Explainable Modeling of Annotations in Crowdsourcing

An T. Nguyen
University of Texas at Austin
atn@cs.utexas.edu

Matthew Lease
University of Texas at Austin
ml@utexas.edu

Byron C. Wallace
Northeastern University
byron@ccs.neu.edu

ABSTRACT

Aggregation models for improving the quality of annotations collected via crowdsourcing have been widely studied, but far less has been done to explain why annotators make the mistakes that they do. To this end, we propose a joint aggregation and worker clustering model that detects patterns underlying crowd worker labels to characterize varieties of labeling errors. We evaluate our approach on a Named Entity Recognition dataset labeled by Mechanical Turk workers in both a retrospective experiment and a small human study. The former shows that our joint model improves the quality of clusters vs. aggregation followed by clustering. Results of the latter suggest that clusters aid human sense-making in interpreting worker labels and predicting worker mistakes. By enabling better explanation of annotator mistakes, our model creates a new opportunity to help Requesters improve task instructions and to help crowd annotators learn from their mistakes. Source code, data, and supplementary material is shared online.

CCS CONCEPTS

• **Information systems** → **Crowdsourcing**; • **Computing methodologies** → *Machine learning*;

KEYWORDS

Crowdsourcing, Explainable, Clustering

ACM Reference format:
An T. Nguyen, Matthew Lease, and Byron C. Wallace. 2019. Explainable Modeling of Annotations in Crowdsourcing. In *Proceedings of 24th International Conference on Intelligent User Interfaces, Marina del Ray, CA, USA, March 17–20, 2019 (IUI '19),* 6 pages.

IUI '19, March 17–20, 2019, Marina del Ray, CA, USA

ACM ISBN 978-1-4503-6272-6/19/03…$15.00
https://doi.org/10.1145/3301275.3302276

Cluster	Words
1	A.S. Corp. E. El-Watan F.C. German-led O'Donnell S&P S
2	BME BNP CNIEC ETA FRENCH GMT IBCA LAKES LG LME MIT
3	Adrian African Albanian Anatolian Australian Canadian Chinese Classic Egyptian European French Israeli Japanese Korean Palestinian Pascal Peruvian Pivotal Portuguese

Table 1: Three clusters of the words that a worker missed in the Named Entity Recognition task. The first contains words with punctuation; the second includes all-capital words; the last (and largest) cluster contains mostly nationalities. Our method aims to discover these clusters while aggregating labels to explain why workers made mistakes.

https://doi.org/10.1145/3301275.3302276

1 INTRODUCTION

Crowdsourcing has emerged as a standard mechanism for distributing work (such as dataset annotation) at modest to low cost. However, ensuring data quality with crowdsourcing remains a significant challenge, especially on paid microtask platforms such as Mechanical Turk, which provide access to inexpert, remote, and unknown annotators via rudimentary communication channels and limited opportunities for training. The annotation process is largely opaque, with only the final labels being observed.

Prior work has aimed to improve the quality of crowdsourced data. Machine learning approaches have often assumed annotator labels are fixed and focused on modeling annotator reliabilities for better aggregation [2, 3, 14, 18]. In contrast, human-centered approaches have sought to improve the quality of annotator work by improving the task design [1, 7, 15, 19], or recording and visualizing worker behavior [12, 13].

Bridging these largely disparate approaches, we extend the classical Dawid & Skene (DS) aggregation model [2] – trained via Expectation Maximization (EM) – with instance

clustering to automatically detect patterns of annotator errors. Such a model could be used to help task designers identify confusing questions [16] and better understand the kinds of mistakes annotators make [12], or to help workers better understand and learn from their own mistakes [6].

At present, Requesters typically evaluate crowd annotators based on simple statistics such as accuracy, with respect to either a small number of 'gold' annotations or aggregated labels. However, workers may have encountered difficult or ambiguous cases, or had a genuine misunderstanding which the Requester could have rectified if only that misunderstanding had been detected. Our joint aggregation and clustering model improves upon this status quo by summarizing the labels provided by each worker, emphasizing explainability rather than aggregation accuracy.

