LEARNING FROM IMPERFECTANNOTATIONS: 
AN END-TO-END APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

Many machine learning systems today are trained on large amounts of human-annotated data. Annotation tasks that require a high level of competency make data acquisition expensive, while the resulting labels are often subjective, inconsistent, and may contain a variety of human biases. To improve data quality, practitioners often need to collect multiple annotations per example and aggregate them before training models. Such a multi-stage approach results in redundant annotations and may often produce imperfect “ground truth” labels that limit the potential of training supervised machine learning models. We propose a new end-to-end framework that enables us to: (i) merge the aggregation step with model training, thus allowing deep learning systems to learn to predict ground truth estimates directly from the available data, and (ii) model difficulties of examples and learn representations of the annotators that allow us to estimate and take into account their competencies. Our approach is general and has many applications, including training more accurate models on crowdsourced data, ensemble learning, as well as classifier accuracy estimation from unlabeled data. We conduct an extensive experimental evaluation of our method on 5 crowdsourcing datasets of varied difficulty and show accuracy gains of up to 25% over the current state-of-the-art approaches for aggregating annotations, as well as significant reductions in the required annotation redundancy.

1 INTRODUCTION

The rising popularity and recent success of deep learning has resulted in machine learning systems that rely on large amounts of annotated training data (LeCun et al., 2015; Wu et al., 2016; Gulshan et al., 2016; Esteva et al., 2017). The most common, scalable way to collect such large amounts of training data is through crowdsourcing (Howe, 2006). Crowdsourcing works well in simple settings where annotation tasks do not require domain expertise—for example, in object detection and recognition tasks in natural images and videos (e.g., Deng et al., 2009; Kovashka et al., 2016). However, annotation in specialized domains such as medical pathology requires a certain level of competency and expertise from the annotators which makes annotation expensive. Moreover, often times there is high rate of disagreement even between experts, which results in increasingly subjective and inconsistent labels (Elmore et al., 2015; Hutson et al., 2019).

A typical approach to dealing with subjectivity is to treat each annotation as simply noisy, collect multiple redundant labels per example (e.g., from different annotators), and then aggregate them using majority voting or other more advanced techniques (e.g., Dawid & Skene, 1979; Liu et al., 2012; Zhou et al., 2015; Zhou & He, 2016) to obtain a single “ground truth” label. At the expense of redundancy, this results in better data quality and more accurate estimates of the ground truth. More recently, the emerging systems for data programming and weak supervision also internally rely on label aggregation techniques similar to methods used for solving the crowdsourcing problem. Snorkel (Ratner et al., 2017; Bach et al., 2019) is a popular such system and was designed for efficient and low-cost creation of large-scale labeled datasets using programmatically generated, so-called weak labels. However, as we show in our empirical evaluation none of these systems solve label aggregation effectively in the presence of high subjectivity. We argue that to become more effective, these methods need to make use of meta-data and other types of information that may be available about the data instances and the annotators labeling them.
To this end, we propose a novel approach that allows us to train accurate predictive models of the ground truth directly on the non-aggregated imperfectly labeled data. Our method merges the two steps of: (i) aggregating subjective, weak, or noisy annotations, and (ii) training machine learning models. At training time, along with learning a model that predicts the ground truth, we also learn models of the difficulty of each example and the competence of each annotator in a generalizable manner (i.e., these models can make predictions for previously unseen examples and annotators). Our approach can be effectively used for training on crowdsourced data as well as on weakly labeled data, and also be used within frameworks such as Snorkel (Ratner et al., 2017; Bach et al., 2019) and significantly improve their performance. We propose a method that can:

1. **Learn truth estimators:** Learn functions representing the underlying ground truth, while imposing almost no constraints (as opposed to prior work). In fact, we are able to leverage the capacity of deep neural networks along with the interpretability provided by Bayesian models, in order to obtain highly expressive estimators of the underlying truth.

2. **Learn quality estimators:** Learn functions that estimate the quality of each annotator. When annotators can be described by some features (e.g., age, gender, location, etc. of an Amazon Mechanical Turk annotator, instead of just an ID), our quality estimators are able to generalize to new, previously unseen, predictors. Previous work only considered estimating accuracies of a fixed set of predictors, without being able to leverage any information we might have about them. Furthermore, in contrast to previous work, we are also able to predict the per-instance predictor competencies (i.e., our method can determine whether a human annotator is an expert for a subset of queries, instead of just estimating his/her overall accuracy), which is done by learning dependencies between the instances and the annotators. Finally, our approach is able to distinguish between multiple different types of errors by estimating the full confusion matrix for each instance-predictor pair.

