Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022, pages 95–99
Marseille, 20 June 2022

Improving Label Quality by Joint Probabilistic Modeling of Items and Annotators

Tharindu Cyril Weerasooriya, Alexander G. Ororbia, Christopher M. Homan
Department of Computer Science
Rochester Institute of Technology, USA
cyriltcw@gmail.com,{ago, cmh}@cs.rit.edu

Abstract

We propose a fully Bayesian framework for learning ground truth labels from noisy annotators. Our framework ensures scalability by factoring a generative, Bayesian soft clustering model over label distributions into the classic David and Skene joint annotator-data model. Earlier research along these lines has neither fully incorporated label distributions nor explored clustering by annotators only or data only. Our framework incorporates all of these properties within a graphical model designed to provide better ground truth estimates of annotator responses as input to *any* black box supervised learning algorithm. We conduct supervised learning experiments with variations of our models and compare them to the performance of several baseline models.

Keywords: modeling annotators, graphical models

1. Introduction

The recent interest in few- and zero-shot learning as well as the re-emergence of weakly supervised learning speaks to the reality that ground truth labels are a limited resource and that, in many common situations, obtaining them remains a major challenge. Multiple sources estimate the global costs of human annotators (only one of many sources of labels) to be approaching \$1–3 billion by 2026 and growing (Metz, 2019; Research, 2020). Among the key cost-driving challenges is the noise that is associated with many of the most common processes for obtaining labels.

In this paper, we explore a novel graphical model that ties together two rather successful approaches, item-annotators tableaus (Dawid and Skene, 1979) and label distribution learning (LDL) (Geng, 2016), based on converging studies in later research (Venanzi et al., 2014; Liu et al., 2019a) on the use of clustering to boost the signal of noisy data. We adopt a theoretical framework motivated by the anthropologist Malinowski (Malinowski, 1967) and first used by Aroyo and Welty (Aroyo and Welty, 2014) in the context of machine learning to characterize meaning as a function of three components: 1) an act (represented by the learning task), 2) the symbols (the labels), and, 3) the referent (the annotators). Human labeling is a special challenge not only due to its great expense but also due to the fact that humans often disagree over the labels that they provide. In fact, it is precisely the problems where disagreement is most common that human input is hardest to replace through automation or sensing.

This paper specifically addresses the following question: *do predictive graphical models for LDL that cluster on both item AND annotator distributions outperform those that do not?* To help us answer this question, we contribute a generative graphical model that boosts conventional label distribution learning by clustering label distributions jointly in item and annotator label distribution spaces. Previous approaches have studied clustering in one space or the other. This is, to our knowledge, the first time that clustering has been applied simultaneously to both.

We evaluate the improved labels produced by our model with a downstream CNN-based classification [1]. We view this work as a universally applicable framework for any learning task where annotators are involved (Gordon et al., 2022).

2. Problem Statement

Let $\boldsymbol{X}$ be an M-element collection of (unlabeled) *data items* and $\boldsymbol{Y} \in \mathbb{N}^{M \times N}$ be a matrix of *annotator labels* for some N, where each row of $\boldsymbol{Y}$ corresponds to a data item and each column to an *annotator*. Ideally, we would regard each entry $\boldsymbol{Y}_{m,n}$ as a probability distribution over a set of labels $\{1, \ldots, P\}$ for some fixed P, where the distribution represents uncertainty about what label annotator n would provide to item m. Here, however, we simplify the model under the assumption that each annotator either provides a single label or none at all.

For our purposes, $\boldsymbol{Y}$ is a sparse matrix, where $\boldsymbol{Y}_{m,n} \in \{0, \ldots, P\}$ and $\boldsymbol{Y}_{m,n} = 0$ indicates that annotator n did not label item m. Crucially, we assume that each annotator *could* label the item if asked; however, we have no information about that particular annotator. Since this is a sparse matrix, it is convenient to simply let $A = \{(m, n) \mid \boldsymbol{Y}_{m,n} \neq 0\}$ and $A_p = \{(m, n) \mid \boldsymbol{Y}_{m,n} = p\}$.

