Mach Learn (2014) 95:165–181
DOI 10.1007/s10994-013-5411-2

Sequence labeling with multiple annotators

Filipe Rodrigues · Francisco Pereira · Bernardete Ribeiro

Received: 13 November 2012 / Accepted: 17 August 2013 / Published online: 4 October 2013

Abstract The increasingly popular use of *Crowdsourcing* as a resource to obtain labeled data has been contributing to the wide awareness of the machine learning community to the problem of supervised learning from multiple annotators. Several approaches have been proposed to deal with this issue, but they disregard sequence labeling problems. However, these are very common, for example, among the Natural Language Processing and Bioinformatics communities. In this paper, we present a probabilistic approach for sequence labeling using Conditional Random Fields (CRF) for situations where label sequences from multiple annotators are available but there is no actual ground truth. The approach uses the Expectation-Maximization algorithm to jointly learn the CRF model parameters, the reliability of the annotators and the estimated ground truth. When it comes to performance, the proposed method (CRF-MA) significantly outperforms typical approaches such as majority voting.

Keywords Multiple annotators · Crowdsourcing · Conditional random fields · Latent variable models · Expectation maximization

Editor: Hal Daume, III.

F. Rodrigues (✉) · B. Ribeiro
Centre for Informatics and Systems of the University of Coimbra (CISUC), Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
e-mail: fmpr@dei.uc.pt

B. Ribeiro
e-mail: bribeiro@dei.uc.pt

F. Pereira
Singapore-MIT Alliance for Research and Technology (SMART), 1 CREATE Way, Singapore 138602, Singapore
e-mail: camara@smart.mit.edu

1 Introduction

The increasing awareness of the importance of *Crowdsourcing* (Howe 2008) as a means of obtaining labeled data is promoting a shift in machine learning towards models that are annotator-aware. A good example is that of online platforms such as Amazon's Mechanical Turk (AMT).[1] These platforms provide an accessible and inexpensive resource to obtain labeled data, whose quality, in many situations, competes directly with the one of "experts" (Snow et al. 2008; Novotney and Callison-Burch 2010). Also, by distributing a labeling task by multiple annotators it can be completed in a considerably smaller amount of time. For such reasons, these online work-recruiting platforms are rapidly changing the way datasets are built.

Furthermore, the social web promotes an implicit form of *Crowdsourcing*, as multiple web users interact and share contents (e.g., document tags, product ratings, opinions, user clicks, etc.). As the social web expands, so does the need for annotator-aware models.

On another perspective, there are tasks for which ground truth labels are simply very hard to obtain. Consider for instance the tasks of Sentiment Analysis, Movie Rating or Keyphrase Extraction. These tasks are subjective in nature and hence the definition of ground truth requires very strict guidelines, which can be very hard to achieve and follow. Even in well studied tasks like Named Entity Recognition linguists argue what should and should not be considered a named entity and consensus is not easily obtained. In cases where the task is inherently subjective an attainable goal is to build a model that captures the *wisdom of the crowds* (Surowiecki 2004) as good as possible while paying less attention to dissonant views.

Another example can be found in the field of medical diagnosis, where obtaining ground truth can mean expensive or invasive medical procedures like biopsies. On the other hand, it is much simpler for a physician to consult his colleagues for an opinion, resulting in a multiple "experts" scenario.

Sequence labeling refers to the supervised learning task of assigning a label to each element of a sequence. Typical examples are Part-of-Speech tagging, Named Entity Recognition and Gene Prediction (Allen et al. 2004; Allen and Salzberg 2005). In such tasks, the individual labels cannot be considered as detached from the context (i.e. the preceding and succeeding elements of the sequence and their corresponding labels). Two of the most popular sequence models are hidden Markov models (HMM) (Rabiner 1989) and Conditional Random Fields (CRF) (Lafferty et al. 2001). Due to the usually high dimensional feature spaces (specially considering CRFs), these models frequently require large amounts of labeled data to be properly trained, which hinders the construction and release of datasets and makes it almost prohibitive to do with a single annotator. Although in some domains, the use of unlabeled data can help in making this problem less severe (Bellare and Mccallum 2007), a more natural solution is to rely on multiple annotators. For example, for many tasks, AMT can be used to label large amounts of data (Callison-Burch and Dredze 2010). However, the large numbers needed to compensate for the heterogeneity of annotators expertise rapidly raise its actual cost beyond acceptable values. A parsimonious solution needs to be designed that is able to deal with such real world constraints and heterogeneity.