3. **Be easily extended:** The truth and quality estimators can take arbitrary functional forms and fully leverage the expressivity of deep neural networks.

Both human annotators and machine learning classifiers may sometimes be unable to make predictions about certain aspects of the ground truth (e.g., human annotators may be unsure about what the correct answer to a question). The proposed method is formulated in a way that allows it to be extended such that it can also learn decision function estimators for the annotators (i.e., estimators that predict whether an annotator will be able to provide a prediction for a given data instance). These estimators can have significant implications for data annotation systems where the cost of querying annotators is high (e.g., when these annotators are highly qualified, such as doctors or other kinds of domain experts). This is because it allows for better matching annotators to instances, thus reducing the required amount of annotation redundancy. An overview of the proposed approach and model is shown in Figure 1, and a detailed description is provided in Section 3.

## 2 Related Work

Research on label aggregation and crowdsourcing dates back to the early 1970s, when Dawid & Skene (1979) proposed a probabilistic model to estimate ground truth labels using the expectation maximization (EM) algorithm. Since then, a variety of generalizations and improvements upon the original method have been proposed (Whitehill et al., 2009; Welinder et al., 2010; Liu et al., 2012; Zhou et al., 2013; Zhou & He, 2016). One of the central parts of the label aggregation algorithms is estimation of the accuracy of the annotators (or predictors) without having access to the ground truth. This problem has been of independent interest to the machine learning community, was termed as estimating accuracy from unlabeled data and studied by Collins & Singer (1999), Dasgupta et al. (2001), Bengio & Chapados (2003), Madani et al. (2004), Schuurmans et al. (2006), Balcan et al. (2013), and Parisi et al. (2014), among others. Almost none of the previous approaches explicitly considered modeling the ground truth, but rather assumed either some form of independence or knowledge of the true label distribution.

Collins & Huynh (2014) reviewed many methods that were proposed for estimating the accuracy of medical tests in the absence of a gold standard. Platanios et al. (2014) proposed formulating the problem as an optimization problem that uses agreement rates between multiple noisy annotators. Platanios et al. (2017) improved upon agreement-based accuracy estimation using logical constraints between the noisy labels. Tian & Zhu (2015) proposed a max-margin majority voting scheme applied
to crowdsourcing. More recently, Khetan et al. (2017) proposed to use a parametric function to model the ground truth and showed that the approach can sometimes be functional even in the limit of a single noisy label per example. Among recent approaches, Zhou et al. (2015) formulated the problem as a form of regularized minimax conditional entropy and established one of the most competitive baselines on many public crowdsourcing datasets.

Our proposed method generalizes the approaches of Zhou et al. (2015), Platanios et al. (2016), and Khetan et al. (2017). Similar to Platanios et al. (2016), we define a generative process for our observations. However, our model is able to handle categorical labels, as opposed to just binary. Similar to Zhou et al. (2015), we define the confusion matrix for each instance-predictor pair as a function of instance difficulty and predictor competence. However, we explicitly learn the difficulty and competence as functions, which allows us to generalize to previously unseen instances and annotators. Interestingly, the inference algorithm for our generative probabilistic model has a similar form to that of Zhou et al. (2015) (except for the explicit learning of the ground truth, difficulty, and competence functions). Thus, we also show that the algorithm of Zhou et al. (2015) can be re-derived as an EM inference algorithm for a generative model, simplifying the argument of the original paper.

Finally, similar to Khetan et al. (2017), we use a parametric function to model the ground truth, and also go a step further and propose to use parametric functions to model the instance difficulties and predictor competences. Thus, our approach enables estimation of which annotators are likely to perform better on which instances, potentially enabling to allocate annotators more optimally and reduce costs.

3 Proposed Method

We denote the observed data by $D = \{x_i, \hat{y}_i\}_{i=1}^N$, where $\hat{y}_i = \{\hat{y}_{ij}\}_{j \in M_i}$, $M_i$ is the set of predictors that made predictions for instance $x_i$, and $\hat{y}_{ij}$ is the output of predictor $f_j$ for instance $x_i$. Our goal is to learn functions representing the underlying ground truth and predictor qualities, given our observations $D$.

