Aggregating and Predicting Sequence Labels from Crowd Annotations

An T. Nguyen[1] **Byron C. Wallace**[2] **Junyi Jessy Li**[3] **Ani Nenkova**[3] **Matthew Lease** [1]

[1]University of Texas at Austin, [2]Northeastern University,
[3]University of Pennsylvania,
atn@cs.utexas.edu, byron@ccs.neu.edu,
{ljunyi|nenkova}@seas.upenn.edu, ml@utexas.edu

Abstract

Despite sequences being core to NLP, scant work has considered how to handle noisy sequence labels from multiple annotators for the same text. Given such annotations, we consider two complementary tasks: (1) aggregating sequential crowd labels to infer a best single set of consensus annotations; and (2) using crowd annotations as training data for a model that can predict sequences in unannotated text. For aggregation, we propose a novel Hidden Markov Model variant. To predict sequences in unannotated text, we propose a neural approach using Long Short Term Memory. We evaluate a suite of methods across two different applications and text genres: Named-Entity Recognition in news articles and Information Extraction from biomedical abstracts. Results show improvement over strong baselines. Our source code and data are available online[1].

1 Introduction

Many important problems in Natural Language Processing (NLP) may be viewed as sequence labeling tasks, such as part-of-speech (PoS) tagging, named-entity recognition (NER), and Information Extraction (IE). As with other machine learning tasks, automatic sequence labeling typically requires annotated corpora on which to train predictive models. While such annotation was traditionally performed by domain experts, crowdsourcing has become a popular means to acquire large labeled datasets at lower cost, though annotations from laypeople may be lower quality than those from domain experts (Snow et al., 2008). It is therefore essential to model crowdsourced label quality, both to estimate individual annotator reliability and to aggregate individual annotations to induce a single set of "reference standard" consensus labels. While many models have been proposed for aggregating crowd labels for binary or multiclass classification problems (Sheshadri and Lease, 2013), far less work has explored crowd-based annotation of sequences (Finin et al., 2010; Hovy et al., 2014; Rodrigues et al., 2014).

In this paper, we investigate two complementary challenges in using sequential crowd labels: how to best aggregate them (Task 1); and how to accurately predict sequences in unannotated text given training data from the crowd (Task 2). For aggregation, one might want to induce a single set of high-quality consensus annotations for various purposes: (i) for direct use at run-time (when a given application requires human-level accuracy in identifying sequences); (ii) for sharing with others; or (iii) for training a predictive model.

When human-level accuracy in tagging of sequences is not crucial, automatic labeling of unannotated text is typically preferable, as it is more efficient, scalable, and cost-effective. Given a training set of crowd labels, how can we best predict sequences in unannotated text? Should we: (i) consider Task 1 as a pre-processing step and train the model using consensus labels; or (ii) instead directly train the model on all of the individual annotations, as done by Yang et al. (2010)? We investigate both directions in this work.

Our approach is to augment existing sequence labeling models such as HMMs (Rabiner and Juang, 1986) and LSTMs (Hochreiter and Schmidhuber, 1997; Lample et al., 2016) by introducing an explicit "crowd component". For HMMs, we model this crowd component by including additional parameters for worker label quality and crowd label variables. For the LSTM, we introduce a vector representation for each annotator. In

[1] Soure code and biomedical abstract data: www.github.com/thanhan/seqcrowd-acl17, www.byronwallace.com/EBM_abstracts_data

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 299–309
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-1028

both cases, the crowd component models both the noise from labels and the label quality from each annotator. We find that principled combination of the "crowd component" with the "sequence component" yields strong improvement.

For evaluation, we consider two practical applications in two text genres: NER in news and IE from medical abstracts. Recognizing named-entities such as people, organizations or locations can be viewed as a sequence labeling task in which each label specifies whether each word is Inside, Outside or Beginning (IOB) a named-entity. For this task, we consider the English portion of the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003), using crowd labels collected by Rodrigues et al. (2014).

For the IE application, we use a set of biomedical abstracts that describe Randomized Controlled Trials (RCTs). The crowdsourced annotations comprise labeled text spans that describe the patient populations enrolled in the corresponding RCTs. For example, an abstract may contain the text: *we recruited and enrolled diabetic patients*. Identifying these sequences is useful for downstream systems that process biomedical literature, e.g., clinical search engines (Huang et al., 2006; Schardt et al., 2007; Wallace et al., 2016).

Contributions. We present a systematic investigation and evaluation of alternative methods for handling and utilizing crowd labels for sequential annotation tasks. We consider both how to best aggregate sequential crowd labels (Task 1) and how to best predict sequences in unannotated text given a training set of crowd annotations (Task 2). As part of this work, we propose novel models for working with noisy sequence labels from the crowd. Reported experiments both benchmark existing state-of-the-art approaches (sequential and non-sequential) and show that our proposed models achieve best-in-class performance. As noted in the Abstract, we have also shared our sourcecode and data online for use by the community.

