On Localizing Keywords in Continuous Speech using Mismatched Crowd

P. Radadia, T. Madhesia, K. Kalra, A. Sriraman, M. Patwardhan, S. Karande
TCS Research, 54-B Hadpasar Industrial Estate, Pune 411013, India
shirish.karande@tcs.com

Abstract

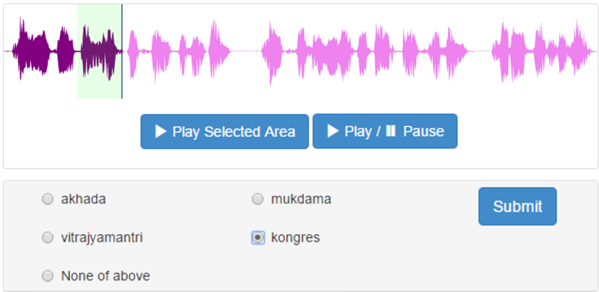
We report results on use of a mismatched crowd in spotting keywords in continuous speech. We observe that unfamiliarity with languages can bias the workers towards declaring non-presence of keywords. We present a joint framework to model the worker bias and reliability of choosing the correct keyword, as well as location. The efficacy of the EM algorithm is demonstrated in identifying the worker parameters and consensus among the annotations. Post aggregation, it can be observed that even a mismatched crowd can provide non-trivial accuracies in spotting keywords.

Introduction

The language demography on crowdsourcing platforms is significantly different from the actual world population (Pavlick et al. 2014). Therefore, recent papers have explored the use of mismatched crowds for speech annotation. (Jyothi and Hasegawa-Johnson 2015b) have shown that, for isolated word recognition, even though the accuracies of individual mismatched workers may be poor, the annotation can be aggregated to provide significantly improved accuracy. Nevertheless, (Jyothi and Hasegawa-Johnson 2015a) observed that accurate annotations in continuous speech is significantly harder . In this paper, we consider a task whose perceptual difficulty lies between the two.

We utilize the mismatched crowd for keyword localization (e.g. see (Sanders, Neville, and Woldorff 2002)) in continuous speech. The task required a worker to listen to a speech utterance and then label the data as follows: (1) Choose a keyword from the list of 5 options. Among them, 4 were words and the fifth option allowed the worker to choose 'None of above'.(2) If the worker choses one of the words then she is required to mark boundaries on the waveform of the utterance. Thus a worker's label consists of keyword identity and its markings (Figure 1). Our setup, while not exactly mapped to a specific usecase, is motivated by low resource scenarios where a keyword spotting engine is desired for a limited dictionary of words. An (in-loop) ASR is not assumed to be available for this *work-in-progress*.

Crowd consensus has been an active research area (Raykar et al. 2010; Zhou et al. 2014). We build upon (Welinder and Perona 2010). Our key contributions are: (1) Demonstration of a non-trivial ability amongst a mismatched crowd to spot and segment word utterances . (2) A framework that explicitly models spammer bias, leading to improved accuracies in label aggregation.



○ akhada
○ mukdama
○ vitrajyamantri
● kongres
○ None of above

Figure 1: Worker's task window where she can listen to the audio multiple times as a whole or the marked segment. This sample task is selected from Hindi language. The word chosen by worker is 'kongres' (represents Congress party of India) and highlighted waveform indicates the markings.

Task Creation and Data Collection

To create tasks, we considered utterances from four languages viz, Arabic, German, Hindi and Russian. The utterances are extracted from online news videos which provided subtitles along with them. The Arabic utterances are extracted from TED talks. The subtitle text of an utterance is used as a ground truth transcription for that utterance. We extracted 50 utterances, for each language, of on average duration of 4.2 second. To generate a task where a keyword is present in the utterance, we picked a random word from the utterance transcription and remaining 3 words are sampled randomly from word corpus specific to the underlying language. We ensured that only one keyword from the options will be present in the utterance. Note that the ground truth markings of the keywords have been generated by native speakers hired from *Upwork*. In case of 'None of above' tasks, we presented the words that have not been spoken in the utterance. We generated an equal number of tasks with a keyword present or absent. Since the mismatched worker may not be familiar with any of the four languages, we transliterated non-roman scripted transcriptions (Arabic, Hindi and Russian in our case) to Roman (English) using Google's read phonetic utility.

