section 3. The IDS channel model is explained in Section 4. We evaluate the performance of the proposed approach in Section 5. We conclude in Section 6.

2. Related Work

Recent studies have shown the utility of non-expert transcribers. Marge et al. [2] utilized a non-expert crowd through Amazon Mechanical Turk (MTurk), familiar with the source language, to transcribe the route instructions of the robot. They used ROVER alogrithm [7] to combine multiple transcriptions of a given utterance and showed the reduction in Word Error Rate (WER). Further, Evanini et al. [3] explored various schemes to combine the noisy transcriptions obtained from the non-experts. They utilize various combinations of ROVER, lattice, longest common sub-sequence etc. to achieve the performance gains. Audhkhasi et al. [4], utilizes non-expert crowd to transcribe Spanish audio clips. They showed that the worker reliability scores can be derived from the ROVER based merged transcriptions. Such, unsupervised scores can again be used in merging to further reduce the WERs. All the above studies require the worker to be familiar with the source language. However, Jyothi et al. [5] presented the first study on acquiring the transcriptions of the isolated Hindi words from the crowd unfamiliar with the source language, i.e, *mismatched crowd*. Further they extended their study for continuous speech transcription in [8] and multilingual ASR adaptation in [9] using mismatched transcriptions. We have presented early results on the data used in this paper at [6]. The current work provides significant performance gains over the reported early results.

Since the classical use of Levenshtein's distance [10] several algorithms have been proposed for decoding on insertion-deletion channels (read [11] for a survey of results) and for trace reconstruction (e.g. [12]). These works are mostly motivated for long sequences (e.g. DNA) and hence cannot be readily used for word transcriptions. Further, majority of the work assumes equal error probability for all symbols.

http://dx.doi.org/10.21437/Interspeech.2016-673

Table 1: *Selected Indian crowd base*

Gujarati	49 students (std: 8th-10th), Gujarati medium Govt. school, Pithadiya, Rajkot, Gujarat
Marathi	50 students (std: 7th-9th), Marathi medium municipal school, Pune, Maharashtra
Telugu	23 villagers from Alavalapadu village, Kadapa, Andhra Pradesh
English	31 volunteers from an Indian IT company, Pune, Maharashtra

3. Transcription Collection

Our corpus is a subset of [6]. The speech utterances consist of 250 words isolated from Russian TED talks and 250 phonetically rich words, synthesized by Google, sampled from a Russian Pronunciation dictionary. The pronunciation dictionary contains 0.5 million words and its phonetic decomposition [13]. Out of 500 words, 159 words have less than 5 arpabets, 266 words have 6-13 arpabets and 75 words have 14 or more arpabets. Arpabet is an ASCII letter(s) representing a phoneme IPA symbol. For simplicity, we treat phonemes as arpabets.

The crowd from 4 distinct demographic segments was utilized (Table 1). Six transcriptions (2 Gujarati, 1 Marathi, 1 Telugu, 2 English) for each Russian word were collected. The system allowed worker to listen to a Russian word and type the response in his/her preferred script. Each worker transcribed 15 to 40 words. Table 2 shows an example transcriptions of word 'поворот' (means 'turn', read as '*povorot*').

We map worker responses to Russian phoneme sequences. Figure 1 shows the process of mapping the Marathi transcription to Russian arpabets. The process in Figure 1 can be generalized for any Indic script. In transliteration stage, We convert all the non-English responses to the Roman script (English alphabets). We use the Pramukh transliteration libraries [14] to transliterate the Indic script responses to the Roman script. Next, English transliteration is processed for further error corrections. Since the crowd workers can make grammatical errors in writing, the transliterated English text is impacted. For example, the typical transcriptions of word 'school' can be 'skul' and 'skool'. We alleviate such disagreements by mapping '*oo*' to '*u*' or vice-versa. Yet another mapping rule can be *'ee'* to *'i'* and so on. Thus we follow the simple mapping rules in preprocessing stage. The pre-processed text is then used to create the English letter(s) to a Russian arpabet mapping, Grapheme to Phoneme (G2P), model. Given the parallel corpus of word graphemes and their corresponding phoneme sequences, G2P training utilize EM framework to maximize the grapheme to phoneme mapping probabilities. In our case, the G2P training creates probabilistic mapping between English letters and Russian arpabets. We consider mapping of one and two English letters to each Russian arpabet. We used Carmel Finite State Toolkit [15] to build G2P model. Once the G2P model is trained, it provides multiple Russian phoneme sequences according to its likelihood probabilities with a given input English letter sequence. However, we use the most probable phonetic sequence of the transliterated English word in further decoding.