We consider two gold standards: f_{dist} and $f_{\max}$, defined for data item $\boldsymbol{X}_m$ as $f_{\text{dist}}(\boldsymbol{X}_m) =_{\text{def}} \mathrm{P}(p = Y_{m,n} \mid m, Y_{m,n} > 0)$, for m, n chosen uniformly

[1] The experimental code available through `https://github.com/Homan-Lab/ldl-pgm`

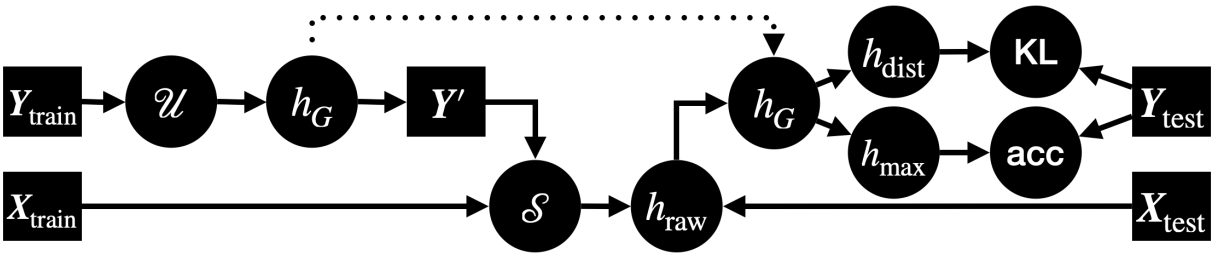

Figure 1: This workflow diagram shows the dual roles of the graphical model h_G, as the output of a supervised learning process $\mathcal{U}$ on the training labels. This model is used to improve the ground truth estimations $\boldsymbol{Y}'$ of the gold standard training label distributions $\boldsymbol{Y}_{\text{train}}$ for supervised learning $\mathcal{S}$ and, once h_{raw} is learned, as a post-processing step after prediction to generate final hypotheses h_{dist} and h_{max}. Evaluation metrics include the accuracy on the most likely label for single label prediction h_{max} and the KL divergence for label distribution learning h_{dist}.

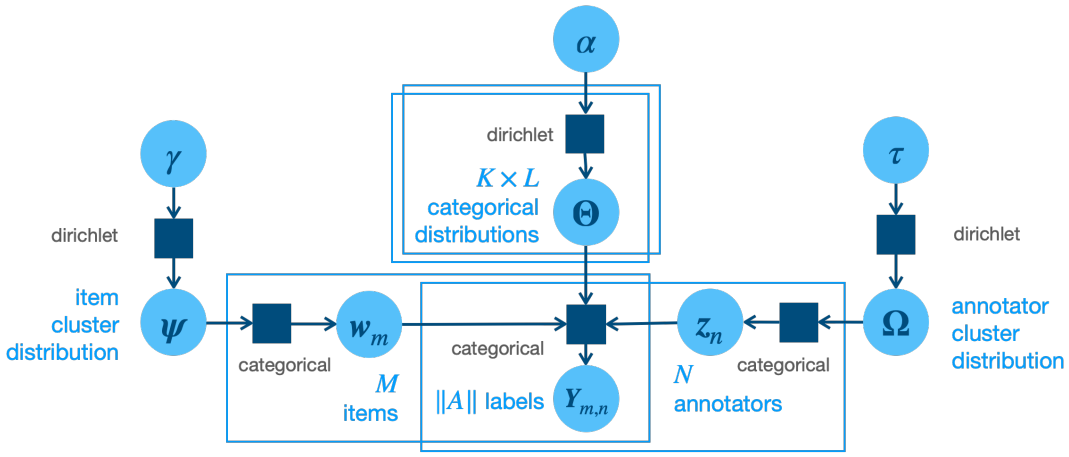

Figure 2: Plate diagram for the proposed probabilistic generative graphical model.

at random and $f_{\text{max}}(\boldsymbol{X}_m) =_{\text{def}} \arg\max_p \mathrm{P}(p = Y_{m,n} \mid m, Y_{m,n} > 0)$. In other words, $f_{\text{dist}}(\boldsymbol{X}_m)$ represents the gold standard *label distribution* associated with each data item and f_{max} is the gold standard single label that is most likely according to f_{dist}. Note that f_{max} is more commonly used than f_{dist}.