2 Related Work

We briefly review two separate threads of relevant prior work: (1) sequence labeling models; and (2) aggregation of crowdsourcing annotations.

Sequence labeling. Early work on learning for sequential tasks used HMMs (Bikel et al., 1997). HMMs are a class of generative probabilistic models comprising two components: an emission model from a hidden state to an observation and a transition model from a hidden state to the next hidden state. Later work focused on discriminative models such as Maximum Entropy Models (Chieu and Ng, 2002) and Conditional Random Fields (CRFs) (Lafferty et al., 2001). These were able to achieve strong predictive performance by exploiting arbitrary features, but they may not be the best choice for label aggregation. Also, compared to the simple HMM model, discriminative sequentially structured models require more complex optimization and are generally more difficult to extend. Here we argue for the generative HMMs for our first task of aggregating crowd labels. The generative nature of HMMs is a good fit for existing crowd modeling techniques and also enables very efficient parameter estimation.

In addition to the supervised setting, previous work has studied unsupervised HMMs, e.g., for PoS induction (Goldwater and Griffiths, 2007; Johnson, 2007). These works are similar to our work in trying to infer the hidden states without labeled data. Our graphical model is different in incorporating signal from the crowd labels.

For Task 2 (training predictive models), we consider CRFs and LSTMs. CRFs are undirected, conditional models that can exploit arbitrary features. They have achieved strong performance on many sequence labeling tasks (McCallum and Li, 2003), but they depend on hand-crafted features. Recent work has considered end-to-end neural architectures that *learn* features, e.g., Convolutional Neural Networks (CNNs) (Collobert et al., 2011; Kim, 2014; Zhang and Wallace, 2015) and LSTMs (Lample et al., 2016). Here we modify the LSTM model proposed by Lample et al. (2016) by augmenting the network with '*crowd worker vectors*'.

Crowdsourcing. Acquiring labeled data is critical for training supervised models. Snow et al. (2008) proposed using Amazon Mechanical Turk to collect labels in NLP quickly and at low cost, albeit with some degradation in quality. Subsequent work has developed models for improving aggregate label quality (Raykar et al., 2010; Felt et al., 2015; Kajino et al., 2012; Bi et al., 2014; Liu et al., 2012; Hovy et al., 2013). Sheshadri and Lease (2013) survey and benchmark methods.

However, these models are almost all in the binary or multiclass classification setting; only a few have considered sequence labeling. Dredze et al. (2009) proposed a method for learning a CRF

model from multiple labels (although the identities of the annotators or workers were not used). Rodrigues et al. (2014) extended this approach to account for worker identities, providing a joint "crowd-CRF" model. They collected a dataset of crowdsourced labels for a portion of the CoNLL 2003 dataset. Using this, they showed that their model outperformed Dredze et al. (2009)'s model and other baselines. However, due to the technical difficulty of the joint approach with CRFs, they resorted to strong modeling assumptions. For example, their model assumes that for each word, only one worker provides the correct answer while all others label the word completely randomly. While this assumption captures some aspects of label quality, it is potentially problematic, such as for 'easy words' labeled correctly by all workers.

More recently, **?** proposed HMM models for aggregating crowdsourced discourse segmentation labels. However, they did not consider the general sequence labeling setting. Their method includes task-specific assumptions, e.g., that discourse segment lengths follow some empirical distribution estimated from data. In the absence of a gold standard, they evaluated by checking that workers accuracies are consistent and by comparing their two models to each other. We include their approach along with Rodrigues et al. (2014) as a baseline in our evaluation.

3 Methods

We present our Task 1 HMM approach in Section 3.1 and our Task 2 LSTM approach in Section 3.2.

3.1 HMMs with Crowd Workers

Model: We first define a standard HMM with hidden states h_i, observations v_i, transition parameter vectors $\boldsymbol{\tau}_{h_i}$ and emission parameter vectors $\boldsymbol{\Omega}_{h_i}$:

$$h_{i+1}|h_i \sim \text{Discrete}(\boldsymbol{\tau}_{h_i}) \quad (1)$$

$$v_i|h_i \sim \text{Discrete}(\boldsymbol{\Omega}_{h_i}) \quad (2)$$

The discrete distributions here are governed by Multinomials. In the context of our task, v_i is the word at position i and h_i is the true, latent class of v_i (e.g., entity or non-entity).

For the crowd component, assume there are n classes, and let l_{ij} be the label for word i provided by worker j. Further, let $\mathbf{C}^{(j)}$ be the confusion matrix for worker j, i.e., $\mathbf{C}^{(j)}_k$ is a vector of size n in which element k' is the probability of worker j providing the label k' for a word of true class k:

$$l_{ij}|h_i \sim \text{Discrete}(\mathbf{C}^{(j)}_{h_i}) \quad (3)$$

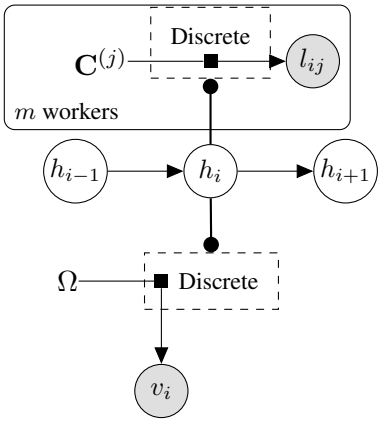

Figure 1: The factor graph for our HMM-Crowd model. Dotted rectangles are gates, where the value of h_i is used to select the parameters for the Multinomial governing the Discrete distribution.