Table 2: *Worker responses of word* поворот *('povorot')*

English	Gujarati	Marathi	Telugu
tavarot	ફવારોત	तवारोज	తవరోచ
	'favaarot'	*'tavaaroj'*	*'tavaroch'*

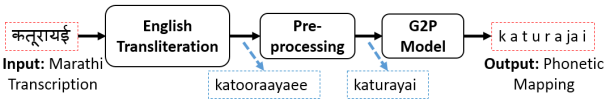

Figure 1: *Mapping worker's transcription to a Russian phoneme sequence*

The Russian dictionary contains total 52 arpabets. However, we observed that removing the arpabets that are similar sounding can improve the decoding performance. Thus we reduce their size down to 33 by representing similar sounding arpabets commonly (e.g. 'ii' and 'i' as 'i', 'yy' and 'y' as 'y' and so on).

4. Model for Transcription Errors

4.1. Insertion-Deletion-Substitution (IDS) Channel Model

We use an insertion-deletion-substitution channel similar to the work of Viswanathan and Swaminathan [12] to model the transcription errors. Let $\mathcal{P}$ be the set of phonemes in the source language (here Russian Arpabets). The channel is then described in probability space as $p(y|x) = \mathcal{P} \cup \{*\} \rightarrow \mathcal{P}^*$, where the channel input x can be any phoneme or a special start symbol $*$. The output y is a string of phonemes. The empty string, $y = \{\}$, represents the deletion of a channel input. Let $q_d(a)$ be the probability of phoneme a being deleted. If $y \neq \{\}$ then the first symbol in the string corresponds to the input symbol. This symbol is susceptible to a substitution error. Let $q_s(a, b)$ be the probability of a being substituted by b . When $y \neq \{\}$ then symbols other than the first symbols represent the insertions. The number of insertions is governed by a geometric random variable (r.v) dependent on the input symbol. Let $g(a)$ be the parameter for the geometric r.v. for the input symbol a. The inserted symbols can take any phoneme value with equal likelihood. Finally, note that while we allow for insertions after the start, we enforce that the start symbol has zero deletion and substitution probability. Multiple transcriptions for a single word are treated as outputs of independent parallel channels.

4.2. Decoding Algorithm

Let $\bar{X} = [X_0, X_1, \cdots, X_m]$ represent a string of phonemes corresponding to a valid Russian word which can be the channel input, meanwhile, let $\bar{Y} = [Y_0, Y_1, \cdots, Y_n]$ be the phoneme corresponding to a transcription which represent the channel output. We enforce that $X_0 = Y_0 = \{*\}$. Assuming uniform priors, the decoding for a given transcription is given by

$$\hat{X} = \max_{\bar{X} \in \mathcal{V}} p(\bar{Y}|\bar{X}) \tag{1}$$

where $\mathcal{V}$ represents a limited vocabulary dictionary.