In the past few years many approaches have been proposed that deal with the problem of supervised learning from multiple annotators in different paradigms (classification, regression, ranking, etc.), however the particular problem of sequence labeling from multiple

[1] http://www.mturk.com.

annotators was practically left untouched, and most of the applications typically rely on majority voting (e.g. Laws et al. 2011). Given its importance in such fields as Natural Language Processing, Bioinformatics, Computer Vision, Speech and Ubiquitous Computing, sequence labeling from multiple annotators is a very important problem. Unfortunately, due to its nature, typical approaches proposed for binary or categorical classification cannot be directly applied for sequences.

In this paper we propose a probabilistic approach using the Expectation-Maximization algorithm (EM) for sequence labeling using CRFs for the scenario where we have multiple annotators providing labels with different levels of "reliability" but no actual ground truth. The proposed method is able to jointly learn the CRF model parameters, the reliabilities of the annotators and the estimated ground truth label sequences. It is empirically shown that this method outperforms the baselines even in situations of high levels of noise in the labels of the annotators and when the less "trustworthy" annotators dominate. The proposed approach also has the advantage of not requiring repeated labeling of the same input sequences by the different annotators. Finally, this approach can be easily modified to work with other sequence labeling models like HMMs.

2 Related work

The first works that relate to the problem of learning from multiple annotators go back to 1979 when Dawid and Skene proposed an approach for estimating the error rates of multiple patients (annotators) given their responses (labels) to multiple medical questions. Although this work just focused on estimating the hidden ground truth labels, it inspired other works where there is an explicit attempt to learn a classifier. For example, Smyth et al. (1995) propose a similar approach to solve the problem of volcano detection and classification in Venus imagery with data labelled by multiple experts. Like in previous works, their approach relies on a latent variable model, where they treat the ground truth labels as latent variables. The main difference is that the authors use the estimated (probabilistic) ground truth to explicitly learn a classifier.

More recently, Snow et al. (2008) demonstrated that learning from labels provided by multiple non-expert annotators can be as good as learning from the labels of one expert. Such kind of findings inspired the development of new approaches that, unlike previous ones (Smyth et al. 1995; Donmez and Carbonell 2008; Sheng et al. 2008), do not rely on repeated labeling, i.e. having the same annotators labeling the same set of instances. In Raykar et al. (2009, 2010) an approach is proposed where the classifier and the annotators reliabilities are learnt jointly. Later works then relaxed the assumption that the annotators' reliabilities do not depend on the instances they are labeling (Yan et al. 2010), and extended the proposed methodology to an active learning scenario (Yan et al. 2011). All these approaches shared a few key aspects: (1) they use a latent variable model where the ground truth labels are treated as latent variables; (2) they rely on the EM algorithm (Dempster et al. 1977) to find maximum likelihood estimates for the model parameters; and (3) they deal mostly with binary classification problems (although some suggest extensions to handle categorical, ordinal and even continuous data).

The acclaimed importance of supervised learning from multiple annotators lead to many interesting alternative approaches and variations/extensions of previous works in the past couple of years. In Donmez et al. (2010) the authors propose the use of a particle filter to model the time-varying accuracies of the different annotators. Groot et al. (2011) propose a annotator-aware methodology for the regression problems using Gaussian processes, and Wu et al. (2011) present a solution for ranking problems with multiple annotators.