Figure 1 shows the factor graph of this model, which we call HMM-Crowd. Note that we assume that individual crowdworker labels are conditionally independent given the (hidden) true label.

A common problem with crowdsourcing models is data sparsity. For workers who provide only a few labels, it is hard to derive a good estimate of their confusion matrices. This is exacerbated when the label distribution is imbalanced, e.g., most words are not part of a named entity, concentrating the counts in a few confusion matrix entries. Solutions for this problem include hierarchical models of 'worker communities' (Venanzi et al., 2014) or correlations between confusion matrix entries (Nguyen et al., 2016). Although effective, these methods are also quite computationally expensive. For our models, to keep parameter estimation efficient, we use a simpler solution of 'collapsing' the confusion matrix into a 'confusion vector'. For worker j, instead of having the $n \times n$ matrix $\mathbf{C}^{(j)}$, we use the $n \times 1$ vector $\mathbf{C'}^{(j)}$ where $\mathbf{C'}^{(j)}_k$ is the probability of worker j labeling a word with true class k correctly. We also smooth the estimate of $\mathbf{C'}$ with prior counts as in (Liu and Wang, 2012; Kim and Ghahramani, 2012).

Learning: We use the Expectation Maximization (EM) algorithm (Dempster et al., 1977) to learn the parameters $(\boldsymbol{\tau}, \boldsymbol{\Omega}, \mathbf{C'})$, given the observations (all the words $\mathbf{V}$ and all the worker labels $\mathbf{L}$).

In the E-step, given the current estimates of the parameters, we take a forward and a backward

pass in the HMM to infer the hidden states, i.e. to calculate $p(h_i|\mathbf{V},\mathbf{L})$ and $p(h_i, h_{i+1}|\mathbf{V},\mathbf{L})$ for all appropriate i. Let $\alpha(h_i) = p(h_i, v_{1:i}, l_{1:i})$ where $v_{1:i}$ are the words from position 1 to i and $l_{1:i}$ are the crowd labels for these words from all workers. Similarly, let $\beta(h_i) = p(v_{i+1:n}, l_{i+1:n}|h_i)$. We have the recursions:

$$\alpha(h_i) = \sum_{h_{i-1}} p(v_i|h_i)p(h_i|h_{i-1}) \prod_j p(l_{ij}|h_i)\alpha(h_{i-1}) \tag{4}$$

$$\beta(h_i) = \sum_{h_{i+1}} p(h_{i+1}|h_i)p(v_{i+1}|h_{i+1}) \prod_j p(l_{i+1,j}|h_{i+1})\beta(h_{i+1}) \tag{5}$$

These are the standard α and β recursions for HMMs augmented with the crowd model: the product $\prod_j$ over the workers j who have provided labels for word i (or $i+1$). The posteriors can then be easily evaluated: $p(h_i|\mathbf{V},\mathbf{L}) \propto \alpha(h_i)\beta(h_i)$ and $p(h_i, h_{i+1}|\mathbf{V},\mathbf{L}) \propto \alpha(h_i)p(h_{i+1}|h_i)p(v_{i+1}|h_{i+1})\beta(h_{i+1})$

In the standard M-step, the parameters are estimated using maximum likelihood. However, we found a Variational Bayesian (VB) update procedure for the HMM parameters similar to (Johnson, 2007; Beal, 2003) provides some improvement and stability. We first define the Dirichlet priors over the transition and emission parameters:

$$p(\boldsymbol{\tau}_{h_i}) = \mathrm{Dir}(a_t) \tag{6}$$

$$p(\boldsymbol{\Omega}_{h_i}) = \mathrm{Dir}(a_e) \tag{7}$$

With these priors, the variational M-step updates the parameters as follows[2]:

$$\boldsymbol{\tau}_{h'|h} = \frac{\exp\{\Psi(\mathbf{E}_{h'|h} + a_t)\}}{\exp\{\Psi(\mathbf{E}_h + na_t)\}} \tag{8}$$

$$\boldsymbol{\Omega}_{v|h} = \frac{\exp\{\Psi(\mathbf{E}_{v|h} + a_e)\}}{\exp\{\Psi(\mathbf{E}_h + ma_e)\}} \tag{9}$$

where Ψ is the Digamma function, n is the number of states and m is the number of observations. $\mathbf{E}$ denotes the expected counts, calculated from the posteriors inferred in the E-step. $\mathbf{E}_{h'|h}$ is the expected number of times the HMM transitioned from state h to state h', where the expectation is with respect to the posterior distribution $p(h_i, h_{i+1}|\mathbf{V},\mathbf{L})$ that we infer in the E step:

$$\mathbf{E}_{h'|h} = \sum_i p(h_i = h, h_{i+1} = h'|\mathbf{V},\mathbf{L}) \tag{10}$$

[2]See Beal (2003) for the derivation and Johnson (2007) for further discussion for the Variational Bayesian approach.