We got responses from 100 CrowdFlower workers. Each worker was asked to label 25 tasks. Each task was attempted by 10 different workers under a random alloca-

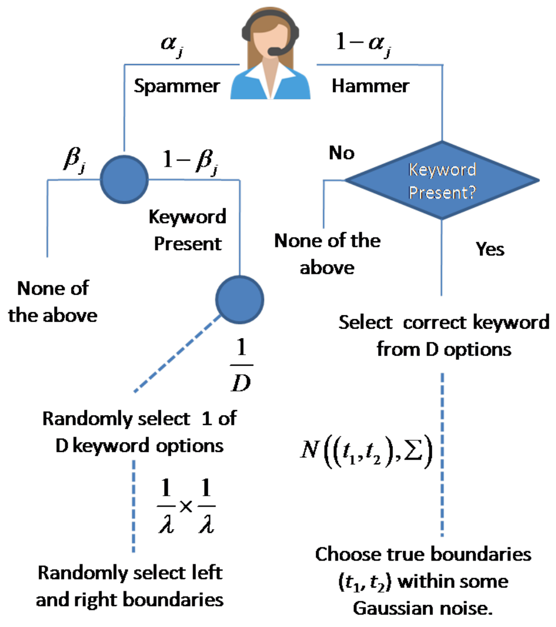

α_j
$1-\alpha_j$
Spammer
Hammer
β_j
$1-\beta_j$
No
Keyword Present?
Keyword Present
None of the above
Yes
None of the above
Select correct keyword from D options
$\frac{1}{D}$
Randomly select 1 of D keyword options
$N((t_1,t_2),\Sigma)$
$\frac{1}{\lambda}\times\frac{1}{\lambda}$
Randomly select left and right boundaries
Choose true boundaries (t_1, t_2) within some Gaussian noise.

Figure 2: The model for worker's annotation behavior

tion.The workers were asked to identify their native language before attempting the task, and responses where the workers were not mismatched have been discarded from the study. The workers were spread across 21 languages e.g. English, Spanish, Czech, Lithuanian, Serbian, Tagalog, Ukrainian etc., however the majority of them identified English (34%) or Spanish (35%) as their native language.

Bias in Crowd Annotations

The overall accuracy of the crowd (in %) for each language is shown in Table 1. We consider the worker's annotation correct only when her choice matches the ground truth and the marking overlaps at least 50% with the ground truth markings. We report accuracy for cases when (1) (KW) keyword is present in utterance, (2) (NKW) none of the word options are present and (3) overall performance. The results from Table 1 indicate a strong bias toward choosing the 'None of above' option. Consequently, in this work, we explicitly model this bias, and demonstrate the utility of it to obtain improved accuracies in aggregation.

Modeling and Label Aggregation

Model for crowd worker

Let the workers be indexed by $j \in \mathcal{W} = \{1, 2, ...M\}$, the tasks be indexed by $i \in \mathcal{T} = \{1, 2, ..., N\}$ and $\mathbf{l}_{ij}$ be the annotation provided by j^{th} worker for the i^{th} task. Moreover, each task i has an underlying target label (to be estimated), $\mathbf{z}_i \in \mathcal{S}_i$. The set $\mathcal{S}_i \subseteq (\mathbb{Z}^+ \times \mathbb{Z}^+ \times \mathcal{O}_i) \cup NK$ where $\mathcal{O}_i$ represents the set of keyword identities in the option list shown to the worker for task i, NK represents the label where keyword is not present in the utterance and the integer tuple represents the boundaries of a marked keyword. We assume that $\mathbf{l}_{ij}$ belong to the same set as $\mathbf{z}_i$.

	Arabic	German	Hindi	Russian
KW	24.82	29.45	27.45	24.81
NKW	71.01	72.51	74.16	73.91
Overall	57.88	48.15	35.81	47.40

Table 1: Language wise performance of crowd

We model the parameters of worker j as $\theta_j = \{\alpha_j, \beta_j\}$, where α_j represents the probability that worker behaves like a spammer. When behaving like a spammer, β_j represents the probability of her bias towards choosing 'None of above' option. The spammer chooses all the remaining keyword option with equal probability. Furthermore, when a spammer chooses a keyword option, he is expected to mark the boundaries. We assume that the spammer randomly chooses the keyword boundaries. On the contrary, when a working is not spamming, we assume it behaves like a Hammer, i.e. if a word is not present in an utterance it always correctly chooses the "None of the above" option, and if a word is present it always chooses the correct option from the list. However, we assume that while identifying the boundaries a slight error may occur. This error is modeled by a bi-variate Gaussian distribution which is common for the entire crowd population. One can refer to Figure 2 to understand the above described notation and the likelihood expression presented in the remainder of the section.