Despite the variety of approaches presented for different learning paradigms, the problem of sequence labeling from multiple annotators was left practically untouched, with the only relevant work being the work by Dredze et al. (2009). In this work the authors propose a method for learning structured predictors, namely CRFs, from instances with multiple labels in the presence of noise. This is achieved by modifying the CRF objective function used for training through the inclusion of a per-label prior, thereby restricting the model from straying too far from the provided priors. The per-label priors are then re-estimated by making use of their likelihoods under the whole dataset. In this way, the model is capable of using knowledge from other parts of the dataset to prefer certain labels over others. By iterating between the computation of the expected values of the label priors and the estimation of the model parameters in an EM-like style, the model is expected to give preference to the less noisy labels. Hence, we can view this process as self-training, a process whereby the model is trained iteratively on its own output. Although this approach makes the model computationally tractable, their experimental results indicate that the method only improves performance in scenarios where there is a small amount of training data (low quantity) and when the labels are noisy (low quality).

It is important to stress that, contrarily to the model proposed in this paper, the model by Dredze et al. (2009) is a multi-label model, and not a multi-annotator model, in the sense that the knowledge about who provided the multiple label sequences is completely discarded. The obvious solution for including this knowledge would be to use a latent ground truth model similar to the one proposed by Raykar et al. (2009, 2010), thus extending this work to sequence labeling tasks. However, treating the ground truth label sequences as latent variables and using the EM algorithm to estimate the model parameters would be problematic, since the number of possible label sequences grows exponentially with the length of the sequence, making the marginalization over the latent variables intractable. In contrast to this, the approach presented in this paper avoids this problem by treating the annotators reliabilities as latent variables, making the marginalization over the latent variables tractable (see Sect. 3).

In the field of Bioinformatics a similar problem has been attracting attention, in which multiple sources of evidence are combined for gene prediction (e.g. Allen et al. 2004; Allen and Salzberg 2005). In these approaches the outputs of multiple predictors (e.g. HMMs) are usually combined using a voting of the labels predicted, weighted by the confidence (posteriors) of the various sources in their predictions (Allen et al. 2004). Non-linear decision schemes also exist, for example using Decision Trees (Allen and Salzberg 2005), but similarly to the linearly weighted voting schemes, the confidence weights are estimated once and never corrected. This contrasts with the approaches discussed in this paper, where the goal is to build a single predictor (CRF) from the knowledge of multiple annotators (sources), and where the confidence of each source is iteratively re-estimated.

3 Approach

3.1 Measuring the reliability of the annotators

Let $\mathbf{y}^r$ be a sequence of labels assigned by the r^{th} annotator to some observed input sequence $\mathbf{x}$. If we were told the actual (unobserved) sequence of true labels $\mathbf{y}$ for that same input sequence $\mathbf{x}$, we could evaluate the quality (or reliability) of the r^{th} annotator in a dataset by measuring its precision and recall. Furthermore, we could combine precision and recall in a single measure by using the traditional F1-measure, and use this combined measure to evaluate how "good" or "reliable" a given annotator is according to some ground

truth. In practice any appropriate loss function can be used to evaluate the quality of the annotators. The choice of one metric over others is purely problem-specific. The F-measure was used here due to its wide applicability in sequence labeling problems and, particularly, in the tasks used in the experiments (Sect. 4).

3.2 Sequence labeling

If for a dataset of N input sequences $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ we knew the actual ground truth label sequences $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N$, we could model the probabilities of the label sequences $\mathcal{Y}$ given the input sequences $\mathcal{X}$ using a linear-chain Conditional Random Field (CRF) (Lafferty et al. 2001).

In a linear-chain CRF the conditional probability of a sequence of labels $\mathbf{y}$ given a sequence of observations $\mathbf{x}$ is given by

$$p_{crf}(\mathbf{y}|\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{\Psi(\mathbf{x})} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}, t) \right\} \tag{1}$$

where $\Psi(\mathbf{x})$ is a normalization constant that makes the sum of the probability of all label sequences equal to one, $f_k(y_{t-1}, y_t, \mathbf{x}, t)$ is a feature function (often binary-valued, but that can also be real-valued) and λ_k is a learned weight associated with feature f_k. The feature functions can capture any aspect of the state transitions $y_{t-1} \rightarrow y_t$ and of the whole input sequence $\mathbf{x}$, which in fact, can be used to understand the relationship between labels and the characteristics of the whole input sequence $\mathbf{x}$ at a given moment t.