We focus on Named Entity Recognition (NER), the task of identifying named entities such as people names, locations, or organizations in text. For instance, given the sentence 'Ashwin Ram will give a keynote talk at ACM IUI in Los Angeles', the task for the worker is to annotate 'Ashwin Ram' as a person, 'ACM' as an organization, and 'Los Angeles' as a location. The detected named entities are useful for many downstream tasks such as question answering or translation. In **Table 1**, we show three clusters of mistakes made by a worker in the NER task. Our model identified that this worker tends to miss nationality words as well as some acronyms and words containing punctuation. Discovering such patterns may help a Requester revise task instructions and/or send tailored feedback to the worker.

A User Interface (UI) for Requesters may show the clusters similar to **Table 1**. The UI may also include information such as the context where each word appears, the properties of each cluster, and the estimated accuracy of the worker.

To evaluate our approach, we first perform a retrospective (simulation) experiment showing that our joint model improves the quality of clusters vs. separate aggregation and clustering. We also report a human experiment whose results suggest that the discovered clusters aid human sense-making in interpreting worker labels and predicting future mistakes, compared to a simple list of annotations.

To the best of our knowledge, this is the first work to propose an explainable model for detecting patterns of annotation errors to support sense-making. We focus on the case of crowd annotators in the NER task. However, we note our approach is potentially applicable to labels collected from other annotator populations (e.g., by volunteers or gamified crowdsourcing, student annotators, or even domain experts), and other labeling tasks where features are available for clustering. Our source code, data, and supplementary material are available online[1].

[1]http://github.com/thanhan/explainable-crowd-iui19

2 METHOD

EM is typically used to estimate the parameters of the DS aggregation model [2]. This works by iterating between Expectation (E-step) and Maximization (M-step) steps until convergence. The E-step estimates the true label for each instance, while the M-step estimates the quality of each worker's labels (represented by a 'confusion matrix'). Given true label estimates, instances can be grouped into four 'confusion categories' (assuming binary labels): True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The key idea behind our method is to perform clustering on each of these four groups. The naive approach (which we will use as a baseline) is to perform the clustering after aggregating labels. However, this approach assumes that the aggregated labels are perfect, ignoring label uncertainty.

To improve upon this baseline, we propose integrating the clustering model into the DS aggregation model [2], and we derive a new EM algorithm for model training. Our integrated joint model enables aggregation and clustering components to share true label estimates and detected labeling patterns, thus allowing the former to reflect the uncertainty in the latter.

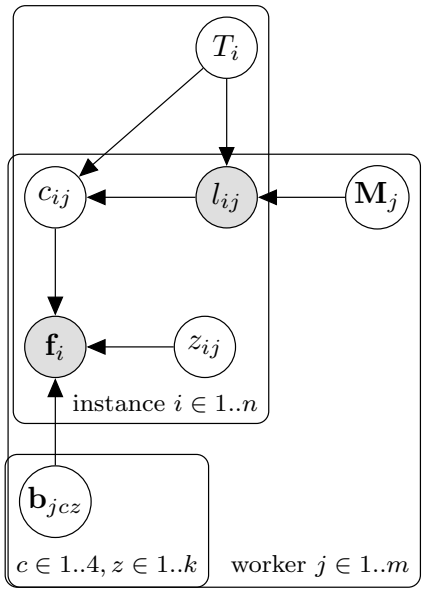
Model

We depict our full graphical model and provide a list of notation in **Figure 1**. We first review the DS aggregation model [2]. Let T_i be the true (unobserved) label for instance i; l_{ij} the crowd label for instance i provided by worker j; and M_j the confusion matrix for worker j. For binary labels ($T_i \in \{0, 1\}$), each confusion matrix is 2×2, and the entry at row x, column y ($M_j[x, y]$) is the probability that worker j provides label y for an instance with true label x:

$$p(l_{ij} \mid T_i, M_j) = M_j[T_i, l_{ij}] \tag{1}$$

We next define the clustering model. Let c_{ij} be the confusion category for the crowd label l_{ij}. Again assuming binary data, c_{ij} assumes one of four values: TP, TN, FP, or FN. Labels provided by worker j can then be partitioned into these four groups. We then cluster the instances in each group to discover the worker labeling patterns. Thus, there are four clustering models for each worker. We denote the parameters of the clustering models associated with worker j by $\mathbf{b}_j$; model parameters are then indexed by the confusion category c_{ij} and the cluster ID $z_{ij} \in 1...k$, where k is the number of clusters. k can be set using information criteria [8] or heuristic metrics such as the silhouette coefficient [11]. However, since we use a small dataset and expect a small number of clusters, we set $k = 3$ for simplicity.

Finally, letting $\mathbf{f}_i$ be the features for instance i and D be the number of features, we assume that the features are binary



Symbol	Description
T_i	true label for instance i
l_{ij}	crowd worker j label for instance i
$\mathbf{M}_j$	confusion matrix for worker j
c_{ij}	confusion category (TP, TN, FP, FN)
$\mathbf{f}_i$	features (binary, D dimensions)
z_{ij}	cluster ID
$\mathbf{b}_{jc}$	cluster parameters

Figure 1: Top: our graphical model. Bottom: the symbols we use and their description. The first three rows are the Dawid & Skene (DS) aggregation model parameters. The remaining refer to our clustering model.

and are generated independently (given the parameters):

$$p(\mathbf{f}_i \mid c_{ij}, z_{ij}, \mathbf{b}_j) = p(\mathbf{f}_i \mid \mathbf{b}_j[c_{ij}, z_{ij}]) \quad (2)$$

$$= \prod_{d=1}^{D} \text{Ber}(\mathbf{f}_i[d] \mid \mathbf{b}_j[c_{ij}, z_{ij}, d]) \quad (3)$$

Where Ber is the probability mass function of the Bernoulli distribution: $\text{Ber}(x \mid p) = p^x(1-p)^{1-x}$. As an example, consider four binary word features: (1) is capitalized, (2) has punctuation, (3) is noun, and, (4) is adjective.

Table 2 shows the cluster parameters for a hypothetical worker in the FN confusion category (FN in the NER task indicates word that is a named entity but which the worker has missed). We see that cluster 1 includes words that have punctuation and nothing else, while cluster 2 includes capitalized words and nothing else. In cluster 3, we expect 10% of the words to be capitalized, 10% to have punctuation, 10% to be nouns, and 90% to be adjectives.

Confusion Cat. c	Cluster ID z	Cluster parameter $\mathbf{b}$
FN	1	[0.0 1.0 0.0 0.0]
FN	2	[1.0 0.0 0.0 0.0]
FN	3	[0.1 0.1 0.1 0.9]
...	...	...

Table 2: Cluster parameters for a hypothetical worker in the False Negative confusion category. The full table would include rows for other confusion categories.

Inference and Learning

We extend the DS model [2], adapting the original EM algorithm [4] for estimation and inference. In the E-step, we perform inference to find the posterior over hidden variables T_i and z_{ij}. In the M-step, we optimize parameters $\mathbf{M}_j$ and $\mathbf{b}_j$ under the expectation found in the last E-step. These two steps are performed iteratively until convergence.