Ground Truth. We define the ground truth as a function $h_\theta(x_i)$ that is parameterized by $\theta$ and that approximates the true distribution of the label given $x_i$. In our setting, $h_\theta(x_i) \in \mathbb{R}^C_0$ and $\sum_j [h_\theta(x_i)]_j = 1$, where $C$ is number of values the label can take (i.e., assuming categorical labels).

More specifically, $[h_\theta(x_i)]_k \triangleq P(y_i = k \mid x_i)$, where we use square brackets and subscripts to denote indexing of vectors, matrices, and tensors. For example, $h_\theta$ could be a deep neural network that would normally be trained in isolation using the cross-entropy loss function. In our method the network is trained using the Expectation-Maximization algorithm, as described in the next section.
Predictor Qualities. We define the predictor qualities as the confusion matrices \( Q_{ij} \in \mathbb{R}^{C \times C} \), for each instance \( x_i \) and predictor \( f_j \), where \( \sum_j (Q_{ij})_{kl} = 1 \), for all \( k \in \{1, \ldots, C\} \). \( (Q_{ij})_{kl} \) represents the probability that predictor \( f_j \) outputs label \( l \) given that the true label of instance \( x_i \) is \( k \). We define these confusion matrix in a way that generalizes the successful approach of Zhou et al. (2015)\

\[
Q_{ij} = D_i \bullet_o C_j,
\]

(1)

where \( \bullet_o \) represents an inner product along the \( i \)th dimension of the two tensors, and:

- \( D_i = d_o(x_i) \) represents the difficulty tensor for instance \( x_i \), where \( d \) is a function parameterized by \( \phi \), \( D_i \in \mathbb{R}^{C \times C \times L} \), and \( L \) is a latent dimension (it is a hyperparameter of our model). \( [D_i]_{klt} \) is an \( L \)-dimensional embedding representing the likelihood of confusing \( x_i \) as having label \( l \) instead of \( k \), when \( k \) is its true label.
- \( C_j = c_o(r_j) \) represents the competence tensor for predictor \( f_j \), where \( c \) is a function parameterized by \( \psi \), \( r_j \) is some representation of \( f_j \) (e.g., could be a one-hot encoding of the predictor, in the simplest case), and \( C_j \in \mathbb{R}^{C \times C \times L} \). \( [C_j]_{klt} \) is an \( L \)-dimensional embedding representing the likelihood that predictor \( f_j \) confuses label \( k \) for \( l \), when \( k \) is the true label.

Using \( L > 1 \) allows the instance difficulties and predictor competences to encode more information. An intuitive way to think about this is that we are embedding difficulties and competencies in a common latent space, which can be thought of as jointly clustering them. This is in fact very similar to how matrix factorization methods are used for collaborative filtering in recommender systems.

Our goal is to learn functions \( h_\theta \), \( d_o \), and \( c_o \), given observations \( D \). In order to do that, we propose the following generative process for our observations. For \( i = 1, \ldots, N \), we first sample the true label for \( x_i \), \( y_i \sim \text{Categorical}(h_\theta(x_i)) \). Then, for \( j \in M_i \), we sample the predictor output \( \hat{y}_{ij} \sim \text{Categorical}(Q_{ij}|y_i) \), where \( [Q_{ij}]_{y_i} \) represents the \( y_i \)th row of \( Q_{ij} \). In the next section, we propose an algorithm learning the parameters \( \theta \), \( \phi \), and \( \psi \).

3.1 Learning

A widespread approach for performing learning with probabilistic generative models, is to maximize the likelihood of the observed data with respect to the model parameters. Let \( y = \{y_i\}_{i=1}^N \). The likelihood of a single observation, \( \hat{y}_{ij} \), can be derived as follows:

\[
p(D, y) = \prod_{i=1}^N p(y_i) \prod_{j \in M_i} p(\hat{y}_{ij} \mid y_i),
\]

(2)

where \( p(\hat{y}_{ij} \mid y_i) \) depends on \( Q_{ij} \). There are two main approaches in which we can maximize the likelihood function of Equation (2): (i) marginalize out the \( y_i \) latent variables and then maximize with respect to \( \theta \), \( \phi \), and \( \psi \), or (ii) use the expectation maximization (EM) algorithm originally proposed by Dempster et al. (1977). It has previously been observed that the EM algorithm can perform much better than approach (i) for mixture models (Bishop, 2006). This is because the latter tends to get stuck in bad local optima. Since our model resembles a Bernoulli mixture model with the latent assignments being defined by the \( y_i \)'s, we decided to use the EM algorithm. We derive the EM algorithm steps for our model, as follows:

E-Step. We need to compute the expectation of \( y_i \) given \( D \) and \( y \setminus i \) (which denotes all of \( y \) except for \( y_i \)), for all \( i = 1, \ldots, N \), and we know that:

\[
p(y_i \mid D, y \setminus i) \propto p(D, y) = \prod_{s=1}^N p(y_s) \prod_{j \in M_s} p(\hat{y}_{sj} \mid y_s).
\]

(3)

Therefore, by removing all terms that do not depend on \( y_i \) and normalizing, we obtain the following expectation (which we compute during this step, while keeping \( \theta \), \( \phi \), and \( \psi \) fixed):

\[
\mathbb{E}_{y|D} \{ \mathbb{I}_{[y_i = k]} \} = \frac{\lambda^k_i}{\sum_{l=1}^C \lambda^l_i}, \quad \text{where} \quad \lambda^k_i = [h_\theta(x_i)]_k \prod_{j \in M_i} \frac{[Q_{ij}]_{y_i}}{\sum_{k=1}^C [Q_{ij}]_{y_i}}.
\]

Footnote 1: We also perform a normalization step such that all elements of \( Q_{ij} \) are non-negative and such that each row sums to 1 (thus making each row a valid probability distribution).

Footnote 2: In fact, we also experimented with the marginalization approach and it consistently underperformed EM.

Footnote 3: Note that \( Q_{ij} \) depends on \( \phi \) and \( \psi \).
For brevity, in what follows, we use the following notation \( \hat{y}_i^k = \mathbb{E}_{y_i | D} \{ 1_{[y_i = k]} \} \).

**M-Step.** We maximize the following log-likelihood function with respect to \( \theta, \phi, \text{and} \psi \), while using the values of \( \hat{y}_i^k \) computed in the last E-step:

\[
\log \mathcal{L} = \sum_{i=1}^{N} \sum_{k=1}^{C} \hat{y}_i^k \log[p(y_i | x_i)]_k + \sum_{i=1}^{N} \sum_{j \in M_i} \sum_{k=1}^{C} \frac{[Q_{ij}]_{k|\hat{y}_j}}{\sum_{l=1}^{C}[Q_{ij}]_{l|\hat{y}_j}}.
\]

The training procedure for learning the parameters \( \theta, \phi, \text{and} \psi \) consists of iterating over the E-step and the M-step shown above, until convergence, where convergence can be measured by computing the change in parameter values across learning iterations. It is important to note that EM finds local optima of the likelihood function, and so the starting point can play an important role. Also, as Platanios et al. (2016) mention, there exists an inherent symmetry in our model that can be problematic. The likelihood of any observed data is the same if we flip the true underlying labels and the predictor qualities (i.e., set \( y_i \text{di\text{pped}} = 1 - y_i \) and \( Q_{ij} \text{di\text{pped}} = 1 - Q_{ij} \)). We would like to somehow encode the prior assumption that most of the predictors are correct most of the time. One way to do this is by choosing the starting point of the EM algorithm carefully.

**Initialization.** In the E-step shown in Equation 3 we compute the expected values of the true underlying labels, \( y_i \). We can encode the assumption mentioned in the previous paragraph by replacing the first E-step with a majority vote among the predictors:

\[
\mathbb{E}_{y_i | D} \{ 1_{[y_i = k]} \} = \frac{\sum_{j \in M_i} 1_{[\hat{y}_j = k]}}{|M_i|},
\]

where \( |M_i| \) denotes the size of set \( M_i \). We initialize the EM algorithm by replacing the first E-step with this majority vote approximation. As we show in our experiments, this helps us avoid the aforementioned symmetry, and thus we refer to this initialization scheme as symmetry-breaking initialization. Note also that in the case where the predictors provide us with \( \mathbb{P}(\hat{y}_j = k) \), instead of a single categorical value, we can still use this initialization scheme by replacing \( 1_{[\hat{y}_j = k]} \) with \( \mathbb{P}(y_j = k) \), in Equation 6.