Consider the case when correct keyword is absent but worker chooses the wrong keyword x from the list and provides markings of it on waveform as a_1 and a_2 respectively. The likelihood is given by:

$$p\left(\mathbf{l}_{ij} = (a_1, a_2, x) | \mathbf{z}_i = NK, \theta_j\right) = \alpha_j(1-\beta_j)\frac{1}{\lambda^2}\frac{1}{D} \quad (1)$$

where λ represents the number of samples in the utterance. D represents number of words in the presented option list. The above equation states that worker is spamming but is not inclined towards choosing 'None of above' option. However, since the worker is spamming, we can assume that it choose the keyword options with equal with uniform probability $1/D$ and her spam boundary markings are also governed with uniform probability $1/\lambda^2$. Meanwhile, when a worker rejects all the words when actually there is no keyword in the utterance, the likelihood probability is given by:

$$p\left(\mathbf{l}_{ij} = NK | \mathbf{z}_i = NK, \theta_j\right) = (1-\alpha_j) + \alpha_j\beta_j \quad (2)$$

Above equation states that either the worker did not spam the label (hence probability $1-\alpha$) or might have spammed but been biased to choose the 'None of above' option (probability $\alpha_j\beta_j$). Similarly when worker spams the label given that a keyword is present in the utterance, its likelihood probability is computed as:

$$p\left(\mathbf{l}_{ij} = NK | \mathbf{z}_i = (t_1, t_2, y), \theta_j\right) = \alpha_j\beta_j \quad (3)$$

Further, when the worker does provide a correct label for the keyword this may occur because he is honest, or a spammer has accidentally made the right choice:

$$p\left(\mathbf{l}_{ij} = (a_1, a_2, y) | \mathbf{z}_i = (t_1, t_2, y), \theta_j\right) = (1-\alpha_j)\mathcal{N}((a_1, a_2); (t_1, t_2), \Sigma) + \alpha_j(1-\beta_j)\frac{1}{\lambda^2}\frac{1}{D} \quad (4)$$

We assume that boundary markings by an honest worker are normally distributed about the ground truth.

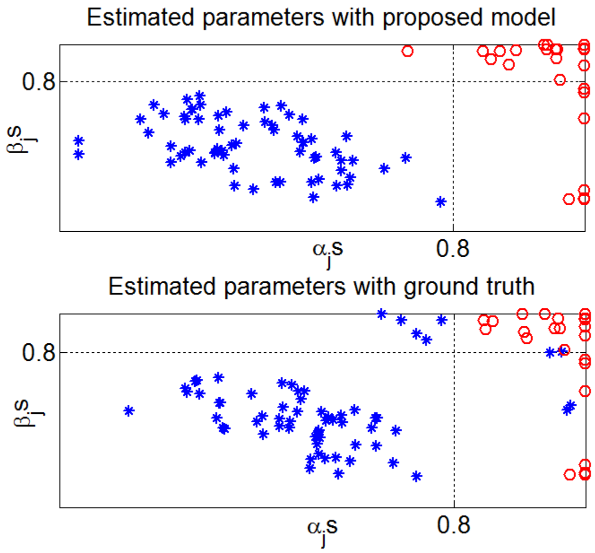


Figure 3: *Comparison of estimated parameters*

Finally, we account for the possibility that provided word choice does not match the groundtruth label. This event can occur only when the worker is spamming:

$$p\left(\mathbf{l}_{ij} = (a_1, a_2, x \neq y)|\mathbf{z}_i = (t_1, t_2, y), \theta_j\right) = \alpha_j(1 - \beta_j)\frac{1}{\lambda^2}\frac{1}{D} \quad (5)$$

EM algorithm for label aggregation

Using the likelihood equations we can setup an EM to estimate the latent true label, $\mathbf{z}_i$, and worker's parameters. We utilize the following notation:Each worker j provides labels $\mathcal{L}^j = \{\mathbf{l}_{ij}\}_{i\in\mathcal{T}_j}$ for the subset $\mathcal{T}_j \subseteq \mathcal{T}$. Similarly, each task i has labels $\mathcal{L}_i = \{\mathbf{l}_{ij}\}_{j\in\mathcal{W}_i}$ provided by subset of workers $\mathcal{W}_i \subseteq \mathcal{W}$. The set of all workers' labels is denoted as $\mathcal{L}$.