Similarly, $\mathbf{E}_h$ is the expected number of times the HMM is at state h: $\mathbf{E}_h = \sum_i p(h_i = h|\mathbf{V},\mathbf{L})$ and $\mathbf{E}_{v|h}$ is the expected number of times the HMM emits the observation v from the state h: $\mathbf{E}_{v|h} = \sum_{i,v_i=v} p(h_i = h|\mathbf{V},\mathbf{L})$.

For the crowd parameters $\mathbf{C}'^{(j)}$, we use the (smoothed) maximum likelihood estimate:

$$\mathbf{C}'^{(j)}_k = \frac{\mathbf{E}^{(j)}_{k|k} + a_c}{\mathbf{E}^{(j)}_k + na_c} \tag{11}$$

where a_c is the smoothing parameter and $\mathbf{E}^{(j)}_{k|k}$ is the expected number of times that worker j correctly labeled a word of true class k as k while $\mathbf{E}^{(j)}_k$ is the expected total number of words belonging to class k worker j has labeled. Again, the expectation in $\mathbf{E}$ is taken under the posterior distributions that we infer in the E step.

3.2 Long Short Term Memory with Crowds

For Task 2, we extend the LSTM architecture (Hochreiter and Schmidhuber, 1997) for NER (Lample et al., 2016) to account for noisy crowdsourced labels (this can be easily adapted to other sequence labeling tasks). In this model, the sentence input is first fed into an LSTM block (which includes character- and word-level bi-directional LSTM units). The LSTM block's output then becomes input to a (fully connected) hidden layer, which produces a vector of tags scores for each word. This *tag score* vector is the word-level prediction, representing the likelihood of the word being from each tag. All the tags scores are then fed into a 'CRF layer' that 'connects' the word-level predictions in the sentence and produces the final output: the most likely sequence of tags.

We introduce a crowd representation in which a worker vector represents the noise associated with her labels. In other words, the parameters in the original architecture learns the correct sequence labeling model while the crowd vectors add noise to its predictions to 'explain' the lower quality of the labels. We assume a perfect worker has a zero vector as her representation while an unreliable worker is represented by a large magnitude vector. At test time, we ignore the crowd component and make predictions by feeding the unlabeled sentence into the original LSTM architecture. At train time, an example consists of the labeled sentence and the ID of the worker who provided the labels. Worker IDs are mapped to vector representations and incorporated into the LSTM architecture.

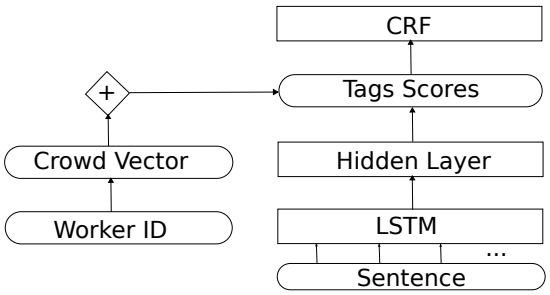

Figure 2: The LSTM-Crowd model. The Crowd Vector is added (element-wise) to the Tags Scores.

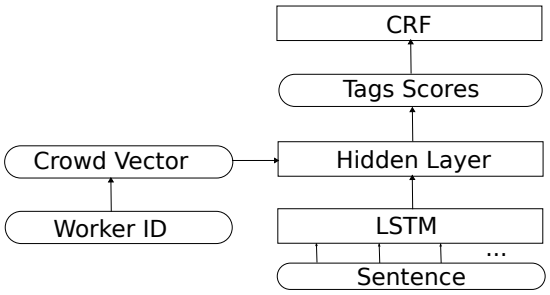

Figure 3: The LSTM-Crowd-cat model. The crowd vectors provide additional input for the Hidden Layer (they are effectively concatenated to the output of the LSTM block).

We propose two strategies for incorporating the crowd vector into the LSTM: (1) adding the crowd vector to the tags scores and (2) concatenating the crowd vector to the output of the LSTM block.

LSTM-Crowd. The first strategy is illustrated in **Figure 2**. We set the dimension of the crowd vectors to be equal to the number of tags and the addition is element-wise. In this strategy, the crowd vectors have a nice interpretation: the tag-conditional noise for the worker. This is useful for worker evaluation and intelligent task routing (i.e. assigning the right work to the right worker).

LSTM-Crowd-cat. The second strategy is illustrated in **Figure 3**. We set the crowd vectors to be additional inputs for the Hidden Layer (along with the LSTM block output). In this way, we are free to set the dimension of the crowd vectors and we have a more flexible model of worker noise. This comes with a cost of reduced interpretability and additional parameters in the hidden layer.