Our learning goals, then, are to produce hypotheses h_{dist} and h_{max} that approximate f_{dist} and f_{max}, respectively, given $\boldsymbol{X}$ and $\boldsymbol{Y}$. Most learning settings tacitly assume that annotator disagreement is a sign of *noise* or *error* and ignore d_{dist} entirely. *Label distribution learning* does the opposite: it assumes that annotator disagreement is *meaningful* and specifically seeks to minimize the loss between h_{dist} and f_{dist}. Obviously, both approaches rely on extreme assumptions that, in practice, are never entirely true. However, research has shown that even when f_{max} is the goal, learning h_{dist} and then taking $h_{\text{max}}(\boldsymbol{X}_m) =_{\text{def}} \arg\max_p \mathrm{P}(h_{\text{dist}}(\boldsymbol{X}_m) = p)$ often provides better results than learning h_{max} directly (Venanzi et al., 2014; Liu et al., 2019a; Weerasooriya et al., 2020), and this is what we do here.

3. The Probabilistic Graphical Model

We call f_{dist} and f_{max} gold standards, not ground truths, because of the sparseness of $\boldsymbol{Y}$. Although several researchers have shown that, for the purpose of estimating f_{max}, three to ten annotators is sufficient (Callison-Burch, 2009; Denkowski and Lavie, 2010), those numbers are far too small to provide reliable samples of the true distributions of annotator opinions. In this section, we introduce a new graphical model that estimates the ground truth label distribution, i.e., the distribution of labels from the entire population of annotators, of each item (which we normally do not have). This model is based on the assumptions that: (1) all data items (respectively, annotators) are drawn from one of K (respectively, L) latent classes[2] or *clusters*, (2) the label distribution for each item is strictly a function of the cluster to which it belongs, (3) the sample of labels given for each item is strictly a function of the distribution of the cluster to which each annotator belongs, and (4) the items and annotators are identically and independently sampled (i.i.d.) and matched uniformly at random.

We then use the graphical model h_G to guide supervised learning as a means of data regularization (see

[2]Hereafter, to reduce confusion, we reserve "class" to refer only to the different label choices, as they typically represent an observable class to which the data item belongs, even though the idea of labels as indivisible classes runs contrary to the spirit of LDL.

Algorithm 1 The generative process for h_G.

Input: Integers K, L, M, N, and P; Dirichlet hyperparameters $\alpha \in \mathbb{R}^P$, $\gamma \in \mathbb{R}^K$, and $\tau \in \mathbb{R}^L$, assignments $A \subseteq \{1, \dots M\} \times \{1, \dots N\}$
function GENGRAPH(K, L, M, N, P, α, γ, τ)
    Choose $\boldsymbol{\Theta} \sim \mathrm{Dir}_P(\alpha)^{K \times L}$, ▷ One distribution for each item/annotator cluster pair (k, l)
    Choose $\boldsymbol{\psi} \sim \mathrm{Dir}_K(\gamma)$, ▷ Distribution of item clusters
    Choose $\boldsymbol{\Omega} \sim \mathrm{Dir}_L(\tau)$, ▷ Distribution of annotator clusters
    Choose $\boldsymbol{w} \sim \mathrm{Cat}_K(\boldsymbol{\psi})^M$, ▷ Assign one latent cluster to each item
    Choose $\boldsymbol{z} \sim \mathrm{Cat}_L(\boldsymbol{\Omega})^N$, ▷ Assign one latent cluster to each annotator
    Choose $\boldsymbol{Y} \sim \bigtimes_{(m,n) \in \boldsymbol{A}} \mathrm{Cat}_P(\boldsymbol{\Theta}_{\boldsymbol{w}_m, \boldsymbol{z}_n})$. ▷ Assign labels according to each annotator, item assignment

Figure 1). We first use it as a preprocessing step to supervised learning on our label matrix Y, by reassigning to each input m the generating distribution of the most likely item cluster. Note that any supervised learning method can work as the target so long as it can use a distribution of labels and the supervising signal. For instance, in our experiments (see Section 4) we use a combination of deep language models and simple dense networks. Next, after the predictive model h_{dist} is learned, we post-process each prediction by snapping each output $h_{\mathrm{dist}}(\boldsymbol{X}_m)$ to the most likely item cluster.