The dynamic programming used to find the edit-distance between two strings [10] can be easily modified to provide the likelihood $p(\bar{Y}|\bar{X})$. For this purpose, let the log-likelihood of a sub-string match be given by:

$$f_{\bar{X},\bar{Y}}(i,j) = -log\left(p\left([Y_0, Y_1, \cdots, Y_j]\,|\,[X_0, X_1, \cdots, X_i]\right)\right) \tag{2}$$

Thus we can recursively evaluate the likelihoods as:

$$f_{\bar{X},\bar{Y}}(i,j) = \begin{cases} f_{\bar{X},\bar{Y}}(i-1,j) - log((1-g(X_i))q_d(X_i)) \\ f_{\bar{X},\bar{Y}}(i,j-1) - log(\frac{g(X_i)}{|\mathcal{P}|}) \\ f_{\bar{X},\bar{Y}}(i-1,j-1) - log((1-g(X_i))q_s(X_i,Y_j)) \end{cases} \tag{3}$$

Algorithm 1 Supervised IDS Parameter Learning

Input: $\mathcal{A}^0,\ \mathcal{X},\ \mathcal{Y},\ T$
Output: $\mathcal{Q}^T$.
for $t = 1$ to T **do**
  Expectation Step:
    $\mathcal{Q}^t \leftarrow$ estimate model($\mathcal{A}^{t-1}$) (Algorithm 2)
  Maximization Step:
    $\mathcal{A}^t \leftarrow$ estimate alignments($\mathcal{X},\ \mathcal{Y},\ \mathcal{Q}^t$) (Equation 3)
end for

Algorithm 2 Estimate Model Parameters

Input: $\mathcal{A}$
Output: $\mathcal{Q}$
Initialization: $\{C_S(X_i, Y_j),\ C_D(X_i),\ C_I(X_i),\ C_B(X_i)\} \leftarrow 0$
for all aligned pairs $(\bar{X}, \bar{Y}) \in \mathcal{A}$ **do**
- $C_S(X_i, Y_j) \leftarrow$ number of substitutions of X_i by Y_j
- $C_D(X_i) \leftarrow$ number of deletions of X_i
- $C_I(X_i) \leftarrow$ total number of insertions after X_i
- $C_B(X_i) \leftarrow$ number of insertion bursts after X_i
- $q_s(X_i, Y_j) = \frac{C_S(X_i,Y_j)}{\sum_{\forall Y_j} C_S(X_i,Y_j)+C_D(X_i)}$
- $q_d(X_i) = \frac{C_D(X_i)}{\sum_{\forall Y_j} C_S(X_i,Y_j)+C_D(X_i)}$
- $g(X_i) = \frac{C_I(X_i)}{C_I(X_i)+C_B(X_i)}$

end for

On account of a uniform prior and a parallel channel assumption, word decoding using its k transcriptions is given by:

$$\hat{X} = \max_{\bar{X}\in\mathcal{V}} p(\bar{Y}^{(1)}, \cdots, \bar{Y}^{(k)}|\bar{X}) = \max_{\bar{X}\in\mathcal{V}} \prod_{j=1}^{k} p(\bar{Y}^{(j)}|\bar{X}) \quad (4)$$

4.3. Supervised Model Training

The parameters $\mathcal{Q} = \{q_s(a,b), q_d(a), g(a)\}$ of IDS channel model are estimated using EM framework as shown in Algorithm 1. The Algorithm 1 takes inputs of ground truth phoneme sequences, $\mathcal{X}$, its corresponding transcription's phoneme sequences, $\mathcal{Y}$, and the initial alignments, $\mathcal{A}^0$ between them. The initial alignments can be computed using dynamic programming with all edit costs being 1. In expectation step, the algorithm estimates the model parameters using given alignments. In maximization step, it re-estimate alignments using estimated model. It stops upon some convergence criteria or after exhausting all the iterations T. The parameter probability estimation given the alignments is shown in Algorithm 2.

4.4. Unsupervised Model Training

Supervised training requires ground truth labels/sequences, $\mathcal{X}$. However, sometimes such labels may not be available. Hence, we explore the utility of unsupervised training of IDS channel $\mathcal{Q}$. Given a set of transcriptions, $\mathcal{Y} = \{\bar{Y}^{(j)}\}$ and the vocabulary, $\mathcal{V}$, we initially use edit distance algorithm, with uniform cost of 1 for each edit, to find the label sequences, $\mathcal{X} \subset \mathcal{V}$. The normalized alignment cost, c_{ji}, between $\bar{Y}^{(j)}$ and vocabulary word $\bar{X}^{(i)} \in \mathcal{V}$ is used to compute likelihood score. Using $p(\bar{Y}^{(j)}|\bar{X}^{(i)}) = 1 - c_{ji}$ in (4), we obtain the labelled sequences for all transcriptions. The estimated sequences, $\mathcal{X}$, are then used as ground truth to estimate the IDS channel model parameters, $\mathcal{Q}$ using Algorithm 1. The estimated model is further applied to re-estimate the labels. The process repeats until convergence.