**Marginal Likelihood Fine-Tuning.** In our experiments we found that maximizing the marginal likelihood function after EM converges tends to improve performance. We refer to this step as marginal likelihood fine-tuning. More specifically, after the values of the parameters \( \theta, \phi, \text{and} \psi \), converge to fixed values across multiple EM steps, we solve the following maximization problem using these fixed values as the initial point:

\[
\max_{\theta,\phi,\psi} \sum_{y} p(D, y) \iff \max_{\theta,\phi,\psi} \sum_{i=1}^{N} \sum_{j \in M_i} \log \sum_{k=1}^{C} h_\theta(x_i)|_k \frac{[Q_{ij}]_{k|\hat{y}_j}}{\sum_{l=1}^{C}[Q_{ij}]_{l|\hat{y}_j}}.
\]

### 3.2 Instance and Predictor Representations

A major benefit of the proposed approach when compared to prior work is that we explicitly learn models that represent the ground truth and the predictor qualities. These models receive as input representations of the data instances (e.g., \( x_i \)) and of the predictors (e.g., \( r_j \)). It is thus important to define these representations. For many problems, the representations of the data instances can be defined in the same manner as was previously done when performing supervised learning (e.g., we can directly use raw pixel values representing images). However, predictor representations were first introduced in this work. A simple approach would be to use a one-hot encoding of the predictors. However, this would not allow for any amount of information sharing across predictor (e.g., what if two predictors are very similar)? We know from prior work that modeling dependencies between the predictors can be very important (e.g., Platanios et al., 2016). One way to allow for that is to learn vector embedding representations for the predictors, which would be implicitly equivalent to clustering them. In fact, that is the approach we used in our experiments. Ideally, we would like to use any information we have about these predictors (e.g., if they are AMT annotators we could use their age, location, etc.) but unfortunately we could not find any public datasets that provide information about the predictors/annotators.
3.3 DISCUSSION

The approach we have proposed in this section can be thought of as introducing a new loss function for training the model \( h_\theta \) using multiple imperfect labels per training instance, each coming from a different source. This new loss function introduces latent variables representing the ground truth labels, as well as a couple of auxiliary models that are learned, and which represent the instance difficulties and predictor competences. We also proposed an EM-based algorithm to minimize this new loss function as well as an initialization scheme. Perhaps most interestingly, a key difference between this approach and previous work is that we are able to explicitly learn functions that output the likelihood that a predictor will label a specific instance correctly. This enables using this approach to perform crowdsourcing more actively by assigning annotators to instances they are more likely to label correctly, thus helping reduce redundancy and drive costs down.

3.4 EXTENDING TO MULTI-LABEL SETTINGS

Our method can easily be extended to handle settings where we have multiple categorical labels that can be assigned to each instance. In that case, the model per label is defined in the same way as previously, except that now the functions \( h_\theta \), \( d_\phi \), and \( c_\psi \) also take as input a representation for the label (e.g., a label embedding). This allows us to share information across labels and can be thought of as a generalization of the approach by Platanios et al. (2016), where information is shared by clustering the labels. Furthermore, it allows us to use the proposed method in extreme classification settings (e.g., Prabhu & Varma, 2014) or settings where the number of labels is not fixed and known a priori and can keep increasing (e.g., face recognition) (Weinberger & Saul, 2009; Liu et al., 2016). This is made possible by learning label representations and then letting the difficulty and competence functions also take as input a pair of labels and return a vector instead of a three-dimensional tensor. In the next section, we show how learning label representations can significantly enhance the robustness and performance of our approach.

4 EXPERIMENTS

Datasets. We evaluate our proposed approach on multiple datasets from the crowdsourcing domain, all of which had ground truth labels, (multiple) subjective annotations for each example, as well as information on who provided each annotation (i.e., the annotator ID):

1. Blue Birds (Welinder et al., 2010): Bird photos labeled as Indigo Bunting or Blue Grosbeak.
2. Word Similarity (Snow et al., 2008): Pairs of words labeled as similar or dissimilar.
3. RTE (Snow et al., 2008): Pairs of sentences labeled as whether the first entails the second.
4. Medical Causes (Dumitrache et al., 2018): Sentences that contain 2 medical terms labeled if one of the terms causes the other (e.g., pancreatic adenocarcinoma causes weight loss).
5. Medical Treats (Dumitrache et al., 2018): The same sentences with medical terms labeled if one of the terms treats the other (e.g., aspirin treats pain).