Algorithm 1 describes the model from a generative perspective (see also Figure 2). In addition to the numbers of item and annotator clusters K and L, the model takes three hyperparameters, $\alpha \in \mathbb{R}^P$ (recall that P is the number of label classes), $\gamma \in \mathbb{R}^K$, and $\tau \in \mathbb{R}^L$, each of which represents a Dirichlet prior on a categorical distribution. It produces $\boldsymbol{\Theta}_{k,l}$ (the label distribution for each item cluster k and annotator cluster l), $\boldsymbol{\psi}$ (the marginal class distribution of items), and $\boldsymbol{\Omega}$ (the marginal class distribution of annotators). $\boldsymbol{w}_n$ is the hidden/latent variable representing the class of item m and $\boldsymbol{z}_n$ is the hidden variable representing the class of annotator n. Each of these objects is a categorical distribution, and so, for convenience, we use subscripts to indicate individual categorical probabilities, e.g., $\boldsymbol{\Theta}_{k,l,p} = \mathrm{P}(\text{The category is } p)$ and $\boldsymbol{\Omega}_l = \mathrm{P}(\text{The category is } l)$.

Note that our distributions are conditioned on $\boldsymbol{A}$, i.e., we always know beforehand which annotators are assigned to which items. Unfortunately, the coupling between items and annotators makes exact inference hard and even resistant to variational approximation. It is, however, relatively easy to perform simulated annealing over the parameters $\boldsymbol{\Theta}$, $\boldsymbol{\psi}$, $\boldsymbol{\Omega}$ and latent variables $\boldsymbol{w}$, as well as $\boldsymbol{z}$. In addition, we may also employ expectation-maximization (EM), specifically using belief propagation to estimate the probability distributions of $\boldsymbol{w}$ and $\boldsymbol{z}$ during the expectation phase. We explore both learning algorithms here.

dataset	# annotators per item	# label classes	mean entropy	# of annotators
JQ1	10	5	0.746	1185
JQ2	10	5	0.586	1185
JQ3	10	12	0.993	1185

Table 1: Summary of datasets on which we conduct our experiments. Each of these contain 2000 items.

We now describe, in more detail, how we use the model. We partition our data into training $(\boldsymbol{X}_{\mathrm{train}}, \boldsymbol{Y}_{\mathrm{train}})$, development $(\boldsymbol{X}_{\mathrm{dev}}, \boldsymbol{Y}_{\mathrm{dev}})$, and test $(\boldsymbol{X}_{\mathrm{test}}, \boldsymbol{Y}_{\mathrm{test}})$ splits. During training, we first apply one of our two unsupervised learning algorithms $h_G = \mathcal{U}(\boldsymbol{Y}_{\mathrm{train}})$ to learn a graphical model $h_G = (\boldsymbol{\Theta}, \boldsymbol{\psi}, \boldsymbol{\Omega}, \boldsymbol{w}, \boldsymbol{z})$ from $\boldsymbol{Y}_{\mathrm{train}}$. Note that this provides estimates $\boldsymbol{w}$ of the latent item cluster to which each item belongs (simulated annealing provides a hard clustering while EM provides a soft clustering, but with EM we consider only the most likely cluster). Then, before supervised learning, we replace row $\boldsymbol{Y}_{\mathrm{test},m}$ with the marginal label distribution associated with item cluster $\boldsymbol{w}_m$,

$$\boldsymbol{Y}'_m = \sum_l \boldsymbol{\Omega}_l \boldsymbol{\Theta}_{\boldsymbol{w}_m l}, \tag{1}$$

and perform supervised learning $h_{\mathrm{raw}} = \mathcal{S}(\boldsymbol{X}_{\mathrm{train}}, \boldsymbol{Y}')$, yielding a raw label distribution learning predictor. Note that $\boldsymbol{Y}'$ is not a matrix of annotator labels, as $\boldsymbol{Y}_{\mathrm{train}}$ is, but a vector of probability distributions over labels.