Algorithm 3 Unsupervised IDS Parameter Learning

Input: $\mathcal{V},\ \mathcal{Y},\ T,\ T_2$
Output: $\mathcal{Q}^T$.
Initialization: estimate $\mathcal{X}^0$ using $\mathcal{V}$ and $\mathcal{Y}$ as in (4) with uniform edit cost of 1 for all edits as in (3) & estimate $\mathcal{A}^0$ using (3)
for $t = 1$ to T **do**
  $\mathcal{Q}^t \leftarrow$ estimate model $(\mathcal{A}^{t-1}, \mathcal{X}^{t-1}, \mathcal{Y}, T_2)$ (Algorithm 1)
  $\mathcal{X}^t \leftarrow$ estimate labels using $\mathcal{Q}^t$ with (4) & (3)
  $\mathcal{A}^t \leftarrow$ estimate alignments using $\mathcal{X}^t$ & $\mathcal{Y}$ (3)
end for

5. Experimental Results

To evaluate the performance, we considered 5000 word vocabulary, $\mathcal{V}$, sampled from a Russian Pronunciation dictionary including 500 corpus words. Given a set of words and their multiple transcriptions per word, we first obtain the Russian phonetic sequence for each transcription as shown in Figure 1 followed by their alignment (using (3)) with each vocabulary word. Finally, we employ (4) to predict a word from vocabulary $\mathcal{V}$. We report experimental results in terms of accuracy.

5.1. IDS Channel Behavior

Figure 2-4 describes the error distributions observed on our data-set. It can be clearly observed that the error probabilities vary significantly with phoneme. We found the error probabilities are highly correlated across different scripts. Hence, we train a single channel model for all our experiments. Nevertheless, we initially apply smoothing to the insertion, deletion and substitution by assigning a small feasible value to all channel transitions. Further, to reduce number of non-trivial substitution parameters (1089 parameters), we threshold each substitution count over 10 (that results in 15% significant entries) before finding the probabilities.

5.2. Results with supervised learning

We estimated IDS parameters using $T = 10$ in Algorithm 1. Further, we experimented with several proportion of training-testing data. We observed stable performance, i.e., accuracies on testing data converged to one value, for all training set sizes greater than 40%. Thus we use (50-50) proportion of training and testing data for next results. We report average accuracy obtained from 50-fold training-testing process. We compare the performance of the proposed IDS model with the ROVER algorithm and a pure FST based approach shown in [5]. ROVER algorithm is used to merge multiple ASR outputs to reduce overall WERs [7]. It aligns multiple ASR outputs i.e., word sequences of a given utterance using dynamic programming followed by voting a word token from each bin. The typical voting method is to chose most frequent word from each bin. Such majority voting results a merged transcription. Instead of using words as tokens, we consider phonemes as tokens followed by most frequent phone voting scheme in each bin. Table 3 shows that the accuracies obtained by proposed method are higher for individual as well as all combination of transcriptions as compared to ROVER and FST based approach [5]. We observe absolute gain of 15% for Gujarati, 10% for Marathi and 15% for Telugu when compared to ROVER and 12%, 8% and 12% improve-