E-step (Inference): Given observed data (l_{ij}, $\mathbf{f}_i$), and model parameters, the inference problem of calculating $p(T_i, z_{ij} \mid l_{ij}, \mathbf{f}_i, \mathbf{M}, \mathbf{b})$ is computationally difficult. We thus resort to approximate inference via Gibbs sampling [5], wherein each variable is sampled conditioned on all others:

$$p(T_i \mid ...) = \prod_j p(l_{ij} \mid T_i, \mathbf{M}_j) p(\mathbf{f}_{ij} \mid c_{ij}, z_{ij}, \mathbf{b}_j)$$

$$p(z_{ij} \mid ...) = p(\mathbf{f}_{ij} \mid c_{ij}, z_{ij}, \mathbf{b}_j)$$

where '...' denotes all other variables and the product $\prod_j$ is over all workers labeling instance i.

M-step: Given the Gibbs samples, the categorical parameters $\mathbf{M}_j$ and worker-specific Bernoulli parameters $\mathbf{b}_j$ can be optimized via Maximum Likelihood Estimation (MLE), which here entails simple counting. Specifically, the confusion matrix entry $\mathbf{M}_j[x, y]$ is proportional to the number of times (over all instances and all Gibbs samples) that the true label is x while the worker j provided label y. The cluster parameter $\mathbf{b}_{jcz}[d]$ is proportional to the number of times that feature d is present in an instance labeled by worker j such that: (1) the instance/label pair is in confusion category c, and; (2) the instance is assigned to cluster z.

3 EVALUATION

Cluster quality experiment

Our method discovers worker annotation patterns by clustering the instances with respect to the *predicted* true labels. Our first experiment measures the extent to which the clusters we discover are consistent with the clusters we would

	Baseline	Proposed
First 1000 words	91.2	92.6
First 10000 words	86.1	88.0
All words	**85.3**	**87.6**

Table 3: Results of the computational experiment; we report F1 scores using different amounts of data.

find if we knew the ground truth gold labels. We partition each worker's labels into TP/TN/FP/FN based on gold labels, then perform clustering (using the proposed mixture of Bernoulli's model) within each set. Because each run of the stochastic clustering method yields slightly different clusters, we generate clusters 100 times and report average results over the 100 trials. In each trial, we compute the Rand index [9] in which two instances belonging to the same cluster are considered positive, while those in different clusters are considered negative. Standard binary classification metrics can then be reported over *pairs* of worker labels: does a given pair assigned to the same cluster induced via gold labels also belong to the same cluster when clusters are estimated via our joint model?

Data: We use the CoNLL NER dataset [17] with crowd labels collected by Rodrigues et al. [10] for 37660 words. We treat the original annotations as reference or 'gold' annotations. We use standard NER word features with no preprocessing: capitalization, Part-or-Speech tags, numbers and punctuation. We also simplify the NER output space into a binary classification problem: words are either part of an entity (1) or not (0).

Baseline: We use DS aggregation [2] to estimate true labels to infer groupings and cluster within these. As noted above, this baseline ignores the inherent uncertainty in the aggregated labels. That is, the model will treat words inferred to be negative (non-entities) with 51% certainty equivalently to tokens inferred to be negative with ∼100% certainty.Discarding this uncertainty will likely yield noisy clusters.

Results: In **Table 3**, we present F1 scores comparing our proposed model to the baseline, averaged over 100 gold clusters. Our first observation is that the F1 scores decrease with more data, likely because larger clusters are harder to predict. The proposed method improves over the baseline, and the improvement is larger with larger data — for example we observe an absolute increase of more than 2 F1 points in the full dataset. We also performed three paired t-tests on the differences between our method and the baseline (one for each row in **Table 3**) and found that these differences are statistically significant: First 1000 words: $t(99)=3.182$, $p<0.002$, First 10000 words: $t(99)=11.537$, $p<0.0001$, All words: $t(99)=19.261$, $p<0.0001$.

Human sense-making experiment

We next report a human experiment to evaluate our approach in practice. To assess the degree to which the discovered clusters are useful in helping individuals ("users" who function as stand-ins for Requesters) understand when and why workers make mistakes, we ran a task on Mechanical Turk. To avoid ambiguity, we refer to those who worked on our task as "users", and those who worked on the original NER task [10] as "workers".