For inference *after* training (i.e., we do not perform this step during training), for any input $\boldsymbol{x}$ we project the output of $h_{\mathrm{raw}}(\boldsymbol{x})$ onto our graphical model h_G to predict the item cluster membership of item x, i.e., let $\mathrm{w}(\boldsymbol{x})$ denote a random variable for the item cluster assignment of $\boldsymbol{x}$. Then, we do the following:

$$\mathrm{P}(\mathrm{w}(\boldsymbol{x}) = k) \sim \sum_l \boldsymbol{\psi}_k \boldsymbol{\Omega}_l \mathrm{P}\left(h_{\mathrm{raw}}(\boldsymbol{x}) \sim \mathrm{Cat}_P(\boldsymbol{\Theta}_{k,l})\right) \tag{2}$$

We then assign to x the item cluster $\arg\max_k \mathrm{P}(\mathrm{w}(\boldsymbol{x}) = k)$, using Equation (1) to compute $h_{\mathrm{dist}}(\boldsymbol{x})$ and define $h_{\max}(\boldsymbol{x}) =_{\mathrm{def}} \arg\max_p \mathrm{P}\left(h_{\mathrm{dist}}(\boldsymbol{x}) = p\right)$.

4. Experiments

4.1. Data

We conducted our experiments on publicly available human-annotated datasets. Each dataset consists of 2000 social media posts and employs a 50/25/25 percent for the train/dev/test split.

Liu et al. (Liu et al., 2016)[3] asked five annotators each from MTurk and FigureEight to label work-related

[3] https://github.com/Homan-Lab/pldl_data

Dataset	CNN	MM + CNN	DS + CNN	CL	PGM (Annealing)	PGM (BP)
KL-Divergence ↓						
JQ1	1.092±0.004	**0.460±0.001**	1.042 ± 0.005	2.077 ± 0.003	0.652±0.005	0.538±0.010
JQ2	1.088±0.003	**0.514±0.002**	1.035 ± 0.003	1.695 ± 0.003	0.884±0.004	0.624±0.017
JQ3	1.462±0.004	**0.888±0.001**	3.197 ± 0.034	3.862 ± 0.001	1.201±0.005	0.951±0.016
Accuracy ↑						
JQ1	0.494±0.001	**0.842 ± 0.001**	0.684 ± 0.004	0.813 ± 0.005	0.730±0.000	0.727±0.007
JQ2	0.475±0.001	0.810 ± 0.002	0.658 ± 0.003	**0.873 ± 0.003**	0.579±0.041	0.663±0.013
JQ3	0.284±0.020	0.456 ± 0.010	0.061 ± 0.031	**0.458 ± 0.005**	0.290±0.002	0.250±0.007

Table 2: Experimental results for classification. New methods (PGM) using the development set for each dataset. CNN is a baseline where only a CNN classifier is run.Predictions are compared against the empirical ground truth.

tweets according to three questions and associated multiple choice responses: point of view of the tweet (**JQ1**: *1st person*, *2nd person*, *3rd person*, *unclear*, or *not job related*), subject's employment status (with 17 response options).
We train and test on the following models:

CNN is a 1D convolutional neural network (Kim, 2014) with no unsupervised graphical model. It contains three convolution/max pool layers followed by a dropout and softmax layer, implemented via TensorFlow (Abadi et al., 2015). We used sentence embeddings from the pretrained `paraphrase-MiniLM-L6-v2` BERT model (Reimers and Gurevych, 2019).

MM + CNN is the baseline model with the best-performing graph-based model from (Weerasooriya et al., 2020) used as a guiding model, in a manner analogous to the use of our graph model introduced earlier in this paper. The main difference between their model and ours is that it only performs item label distribution clustering; there are no annotator clusters.

DS + CNN uses the label aggregation methods introduced in DS (Dawid and Skene, 1979) and this is ultimately paired with a CNN classifier.

CL (Rodrigues and Pereira, 2018) is a neural joint modeling approach for modeling annotators and data features. Crowd layer (CL) attaches to the output of any network with a Q-dimensional output, i.e., a *crowd-layer*, which has multiple, parallel, Q-dimensional, new output layers, one for each annotator, and takes as input the old output layer. This extended model trains as a single, monolithic neural network. It then learns to predict the labels of each annotator simultaneously.

PGM is our proposed Bayesian probabilistic model, with the graph model introduced here for guidance. We set all of the Dirichlet parameters, i.e., α, γ, and τ, to 2. We consider two different learning algorithms: simulated annealing (with temperature schedule $T(t) = 1/(t+1)$) and expectation maximization (EM) with belief propagation.