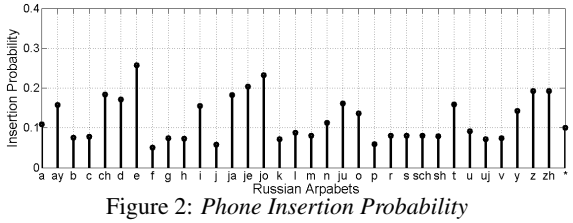

Figure 2: *Phone Insertion Probability*

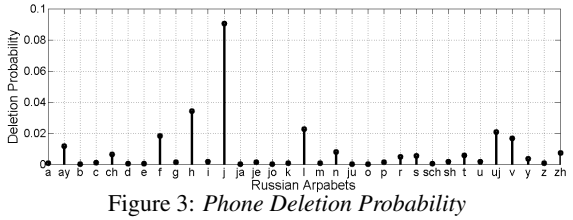

Figure 3: *Phone Deletion Probability*

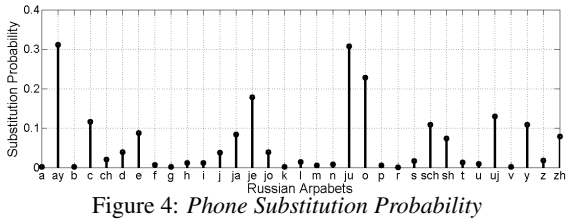

Figure 4: *Phone Substitution Probability*

ment over a pure FST based approach [5]. ROVER due to its structure cannot provide performance gains with just two transcriptions. However, we can clearly say that our model allows for soft-weighting and can thus provide gains of 10-16% over a single transcription. The performance of just two combinations is greater than the best possible performance provided by ROVER (combination of 6 transcriptions.) We observe that we continue get gains as more evidences are combined. We highlight that a transcription each from the three Indic languages are able to provide an accuracy of 69%. Finally, we also bring attention to the 4-Best decoding. We achieve an accuracy greater than 80% with just two mismatched workers. This should encourage the use of a mismatched crowd at least for a validation. Note that the performance of FST based approach [5] degrades when 4-Indic and 6-all responses are combined. This is due to the fact that phoneme insertion bursts are not modeled in [5]. This results in bad alignments when either transcription letters are less than the number of phonemes or transcription is too long compared to its phonetic length. As a result, it achieves very poor score in combining transcriptions using (4). However, our approach explicitly models phoneme edits and hence

Table 3: *Performance comparison: ROVER, [5] and IDS Model*

	ROVER		**FST [5]**		**IDS Model**	
	Best	4Best	Best	4Best	Best	4Best
G	0.34	0.49	0.37	0.46	0.49	0.63
M	0.29	0.42	0.31	0.4	0.39	0.55
T	0.36	0.53	0.39	0.49	0.51	0.68
E	0.44	0.6	0.46	0.51	0.59	0.74
2-G	0.32	0.46	0.43	0.51	0.64	0.79
2-E	0.4	0.57	0.48	0.53	0.71	0.84
G-M	0.25	0.37	0.39	0.48	0.61	0.75
G-T	0.27	0.41	0.43	0.51	0.65	0.81
M-T	0.26	0.41	0.41	0.48	0.60	0.77
G-M-T	0.4	0.55	0.42	0.48	0.69	0.84
4-Ind	0.47	0.65	0.43	0.46	0.74	0.86
6-All	0.55	0.7	0.40	0.42	0.79	0.91

G-Gujarati, M-Marathi, T-Telugu and E-English

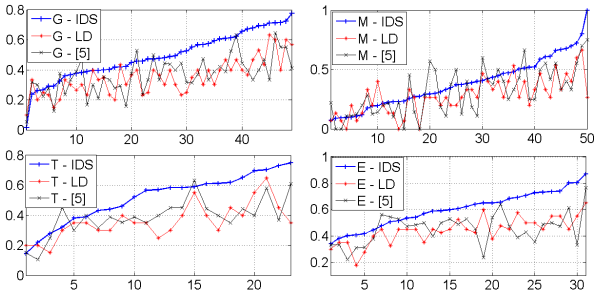

Figure 5: *Worker Accuracy Comparisons (sorted w.r.t. IDS)*

achieves superior performance.

5.2.1. Worker Performance

We also evaluate individual worker's accuracies. Figure 5 shows that the worker's accuracies obtained using IDS channel model is significantly better than Levenshtein distance [6] and pure FST based approach in [5]. We observe on average, 14%, 10.49%, 15.45% and 15.73% absolute improvements compared to [6] and 12.46%, 7.2%, 12.49% and 14.50% improvements compared to FST based approach [5] for Gujarati, Marathi, Telugu and English workers.