For both strategies, the crowd vectors are randomly initialized and learned in the same LSTM architecture using Back Propagation (Rumelhart et al., 1985) and Stochastic Gradient Descent (SGD) (Bottou, 2010).

Dataset	Application	Size	Gold	Crowd
CoNLL'03	NER	1393	All	400
Medical	IE	5000	200	All

Table 1: Datasets used for each application. We list the total number of articles/abstracts and the number which have Gold/Crowd labels.

4 Evaluation Setup

4.1 Datasets & Tuning

NER. We use the English portion of the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003), which includes over 21,000 annotated sentences from 1,393 news articles split into 3 sets: train, validation and test. We also use crowd labels collected by Rodrigues et al. (2014) for 400 articles in the train set[3]. For Task 1 (aggregating crowd labels), to avoid overfitting, we split these 400 articles into 50% validation and 50% test[4]. For Task 2 (predicting sequences on unannotated text), we follow Rodrigues et al. (2014) in using the CoNLL validation and test sets.

Biomedical IE. We use 5,000 medical paper abstracts describing randomized control trials (RCTs) involving people. Each abstract is annotated by roughly 5 Amazon Mechanical Turk workers. Annotators were asked to mark all text spans in a given abstract which identify the population enrolled in the clinical trial. The annotations are therefore binary: inside or outside a span. In addition to annotations collected from laypeople via Mechanical Turk, we also use gold annotations by medical students for a small set of 200 abstracts, which we split into 50% validation and 50% test. For Task 1, we run methods being compared on all 5,000 abstracts, but we evaluate them only using the validation/test set. For Task 2, the validation and test sets are held out. **Table 1** presents key statistics of datasets used.

Tuning: In all experiments, validation set results are used to tune the models hyper-parameters. For HMM-Crowd, we have a smoothing parameter and two Dirichlet priors. For our two LSTMs, we have a L2 regularization parameter. For LSTM-Crowd-cat, we also have the crowd vector dimen-

[3] http://www.fprodrigues.com/software/crf-ma-sequence-labeling-with-multiple-annotators/

[4] Rodrigues et al. (2014)'s results on the 'training set' are not directly comparable to ours since they do not partition the crowd labels into validation and test sets.

sion. For each hyper-parameter, we consider a few (less then 5) different parameter settings for light tuning. We report results achieved on the test set.

4.2 Baselines

Task 1. For aggregating crowd labels, we consider the following baselines:

- Majority Voting (MV) at the token level. Rodrigues et al. (2014) show that this generally performs better than MV at the entity level.
- Dawid and Skene (1979) weighted voting at the token level. We tested both a popular public implementation[5] of Dawid-Skene and our own and found that ours performed better (likely due to smoothing), so we report it.
- MACE (Hovy et al., 2013), using the authors' public implementation[6].
- *Dawid-Skene then HMM*. We propose a simple heuristic to aggregate sequential crowd labels: (1) use Dawid and Skene (1979) to induce consensus labels from individual crowd labels; (2) train a HMM using the input text and consensus labels; and then (3) use the trained HMM to predict and output labels for the input text. We also tried using a CRF or LSTM as the sequence labeler but found the HMM performed best. This is not surprising: CRFs and LSTM are good at predicting unseen sequences, whereas the predictions here are on the seen training sequences.
- Rodrigues et al. (2014)'s CRF with Multiple Annotators (CRF-MA). We use the source code provided by the authors.
- **?**'s Interval-dependent (ID) HMM using the authors' source code[7]. Since they assume binary labels, we can only apply this to the biomedical IE task.

For non-sequential aggregation baselines, we evaluate majority voting (MV) and Dawid and Skene (1979) as perhaps the most widely known and used in practice. A recent benchmark evaluation of aggregation methods for (non-sequential) crowd labels found that classic Dawid-Skene was the most consistently strong performing method among those considered, despite its age, while majority voting was often outperformed by other methods (Sheshadri and Lease, 2013).

Dawid and Skene (1979) models a confusion matrix for each annotator, using EM estimation of these matrices as parameters and the true token labels as hidden variables. This is roughly equivalent to our proposed HMM-Crowd model (Section 3), but without the HMM component.

Task 2. To predict sequences on unannotated text when trained on crowd labels, we consider two broad approaches: (1) directly train the model on all individual crowd annotations; and (2) induce consensus labels via Task 1 and train on them.

For approach (1), we report as baselines:

- Rodrigues et al. (2014)'s CRF-MA
- Lample et al. (2016)'s LSTM trained on all crowd labels (ignoring worker IDs)

For approach (2), we report as baselines:

- Majority Voting (MV) then Conditional Random Field (CRF). We train the CRF using the *CRF Suite* package (Okazaki, 2007) with the same features as in Rodrigues et al. (2014), who also report this baseline.
- Lample et al. (2016)'s LSTM trained on Dawid-Skene (DS) consensus labels.