For each of the the graphical models we performed (meta-)parameter search on the number of item and annotator clusters $K, L \in \{3, \ldots, 20\}$ and report the results of the best performing model (validated on development data). We evaluate these models using two different metrics. To evaluate the label distribution prediction, we report, over the test set, the mean KL divergence between each gold standard label distribution and the predicted label distribution $\mathrm{KL}(h_{\mathrm{dist}}(x) \| y)$. To evaluate single label prediction, we report the accuracy measured over the test set.

4.2. Results and Discussion

Table 2 shows the main results. We note that, with respect to KL divergence, our PGM models perform second-best, yielding better divergence than even the powerful CL model (MM+CNN only outperforming our BP/EM model by a bit). In terms of accuracy, our PGMs, while outperforming the CNN lower-bound baseline, do not unfortunately, according to this set of experiments, outperform the other baseline approaches. We suspect that our lower performance in terms of accuracy might be related to some degree of overfitting that we have, thus far, not been to control for.
Note that, in the case of all models (baselines and our proposed PGM variants), the final supervised learning classification phase was repeated 100 times (trained and evaluated) to calculate the reported error bars.

Limitations. Although we directly compared our models performance to those of (Weerasooriya et al., 2020), which represented clustering in item label space only, we did not perform head-to-head comparisons to the model of (Venanzi et al., 2014), which represents clustering in annotator label space only. This is due, in part, to the fact that the data from their studies is no longer being available. Nonetheless, we intend to run their models on the data that we do have in our next follow-up study.

5. Conclusion

In this work, we introduced a new graphical model for improving the quality of annotator labels, both from the perspective of the conventional problem of predicting the most common label as well as the emerging problem of predicting the distribution of labels that have been acquired/provided. Our methods combine label distribution learning with clustering jointly in the item and annotator label distribution spaces.

6. Bibliographical References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Aroyo, L. and Welty, C. (2014). The Three Sides of CrowdTruth. In *Journal of Human Computation*, volume 1, pages 31–34.

Callison-Burch, C. (2009). Fast, cheap, and creative: Evaluating translation quality using amazon's mechanical turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 286–295, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. 28(1):20–28.

Denkowski, M. and Lavie, A. (2010). Exploring normalization techniques for human judgments of machine translation adequacy collected using amazon mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 57–61, Stroudsburg, PA, USA. Association for Computational Linguistics.

Geng, X. (2016). Label Distribution Learning. In *IEEE Transactions on Knowledge and Data Engineering*, volume 28, pages 1734–1748.

Gordon, M. L., Lam, M. S., Park, J. S., Patel, K., Hancock, J. T., Hashimoto, T., and Bernstein, M. S. (2022). Jury Learning: Integrating Dissenting Voices into Machine Learning Models. *arXiv:2202.02950 [cs]*, February.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

Liu, T., Homan, C., Ovesdotter Alm, C., Lytle, M., Marie White, A., and Kautz, H. (2016). Understanding discourse on work and job-related well-being in public social media. In *Proceedings of the 54th Annual Meeting of the ACL*.

Liu, T., Venkatachalam, A., Bongale, P. S., and Homan, C. M. (2019a). Learning to Predict Population-Level Label Distributions. In *Seventh AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 68–76. A preliminary version appears in (Liu et al., 2019b).

Liu, T., Venkatachalam, A., Sanjay Bongale, P., and Homan, C. (2019b). Learning to predict population-level label distributions. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, pages 1111–1120. ACM.

Malinowski, B. (1967). The problem of meaning in primitive languages. *Meaning in Meaning*.

Metz, C. (2019). A.i. is learning from humans. many humans. *New York Times*, August 16. `https://www.nytimes.com/2019/08/16/technology/ai-humans.html`, retrieved 5/21/2021.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

Research, K. (2020). *Global Data Collection and Labeling Market By Data type By End User By Region, Industry Analysis and Forecast, 2020 - 2026*.

Rodrigues, F. and Pereira, F. C. (2018). Deep learning from crowds. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 1611–1618.

Venanzi, M., Guiver, J., Kazai, G., Kohli, P., and Shokouhi, M. (2014). Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 155–164.

Weerasooriya, T. C., Liu, T., and Homan, C. M. (2020). Neighborhood-based Pooling for Population-level Label Distribution Learning. In *Twenty Fourth European Conference on Artificial Intelligence*.