5.3. Results with unsupervised learning

Figure 6 shows that unsupervised estimation of IDS parameters (at 5^{th} iteration) provides an accuracy of 78.95% using all 6 responses and 73.52% when using 4 Indic responses (Guj-Guj-Mar-Tel). These accuracies are almost identical to supervised accuracies. It can be observed that the algorithm has a stable performance, i.e. the accuracies converge to stable values. Even one iteration is sufficient to provide significant improvments over an Edit-distance and a pure FST based approaches.

6. Conclusion

We have shown the utility of a highly mismatched crowd in transcribing Russian words. The transcriptions in multi-Indic scripts can be effectively combined to recover the word transcription in a source language. We observe that modeling the transcription errors using IDS channel universally improves the transcription accuracies. We have observed more than 10% gain for individual scripts as well as all possible combinations of transcriptions. We believe that reported accuracies are high enough to show utility in practical settings for employing transcriptions from highly mismatched crowd workers. Our future work directions include considering the (i) phonological aspects between worker's language and source language and (ii) utility of IDS channel model for continuous speech.

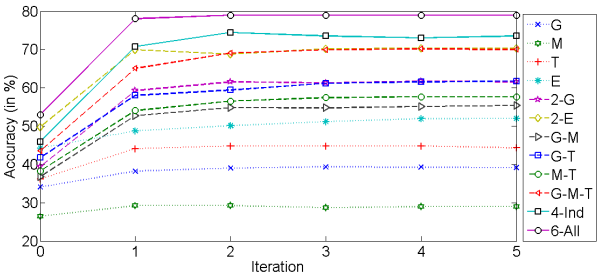

Figure 6: *Unsupervised Accuracies*

7. References

[1] S. Novotney and C. Callison-Burch, "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 Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 207–215.

[2] M. Marge, S. Banerjee, and A. I. Rudnicky, "Using the amazon mechanical turk for transcription of spoken language," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5270–5273.

[3] K. Evanini, D. Higgins, and K. Zechner, "Using amazon mechanical turk for transcription of non-native speech," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Association for Computational Linguistics, 2010, pp. 53–56.

[4] K. Audhkhasi, P. Georgiou, and S. S. Narayanan, "Accurate transcription of broadcast news speech using multiple noisy transcribers and unsupervised reliability metrics," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4980–4983.

[5] P. Jyothi and M. Hasegawa-Johnson, "Acquiring speech transcriptions using mismatched crowdsourcing." in *AAAI*, 2015, pp. 1263–1269.

[6] P. Radadia, S. Karande, and S. Lodha, "On transcribing russian with a highly mismatched indian crowd," in *Third AAAI Conference on Human Computation*, 2015.

[7] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 347–354.

[8] P. Jyothi and M. Hasegawa-Johnson, "Transcribing continuous speech using mismatched crowdsourcing," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] C. Liu, P. Jyothi, H. Tang, V. Manohar, M. Hasegawa-Johnson, and S. Khudanpur, "Adapting asr for under-resourced languages using mismatched transcriptions," 2016.

[10] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[11] M. Mitzenmacher, "A survey of results for deletion channels and related synchronization channels," *Probability Surveys*, vol. 6, pp. 1–33, 2009.

[12] K. Viswanathan and R. Swaminathan, "Improved string reconstruction over insertion-deletion channels," in *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2008, pp. 399–408.

[13] Z. Ostroukhov, "Russian pronunciation dictionary." [Online]. Available: http://sourceforge.net/projects/ cmusphinx/files/Acoustic and LanguageModels/Russian

[14] "Pramukh: Devnagari transliteration engine." [Online]. Available: http://service.vishalon.net/pramukhtypepad.aspx

[15] J. Graehl, "Carmel finite state toolkit." [Online]. Available: http://www.isi.edu/licensed-sw/carmel/