The last two datasets are in fact part of a single dataset on medical relations, and thus we are able to perform experiments both with the single task formulation of our algorithm and the multi-task formulation. As we discuss at the end of this section, this allows us to show how our approach can be used to share information across labels and improve the quality of the learned models.

Statistics for these datasets are provided in Table 1. Note that most of these datasets were provided without the associated features of the annotator identifiers. We have collected features and annotator identification information for each dataset and we are making them all publicly available in a standardized format. More details are provided in our code and data repository which is available at [http://anonymous](http://anonymous).

Variants of Our Method. We evaluate the following two variants of our approach:

- LIA: A version, where we use instance and predictor features specific to each dataset. When features are not available for the instances and/or the predictors, we learn embeddings of size 4 which are initialized randomly and optimized along with the other model parameters during the M-step (see Section 3.1).
Table 1: Statistics for the datasets we used in our experiments. “#Predictors” refers to the total number of predictors in the dataset, “Average Redundancy” refers to the average number of predictions provided for each instance, “Average Accuracy” refers to the average predictor accuracy, and “Random Accuracy” refers to the accuracy obtained of a completely random predictor.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Instances</th>
<th>#Predictors</th>
<th>Average Redundancy</th>
<th>Average Accuracy (%)</th>
<th>Random Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Birds (Welinder et al., 2010)</td>
<td>108</td>
<td>39</td>
<td>39</td>
<td>63.56</td>
<td>50.00</td>
</tr>
<tr>
<td>Word Similarity (Snow et al., 2008)</td>
<td>30</td>
<td>10</td>
<td>10</td>
<td>81.33</td>
<td>50.00</td>
</tr>
<tr>
<td>RTE (Snow et al., 2008)</td>
<td>800</td>
<td>164</td>
<td>10</td>
<td>84.13</td>
<td>50.00</td>
</tr>
<tr>
<td>Medical Causes (Dumitrache et al., 2018)</td>
<td>3,984</td>
<td>408</td>
<td>15</td>
<td>32.40</td>
<td>7.00</td>
</tr>
<tr>
<td>Medical Treats (Dumitrache et al., 2018)</td>
<td>3,984</td>
<td>408</td>
<td>15</td>
<td>38.88</td>
<td>7.00</td>
</tr>
</tbody>
</table>

- **LIA-ML**: Multi-label variant of the above method. This method is only used with the medical relations datasets. In this case, we consider all 14 medical relations included in the dataset jointly and only evaluate on the two for which the ground truth is provided (i.e., “causes” and “treats”). We use this method variant in order to show how our approach can effectively share information across labels.

**Baselines.** We compare our methods against the following baselines for ground truth estimation:

- **MAJ**: Simple majority voting. Note that we use soft majority voting whenever possible, i.e., we use soft labels (probabilities or confidence scores) whenever the predictors provide them, instead of always thresholding them to obtain discrete labels.

- **MMCE**: Regularized minimax conditional entropy by Zhou et al. (2015). This method has been shown to outperform alternatives and thus we consider it the current state-of-the-art for the crowdsourcing problem setting.

- **MAJ-ME**: First, we estimate ground truth using MAJ. Next, we learn embeddings of size 16 for the instances and the predictors, as well as the same model $h_\theta$ as LIA-E but use MAJ-estimated ground truth, instead of modeling it with latent variables.

- **MAJ-M**: The same method as MAJ-ME, but instead of learning embeddings, we use features for the instances and the predictors, whenever available.

- **Snorkel**: A method originally designed for aggregating annotations of programmatic weak predictors proposed by Ratner et al. (2017), that is part of a popular software package that allows for subsequent training of machine learning models on the aggregated data.

- **MeTaL**: Successor to Snorkel, proposed by Ratner et al. (2018).

For the **Snorkel** and **MeTaL** baselines, we use the original implementation provided by the authors.

In all our models, $h_\theta$ and $d_\phi$ are multi-layer perceptrons (MLPs) with 4 layers of 16 hidden units each, with the only exception being the medical relations dataset where we used 32 units for each layer. $c_\psi$ is always modeled as a linear function. Note that for both the embedding sizes and the MLP sizes, we did not perform an extensive search to choose these values; we rather performed a small grid search and selected the number that resulted in the highest validation data likelihood.