4.3 Metrics

NER. We use the CoNLL 2003 metrics of entity-level precision, recall and F1. The predicted entity must match the gold entity exactly (i.e. no partial credit is given for partial matches).

Biomedical IE. The above metrics are overly strict for the biomedical IR task, in which annotated sequences are typically far longer than named-entities. We therefore 'relax' the metric to credit partial matches as follows. For each predicted positive contiguous text span, we calculate:

$$\text{Precision} = \frac{\text{\# true positive words}}{\text{\# words in this predicted span}}$$

For example, for a predicted span of 10 words, if 6 words are truly positive, the Precision is 60%. We evaluate this 'local' precision for each predicted span and then take the average as the 'global' precision. Similarly, for each gold span, we calculate:

$$\text{Recall} = \frac{\text{\# words in a predicted span}}{\text{\# words in this gold span}}$$

[5] https://github.com/ipeirotis/Get-Another-Label
[6] http://www.isi.edu/publications/licensed-sw/mace/
[7] https://academiccommons.columbia.edu/catalog/ac:199939

Method	Precision	Recall	F1
Majority Vote	78.35	56.57	65.71
MACE	65.10	69.81	67.37
Dawid-Skene (DS)	78.05	65.78	71.39
CRF-MA	**80.29**	51.20	62.53
DS then HMM	76.81	71.41	74.01
HMM-Crowd	77.40	**72.29**	**74.76**

Table 2: **NER** results for Task 1 (crowd label aggregation). Rows 1-3 show non-sequential methods while Rows 4-6 show sequential methods.

The recall scores for all the gold spans are again averaged to get a global recall score.

For the biomedical IE task, because we have gold labels for only a small set of 200 abstracts, we create 100 bootstrap re-samples of the (predicted and gold) spans and perform the evaluation for each re-sample. We then report the mean and standard deviation over these 100 re-samples.

5 Evaluation Results

5.1 Named-Entity Recognition (NER)

Table 2 presents Task 1 results for aggregating crowd labels. For the non-sequential aggregation baselines, we see that Dawid and Skene (1979) outperforms both majority voting and MACE (Hovy et al., 2013). For sequential methods, our heuristic 'Dawid-Skene then HMM' method performs surprisingly well, nearly as well as HMM-Crowd. However, we will see that this heuristic does not work as well for biomedical IR.

Rodrigues et al. (2014)'s CRF-MA achieves the highest Precision of all methods, but surprisingly the lowest F1. We use their public implementation but observe different results from what they report (we observed similar results when using all the crowd data without validation/test split as they do). We suspect their released source code may be optimized for Task 2, though we could not reach the authors to verify this.

Table 3 reports NER results for Task 2: predicting sequences on unannotated text when trained on crowd labels. Results for Rodrigues et al. (2014)'s CRF-MA are reproduced using their public implementation and match their reported results. While CRF-MA outperforms 'Majority Vote then CRF' as the authors reported, and achieves the highest Recall of all methods, its F1 results are generally not competitive with other methods.

Methods based on Lample et al. (2016)'s LSTM generally outperform the CRF methods. Adding a crowd component to the LSTM yields marked improvement of 2.5-3 points F1 vs. the LSTM trained on individual crowd annotations or consensus MV annotations. LSTM-Crowd (trained on individual labels) and 'HMM-Crowd then LSTM' (LSTM trained on HMM consensus labels) offer different paths to achieving comparable, best results.

5.2 Biomedical Information Extraction (IE)

Tables 4 and **5** present Biomedical IE results for Tasks 1 and 2, respectively. We were unable to run Rodrigues et al. (2014)'s CRF-MA public implementation on the Biomedical IE dataset (due to an 'Out of Memory Error' with 8gb max heapsize).

For Task 1, Majority Vote achieves nearly 92% Precision but suffers from very low Recall. As with NER, HMM-Crowd achieves the highest Recall and F1, showing 2 points F1 improvement here over non-sequential Dawid and Skene (1979). In contrast with the NER results, our heuristic 'Dawid-Skene then HMM' performs much worse for Biomedical IE. In general, we expect heuristics to be less robust than principled methods.

For Task 2, as with NER, we again see that LSTM-Crowd (trained on individual labels) and 'HMM-Crowd then LSTM' (LSTM trained on HMM consensus labels) offer different paths to achieving fairly comparable results. While LSTM-Crowd-cat again achieves slightly lower F1, simply training Lample et al. (2016)'s LSTM directly on all crowd labels performs much better than seen earlier with NER, likely due to the relatively larger size of this dataset (see Table 1). To further investigate, we study the performances of these LSTM models as a function of training data available. In **Figure 4**, we see that as the amount of training data decreases, our crowd-augmented LSTM models produce greater relative improvement compared to the original LSTM architecture.