Users were randomly assigned to one of two groups: List and Cluster. The difference between the two groups is:

(1) The List group is shown a *list* of 12 randomly-selected instances correctly annotated by the worker, and 12 that the worker annotated incorrectly (we simplify four confusion categories into just correct and incorrect).
(2) The Cluster group is shown 3 *clusters* of instances that the worker was correct on, and 3 clusters of instances on which they made mistakes. Each cluster c is represented by 4 instances randomly selected from c, its size in number of instances, and a list of majority features for c (that appear in more than 50% instances).

Users in both groups then answer five questions. Each question comprises a pair of instances; one that the given worker provided a *correct* annotation for, and one with an *incorrect* annotation. Users are tasked with predicting which of these the worker made a mistake on. They also have the option to say "I can't tell" if a pair is too difficult. An instance is presented as a sentence with the named entity in bold. Users are advised to spend 5 minutes on the task and are paid $0.50 each. Screen-shots for the task are available in our supplementary material.

Preliminary experiments suggested that this task was very difficult for users. To ease this difficulty, we used a Logistic Regression (LR) classifier to predict the "easiest" questions for the "easiest" workers. The classifier is trained to predict the correctness of each worker's labels (i.e., to do the task), with the NER features enumerated above. The "easiest" questions are those the LR classifier is most certain. The "easiest" workers are those that the LR classifier achieved the highest accuracy on. Our final data for this experiment contains 8 "easiest" workers and 5 "easiest" questions each.

We collected 1335 answers from 111 users and found that the average accuracy of the List group is 51.8%, while the average accuracy of the Cluster group is higher at 61.6%. In calculating these accuracies, a "I can't tell" answer is considered a wrong answer (awarding partial credit for this answer produces similar results). In **Figure 2**, we further show the accuracies with respect to the number of users' answers. The general observation is that the cluster group has better performance. To assess the statistical significance

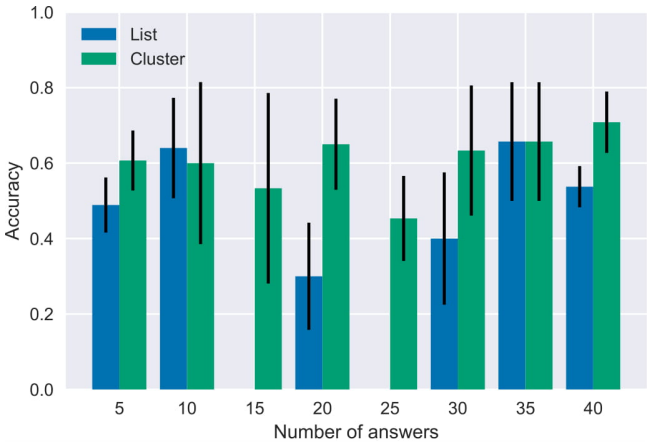


Figure 2: Accuracy vs. the number of answers given by each user. Bernoulli 95% confidence intervals over all answers in each group are reported.

of this difference, we fit a Generalized Linear Model (GLM) where the dependent variable was the label correctness (i.e., whether the user was able to guess correctly which instance the worker annotated correctly). The fixed effect was the group and the random effect was the user (there is a parameter for the group and for each user in the model). We find that the parameter for group is non-zero (p-value 0.0417, meaning the difference between the list group and the cluster group is significant at the 0.05 level).

4 DISCUSSION

Limitations

This work is a first step towards more explainable crowdsourcing models, which have the potential to benefit both Requesters and workers. We note several important limitations. First, our human experiment is a difficult task for Mechanical Turk workers, who need to 'think like a Requester' for the NER task, which they are likely not familiar with. Future work may address this using experiments with Requesters or experiments on the workers of the original NER task (or other appropriate tasks). Second, in our human experiment, we resorted to considering only the "easiest" workers and questions, which means the results may not generalize beyond these examples. Third, some of the clusters discovered by our method are hard to interpret and may not correspond to meaningful annotation patterns. The reason why workers make mistakes can be more complex than what can be observed in the data. Future work may consider richer annotation pattern representations.