**Setup.** During each M-step we use the AMSGrad optimizer (Reddi et al., 2018) to maximize the log-likelihood function with the learning rate set to 0.001, and we perform 1,000 optimization iterations using a batch size of 1,024. Overall, we perform 10 EM iterations (all models did converge within that limit) with warm starting (i.e., the model parameters are always initialized to the values obtained during the previous M-step). When using LIA with image instances we use as image features the activations of the last layer of a pre-trained VGG-16 Convolutional Neural Network (CNN). Similarly, for all text instances we use as text features the representations provided by a pre-trained BERT model (Devlin et al., 2018). More details on our setup and the model hyperparameters can be found in our code repository at [http://anonymous](http://anonymous).

**Metric.** We evaluate all methods by computing the *accuracy* of the predicted instance labels. This is a common metric for evaluating crowdsourcing methods and it also implicitly measures the quality of the confusion matrices predicted by our model. This is because these confusion matrices heavily

4 [https://github.com/HazyResearch/snorkel](https://github.com/HazyResearch/snorkel)
Table 2: Accuracy across varying levels of redundancy, for all datasets we used in our experiments. For each experiment, we report mean accuracy and standard error over 50 runs from random initializations. The best statistically significant results based on standard error are shown in red color.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Redundancy</th>
<th>Accuracy (%)</th>
<th></th>
<th></th>
<th>Snorkel</th>
<th>MeTaL</th>
<th>LIA</th>
<th>LIA-ML</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MAJ</td>
<td>MMCE</td>
<td>MAJ-ME</td>
<td>MAJ-M</td>
<td>Snorkel</td>
<td>MeTaL</td>
<td>LIA</td>
</tr>
<tr>
<td>Blue Birds</td>
<td>2</td>
<td>63.9±0.5</td>
<td>66.9±0.7</td>
<td>64.4±0.5</td>
<td>65.1±0.7</td>
<td>63.8±0.7</td>
<td>63.0±0.7</td>
<td>63.1±0.6</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>71.1±0.4</td>
<td>73.9±0.5</td>
<td>70.9±0.5</td>
<td>70.9±0.5</td>
<td>72.4±0.6</td>
<td>71.5±0.6</td>
<td>72.8±0.5</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>74.4±0.4</td>
<td>76.5±0.5</td>
<td>75.9±0.5</td>
<td>75.4±0.5</td>
<td>76.0±0.4</td>
<td>83.0±0.3</td>
<td>81.5±0.5</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>76.4±0.3</td>
<td>78.5±0.4</td>
<td>76.2±0.3</td>
<td>76.1±0.2</td>
<td>76.0±0.3</td>
<td>87.0±0.1</td>
<td>88.0±0.3</td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>75.9±0.0</td>
<td>79.6±0.1</td>
<td>78.4±0.0</td>
<td>78.4±0.0</td>
<td>76.0±0.0</td>
<td>89.0±0.0</td>
<td>90.2±1.1</td>
</tr>
<tr>
<td>Word Similarity</td>
<td>2</td>
<td>82.8±0.8</td>
<td>80.1±0.9</td>
<td>87.2±0.5</td>
<td>87.7±0.7</td>
<td>76.2±0.7</td>
<td>76.0±1.0</td>
<td>88.7±0.7</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>87.1±0.6</td>
<td>87.0±0.6</td>
<td>91.4±0.4</td>
<td>91.3±0.4</td>
<td>76.3±0.6</td>
<td>85.1±0.7</td>
<td>92.7±0.4</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>88.6±0.2</td>
<td>90.2±0.3</td>
<td>93.3±0.0</td>
<td>93.3±0.0</td>
<td>76.3±0.1</td>
<td>93.1±0.0</td>
<td>96.3±0.1</td>
</tr>
<tr>
<td>RTE</td>
<td>2</td>
<td>72.8±0.2</td>
<td>75.3±0.3</td>
<td>72.8±0.4</td>
<td>74.5±0.3</td>
<td>65.2±0.3</td>
<td>61.0±0.3</td>
<td>78.0±0.3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>84.8±0.1</td>
<td>89.7±0.1</td>
<td>84.0±0.2</td>
<td>84.8±0.1</td>
<td>79.1±0.1</td>
<td>72.4±0.3</td>
<td>89.1±0.1</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>90.0±0.1</td>
<td>93.1±0.1</td>
<td>90.4±0.1</td>
<td>89.9±0.1</td>
<td>90.0±0.0</td>
<td>78.0±0.0</td>
<td>93.1±0.1</td>
</tr>
<tr>
<td>Medical Causes</td>
<td>2</td>
<td>26.8±0.1</td>
<td>29.1±0.2</td>
<td>24.2±0.2</td>
<td>26.5±0.1</td>
<td>27.1±0.1</td>
<td>25.3±0.3</td>
<td>29.5±0.1</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>24.1±0.1</td>
<td>24.5±0.1</td>
<td>23.6±0.1</td>
<td>24.2±0.1</td>
<td>24.0±0.1</td>
<td>21.0±0.1</td>
<td>30.9±0.3</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>24.1±0.1</td>
<td>24.4±0.1</td>
<td>23.6±0.1</td>
<td>24.1±0.1</td>
<td>24.0±0.0</td>
<td>20.0±0.0</td>
<td>30.5±0.2</td>
</tr>
<tr>
<td>Medical Treats</td>
<td>2</td>
<td>33.8±0.4</td>
<td>35.3±0.3</td>
<td>35.7±0.4</td>
<td>34.2±0.2</td>
<td>33.3±0.1</td>
<td>22.1±0.3</td>
<td>38.6±0.4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>34.2±0.3</td>
<td>36.8±0.2</td>
<td>34.1±0.3</td>
<td>33.6±0.1</td>
<td>34.0±0.3</td>
<td>31.0±0.3</td>
<td>38.3±0.2</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>34.2±0.2</td>
<td>38.5±0.1</td>
<td>34.3±0.3</td>
<td>35.2±0.2</td>
<td>35.0±0.1</td>
<td>33.1±0.1</td>
<td>42.1±0.2</td>
</tr>
</tbody>
</table>