Table 6 presents an example from Task 1 of a sentence with its gold span, annotations and the outputs from Dawid-Skene and HMM-Crowd. Dawid-Skene aggregates labels based only on the crowd labels while our HMM-Crowd combines that with a sequence model. HMM-Crowd is able to return the longer part of the correct span.

Method	Precision	Recall	F1
CRF-MA (Rodrigues et al., 2014)	49.40	**85.60**	62.60
LSTM (Lample et al., 2016)	**83.19**	57.12	67.73
LSTM-Crowd	82.38	62.10	70.82
LSTM-Crowd-cat	79.61	62.87	70.26
Majority Vote then CRF	45.50	80.90	58.20
Dawid-Skene then LSTM	72.30	61.17	66.27
HMM-Crowd then CRF	77.40	61.40	68.50
HMM-Crowd then LSTM	76.19	66.24	**70.87**
LSTM on Gold Labels (upper-bound)	85.27	83.19	84.22

Table 3: **NER** results on Task 2: predicting sequences on unannotated text when trained on crowd labels. Rows 1-4 train the predictive model using individual crowd labels, while Rows 5-8 first aggregate crowd labels then train the model on the induced consensus labels. The last row indicates an upper-bound from training on gold labels. LSTM-Crowd and LSTM-Crowd-cat are described in Section 3.

Method	Precision	Recall	F1	std
Majority Vote	**91.89**	48.03	63.03	2.6
MACE	45.01	88.49	59.63	1.7
Dawid-Skene	77.85	66.77	71.84	1.7
Dawid-Skene then HMM	72.49	58.77	64.86	2.0
ID HMM (**?**)	78.99	68.10	73.11	1.9
HMM-Crowd	72.81	**75.14**	**73.93**	1.8

Table 4: **Biomedical IE** results for Task 1: aggregating sequential crowd labels to induce consensus labels. Rows 1-3 indicate non-sequential baselines. Results are averaged over 100 bootstrap re-samples. We report the standard deviation of F1, *std*, due to this dataset having fewer gold labels for evaluation.

Method	Precision	Recall	F1	std
LSTM (Lample et al., 2016)	77.43	61.13	68.27	1.9
LSTM-Crowd	73.83	63.93	68.47	1.6
LSTM-Crowd-cat	68.08	**68.41**	68.20	1.8
Majority Vote then CRF	**93.71**	33.16	48.92	2.8
Dawid-Skene then LSTM	70.21	65.26	67.59	1.7
HMM-Crowd then CRF	79.54	54.76	64.81	2.0
HMM-Crowd then LSTM	73.65	64.64	**68.81**	1.9

Table 5: **Biomedical IE** results for Task 2. Rows 1-3 correspond to training on all labels, while Rows 4-7 first aggregate crowd labels then train the sequence labeling model on consensus annotations.

Gold	... was as safe and effective as ... for the empiric treatment of acute invasive diarrhea in ambulatory pediatric patients requiring an emergency room visit
Annotations (2 out of 5)	... was as safe and effective as ... for the empiric treatment of acute invasive diarrhea in ambulatory pediatric patients requiring an emergency room visit
Dawid-Skene	... was as safe and effective as ... for the empiric treatment of acute invasive diarrhea in ambulatory pediatric patients requiring an emergency room visit
HMM-Crowd	... was as safe and effective as ... for the empiric treatment of acute invasive diarrhea in ambulatory pediatric patients requiring an emergency room visit

Table 6: An example from the medical abstract dataset for task 1: inferring true labels. Out of 5 annotations, only 2 have identified a positive span (the other 3 are empty). Dawid-Skene is able to assign higher weights to the minority of 2 annotations to return a part of the correct span. HMM-Crowd returns a longer part of the span, which we believe is due to useful signal from its sequence model.

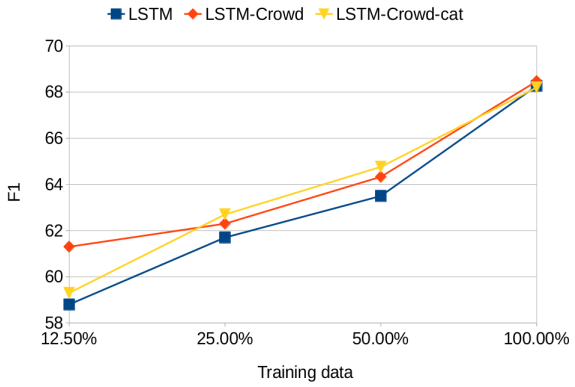

Figure 4: F1 scores in Task 2 for biomedical IE with varying percentages of training data.

6 Conclusions and Future Work

Given a dataset of crowdsourced sequence labels, we presented novel methods to: (1) aggregate sequential crowd labels to infer a best single set of consensus annotations; and (2) use crowd annotations as training data for a model that can predict sequences in unannotated text. We evaluated our approaches on two datasets representing different domains and tasks: general NER and biomedical IE. Results showed that our methods show improvement over strong baselines.