E-step: Assuming current estimate of model parameters $\hat{\theta}$, we approximate the posterior on each target value $\mathbf{z}_i$:

$$\hat{p}(\mathbf{z}_i) \propto p(\mathbf{z}_i|\zeta) \prod_{j\in\mathcal{W}_i} p\left(\mathbf{l}_{ij}|\mathbf{z}_i, \theta_j\right) \quad (6)$$

where the label prior ζ has uniform distribution. The probability of target values can be decomposed as follows:

$$\hat{p}(\mathbf{z}_i = (t_1, t_2, y)) = \max_{(t_1,t_2,y)} \left(\frac{\gamma}{\lambda^2 D} \prod_{j\in\mathcal{W}_i} p\left(\mathbf{l}_{ij}|\mathbf{z}_i = (t_1, t_2, y), \theta_j\right)\right) \quad (7)$$

$$\hat{p}(\mathbf{z}_i = NK) = (1 - \gamma) \prod_{j\in\mathcal{W}_i} p\left(\mathbf{l}_{ij}|\mathbf{z}_i = NK, \theta_j\right) \quad (8)$$

where γ is the prior probability of the cases when keyword is present. We estimate the target label as follows:

$$\hat{\mathbf{z}}_i = \arg\max_{\mathbf{z}_i} \left(\hat{p}(\mathbf{z}_i = (t_1, t_2, y)), \hat{p}(\mathbf{z}_i = (NK))\right) \quad (9)$$

To avoid slow sampling, we approximate the posterior on $\mathbf{z}_i$ with delta function (Welinder and Perona 2010),

$$\hat{p}(\mathbf{z}_i) = \delta(\hat{\mathbf{z}}_i) \quad (10)$$

	MV	Multiclass	No-Bias	Proposed
KW	37.27	31.81	46.36	67.27
NKW	94.44	96.66	94.44	80.0
Overall	63.0	61	68	73.0

Table 2: Performance of aggregation over all languages

	Arabic	German	Hindi	Russian
KW	64.28	75	68.29	59.25
NKW	86.11	77.27	77.77	73.91
Overall	80	76	70	66

Table 3: Language wise aggregation accuracy

M-step: The parameters for worker j are estimated by maximizing the expectation of logarithm of posterior on θ_j with respect to $\hat{p}(\mathbf{z}_i)$:

$$\hat{\theta}_j^* = \arg\max_{\theta_j} \left[\log p(\theta_j|\eta) + \sum_{i\in\mathcal{T}_j} \log p\left(\mathbf{l}_{ij}|\mathbf{z}_i, \theta_j\right)\right] \quad (11)$$

where η is represented by mixture of beta distributions.

Performance of Aggregation

Label Aggregation

We compare the proposed model against: (1) Majority Voting (MV), (2) EM for multiclass as described in (Welinder and Perona 2010) and (3) Model without considering bias (i.e. $\beta_j = 0$). Note that baseline methods (1) and (2), rather unfairly, only consider the keyword identity and not the markings while evaluating the accuracies.

Table 2 shows that the proposed model provides a gain of 30%, 35.46% and 20.91% over MV, Multiclass and the unbiased method in terms of overall accuracy. Furthermore, Table 3 shows that there is a significant gain for all languages. The improvement can be attributed to: (1) The location annotation has higher dimension making it resilient to random spam (2) The workforce is biased towards selecting the 'None of above' option. The introduction of the bias parameter is responsible for providing a gain of 5%.

Model Parameter Estimation

Figure 3 compares the worker parameters estimated by the EM against those obtained from the ground truth. It can be observed that overall reliability cannot be the only parameter to characterize the workers. In Figure 3, red points (blue points) represent spam workers (honest annotators) as identified by the proposed EM setup when we use 0.8 as a filtering threshold on α_j, β_j values. It was observed that the EM was able to correctly identify 36 out of 46 spammers.

Future work

The list of options determines the difficult of the task. In practice, there can be great efficiency in employing an ASR in the loop, where, the word options are generated by the ASR engine. We anticipate these words to have greater phonetic proximity compared to our task. We intend to extend our experiments and systems to study the above scenario.

References

Jyothi, P., and Hasegawa-Johnson, M. 2015a. Acquiring speech transcriptions using mismatched crowdsourcing. In *AAAI*, 1263–1269.

Jyothi, P., and Hasegawa-Johnson, M. 2015b. Transcribing continuous speech using mismatched crowdsourcing. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Pavlick, E.; Post, M.; Irvine, A.; Kachaev, D.; and Callison-Burch, C. 2014. The language demographics of amazon mechanical turk. *Transactions of the Association for Computational Linguistics* 2:79–92.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *Journal of Machine Learning Research* 11(Apr):1297–1322.

Sanders, L. D.; Neville, H. J.; and Woldorff, M. G. 2002. Speech segmentation by native and non-native speakersthe use of lexical, syntactic, and stress-pattern cues. *Journal of Speech, Language, and Hearing Research* 45(3):519–530.

Welinder, P., and Perona, P. 2010. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW)*, 25–32. IEEE.

Zhou, D.; Liu, Q.; Platt, J. C.; and Meek, C. 2014. Aggregating ordinal labels from crowds by minimax conditional entropy. In *ICML*, 262–270.