Influence the supervision provided to the ground truth model, \( h_g \), while training. Furthermore, instead of just computing accuracy for the full datasets, we also measure how performance varies as a function of redundancy—the maximum number of annotations provided per instance. In order to limit the redundancy for existing datasets we randomly sample subsets of the provided annotations. Performing well in low redundancy settings is very important because it can result in significantly reduced crowdsourcing costs.

Results. Our results are presented in Table 2. LIA methods consistently outperform or match alternative approaches. In certain cases (e.g., in “Blue Birds”) we are able to boost accuracy over the best alternative method by more than 10%, thus establishing a new state-of-the-art for this dataset. In the multi-task setting, where we train LIA–ML model to jointly infer ground truth for both Medical Cause and Medical Treats while sharing representations of the instance difficulties and annotator competencies. We observe that multi-task training boosts performance of the model by more 8% absolute (or over 20% relative) of its single task counterpart, outperforming the baselines by over 25% relative. Finally, our approach can obtain the performance of the best alternative method using up to 4 times less redundancy, which can have significant implications for the cost of crowdsourcing, especially when annotation requires domain expertise (e.g., in healthcare).

We note that Snorkel and MeTaL tend to perform well overall, but sometimes fail entirely (often performing on par with or worse than majority voting)\(^3\). MeTaL also suffers from calibration issues, as it often achieves very low accuracy while having reasonable mean average precision. We believe that data programming systems could further benefit significantly by integrating our method in their pipeline, tying together the label aggregation and model training phases.

5 Conclusion

In this paper, we have introduced a learning framework for: (i) training deep models directly on data with imperfect annotations, and (ii) modeling the processes that produced the labels. Our approach improves upon the classical and widely used two-stage setup (first aggregate and denoise the labels, then train the model) and by merging the two stages. As a result, we are able to train models end-to-end using multiple noisy labels, while estimating the difficulties of the examples and learning accurate representations for the annotators that produced the labels. Experimental results on multiple small and large scale publicly available crowdsourcing datasets indicate that our method results in significant gains in accuracy (up to 25% relative gain over the current state-of-the-art approaches for aggregating noisy labels). Moreover, it turns out that training the model to predict multiple related labels simultaneously improves the learned representations and results in further gains in predictive performance of the model.

\(^3\)This behavior has also been observed by others (e.g., https://github.com/HazyResearch/snorkel/issues/1073).
REFERENCES


Miles Hutson, Olga Kanzheleva, Caitlin Taggart, Bilson Campana, and Quang Anh Duong. Quality control challenges in crowdsourcing medical labeling and diagnosis. 2019.