We expect our methods to be applicable to and similarly benefit other sequence labeling tasks, such as POS tagging and chunking (Hovy et al., 2014). Our methods also provide an estimate of each worker's label quality, which can be transfered between tasks and is useful for error analysis and intelligent task routing (Bragg et al., 2014). We also plan to investigate extension of the crowd component in our HMM method with hierarchical models, as well as a fully-Bayesian approach.

Acknowledgements

We thank the reviewers for their valuable comments. This work is supported in part by by National Science Foundation grant No. 1253413 and the National Cancer Institute (NCI) of the National Institutes of Health (NIH), award number UH2CA203711. 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

Matthew James Beal. 2003. *Variational algorithms for approximate Bayesian inference*. University of London United Kingdom.

Wei Bi, Liwei Wang, James T. Kwok, and Zhuowen Tu. 2014. Learning to predict from crowdsourced data. In *Uncertainty in Artificial Intelligence*.

Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*. Association for Computational Linguistics, pages 194–201. https://doi.org/10.3115/974557.974586.

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, Springer, pages 177–186.

Jonathan Bragg, Andrey Kolobov, Mausam Mausam, and Daniel S Weld. 2014. Parallel task routing for crowdsourcing. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Hai Leong Chieu and Hwee Tou Ng. 2002. Named entity recognition: a maximum entropy approach using global information. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguis-

tics, pages 1–7. http://aclweb.org/anthology/C02-1025.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics* pages 20–28.

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)* pages 1–38.

Mark Dredze, Partha Pratim Talukdar, and Koby Crammer. 2009. Sequence learning from data with multiple labels. In *ECML-PKDD 2009 workshop on Learning from Multi- Label Data*.

Paul Felt, Eric Ringger, Kevin Seppi, and Robbie Haertel. 2015. Early gains matter: A case for preferring generative over discriminative crowdsourcing models. In *Conference of the North American Chapter of the Association for Computational Linguistics*. https://doi.org/10.3115/v1/N15-1089.

Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Association for Computational Linguistics, pages 80–88.

Sharon Goldwater and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Annual meeting-association for computational linguistics*. Citeseer, volume 45, page 744. http://aclweb.org/anthology/P07-1094.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with mace. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 1120–1130. http://aclweb.org/anthology/N13-1132.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. Experiments with crowdsourced re-annotation of a pos tagging data set. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pages 377–382. https://doi.org/10.3115/v1/P14-2062.

Xiaoli Huang, Jimmy Lin, and Dina Demner-Fushman. 2006. PICO as a Knowledge Representation for Clinical Questions. In *AMIA 2006 Symposium Proceedings*. pages 359–363.

Ziheng Huang, Jialu Zhong, and Rebecca J. Passonneau. 2015. Estimation of discourse segmentation labels from crowd data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 2190–2200. http://aclweb.org/anthology/D15-1261.

Mark Johnson. 2007. Why doesn't em find good hmm pos-taggers? In *EMNLP-CoNLL*. pages 296–305. http://aclweb.org/anthology/D07-1031.

Hiroshi Kajino, Yuta Tsuboi, and Hisashi Kashima. 2012. A convex formulation for learning from crowds. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *International conference on artificial intelligence and statistics*. pages 619–627.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1746–1751. http://www.aclweb.org/anthology/D14-1181.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*. volume 1, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 260–270. https://doi.org/10.18653/v1/N16-1030.

Chao Liu and Yi-min Wang. 2012. Truelabel+ confusions: A spectrum of probabilistic models in analyzing multiple ratings. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*. pages 225–232.

Qiang Liu, Jian Peng, and Alex T Ihler. 2012. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems*. pages 692–700.

Andrew McCallum and Wei Li. 2003. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*,

chapter Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. http://aclweb.org/anthology/W03-0430.

An T Nguyen, Byron C Wallace, and Matthew Lease. 2016. A correlated worker model for grouped, imbalanced and multitask data. In *Uncertainty in Artificial Intelligence*.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs). http://www.chokkan.org/software/crfsuite/.

Lawrence Rabiner and B Juang. 1986. An introduction to hidden markov models. *ieee assp magazine* 3(1):4–16.

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *Journal of Machine Learning Research* 11(Apr):1297–1322.

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

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, DTIC Document.

Connie Schardt, Martha B Adams, Thomas Owens, Sheri Keitz, and Paul Fontelo. 2007. Utilization of the PICO framework to improve searching PubMed for clinical questions. *BMC medical informatics and decision making* 7(1):16.

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

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 254–263. http://aclweb.org/anthology/D08-1027.

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

Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. 2014. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*. ACM, pages 155–164.

Byron C Wallace, Joël Kuiper, Aakash Sharma, Mingxi Brian Zhu, and Iain J Marshall. 2016. Extracting pico sentences from clinical trial reports using supervised distant supervision. *Journal of Machine Learning Research* 17(132):1–25.

Hui Yang, Anton Mityagin, Krysta M Svore, and Sergey Markov. 2010. Collecting high quality overlapping labels at low cost. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 459–466.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* .