Conclusions

Crowdsourcing has been widely used to collect labels for datasets, but to the best of our knowledge, this is the first work to explicitly attempt to improve the explanation of worker mistakes, especially to inform Requesters. To achieve this, we proposed a joint aggregation and clustering model summarizing 'annotation categories' (e.g., false positives) specific to each worker as a generative model over interpretable discrete features.

We first validated that this approach improved clustering quality (using reference standard expert annotations). Secondly, we performed a prospective sense-making experiment that demonstrated that users could better predict the instances on which particular workers might make mistakes given the cluster output from our model. Overall, our results suggest that models for better explaining worker mistakes constitute a promising direction for further research.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and the area chair for their time in reviewing this paper. We also thank the many workers who participated in our and Rodrigues et al. [10]'s experiments. This work is supported in part by National Science Foundation grant No. 1253413. Any opinions, findings, and conclusions or recommendations expressed by the authors are entirely their own and do not represent those of the sponsoring agencies.

REFERENCES

[1] Nancy Chang, Praveen Paritosh, David Huynh, and Collin Baker. 2015. Scaling Semantic Frame Annotation. In *Proceedings of The 9th Linguistic Annotation Workshop*. Association for Computational Linguistics, Denver, Colorado, USA, 1–10. http://www.aclweb.org/anthology/W15-1601

[2] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics* (1979), 20–28.

[3] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 469–478.

[4] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), 1–38.

[5] Stuart Geman and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), 721–741.

[6] Hyun Joon Jung and Matthew Lease. 2013. UT Austin in the TREC 2012 Crowdsourcing Track's Image Relevance Assessment Task. In *Proceedings of the 21st NIST Text Retrieval Conference (TREC)*.

[7] Angli Liu, Stephen Soderland, Jonathan Bragg, Christopher H. Lin, Xiao Ling, and Daniel S. Weld. 2016. Effective Crowd Annotation for Relation Extraction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational

Linguistics, San Diego, California, 897–906. http://www.aclweb.org/anthology/N16-1104

[8] Dan Pelleg, Andrew W Moore, et al. 2000. X-means: Extending k-means with efficient estimation of the number of clusters.. In *Icml*, Vol. 1. 727–734.

[9] William M Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* 66, 336 (1971), 846–850.

[10] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. Sequence labeling with multiple annotators. *Machine learning* 95, 2 (2014), 165–181.

[11] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[12] Jeffrey Rzeszotarski and Aniket Kittur. 2012. CrowdScape: interactively visualizing user behavior and output. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 55–62.

[13] Jeffrey M Rzeszotarski and Aniket Kittur. 2011. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 13–22.

[14] Aashish Sheshadri and Matthew Lease. 2013. Square: A benchmark for research on computing crowd consensus. In *First AAAI Conference on Human Computation and Crowdsourcing*.

[15] Jean Y. Song, Raymond Fok, Alan Lundgard, Fan Yang, Juho Kim, and Walter S. Lasecki. 2018. Two Tools Are Better Than One: Tool Diversity As a Means of Improving Aggregate Crowd Performance. In *23rd International Conference on Intelligent User Interfaces (IUI '18)*. ACM, New York, NY, USA, 559–570. https://doi.org/10.1145/3172944.3172948

[16] Yuandong Tian and Jun Zhu. 2012. Learning from crowds in the presence of schools of thought. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 226–234.

[17] Erik F Tjong Kim Sang and Fien De Meulder. 2003. *Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition*. 142–147. http://aclweb.org/anthology/W03-0419

[18] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. 2010. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*. 2424–2432.

[19] MH Wu and AJ Quinn. 2017. Confusing the crowd: task instruction quality on Amazon Mechanical Turk. In *Proceedings of the 5th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. 149–158.