According to the model defined in Eq. (1), the most probable labeling sequence for an input sequence $\mathbf{x}$ is given by $\mathbf{y}^* = \arg\max_{\mathbf{y}} p_{crf}(\mathbf{y}|\mathbf{x}, \boldsymbol{\lambda})$, which can be efficiently determined through dynamic programming using the Viterbi algorithm.

The parameters $\boldsymbol{\lambda}$ of the CRF model are typically estimated from an i.i.d. dataset by maximum likelihood using limited-memory BFGS (Liu and Nocedal 1989). The loglikelihood for a dataset $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ is given by $\sum_{i=1}^N \ln p_{crf}(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\lambda})$.

3.3 Maximum likelihood estimator

Since we do not know the set of actual ground truth label sequences $\mathcal{Y}$ for the set of input sequences $\mathcal{X}$, we must find a way to estimate it using the sets of label sequences provided by the R different annotators $\{\mathcal{Y}^1, \ldots, \mathcal{Y}^R\}$, and learn a CRF model along the way.

Let the observed data $\{\mathbf{y}_i^1, \ldots, \mathbf{y}_i^R, \mathbf{x}_i\}_{i=1}^N$ be denoted by $\mathcal{D}$. Given this data, and assuming the instances are i.i.d., the likelihood function, for some parameters θ (their definition can be ignored for now) can be factored as

$$p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p\left(\mathbf{y}_i^1, \ldots, \mathbf{y}_i^R|\mathbf{x}_i, \theta\right). \tag{2}$$

We now introduce a random vector $\mathbf{z}$ that represents the reliability of the annotators. We can define $\mathbf{z}$ to be an R-dimensional vector with values $\{z^1, \ldots, z^R\}$, so that $\mathbf{z} \sim \mathit{Multinomial}(\pi_1, \ldots, \pi_R)$ with probability

$$p(\mathbf{z}|\boldsymbol{\pi}) = \prod_{r=1}^{R} (\pi_r)^{z^r} \tag{3}$$

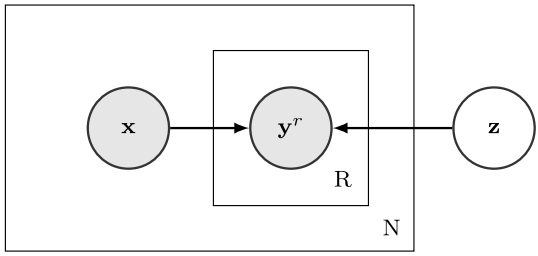

Fig. 1 Plate representation of proposed model

where we made the parameters of the multinomial ($\boldsymbol{\pi}$) explicit. If we define z^r to be the F1-measure of the r^{th} annotator, the parameters $\boldsymbol{\pi}$ of the multinomial are then defined as

$$\pi_r = p\big(z^r = 1\big) = \frac{F1\text{-}measure_r}{\sum_{j=1}^{R} F1\text{-}measure_j} \tag{4}$$

thus ensuring the constraints $\pi_r \geq 0$ (since the F1-measure is always non-negative) and $\sum_r \pi_r = 1$.[2] The expectation of this multinomial random variable, $\mathbb{E}\{z^r\} = p(z^r = 1)$, can be interpreted as the probability picking the label sequences provided by the r^{th} annotator as the correct ones (i.e. for which $F1\text{-}measure_r = 1$) and using those for training. An analogy for this model would be a student picking a book to learn about some subject. The student is provided by the University's library with a set of books that cover that subject but differ only in the correctness of the content. The student then has to pick one of the books from which to learn about that subject. Transferring this analogy back to our multiple annotator setting, the random vector $\mathbf{z}$ can be viewed as picking the best annotator from which to learn from, thus enforcing competition among the annotators. The correct annotator is assumed to provide label sequences according to $p_{crf}(\mathbf{y}_i^r|\mathbf{x}_i, \lambda)$. The others are assumed to provide incorrect labels which we assume to come from a random model $p_{rand}(\mathbf{y}_i^r|\mathbf{x}_i)$). The generative process can then be summarized as follows:

1. draw $\mathbf{z} \sim Multinomial(\pi_1, \ldots, \pi_R)$
2. for each instance $\mathbf{x}_i$:
   (a) for each annotator r:
      (i) if $z^r = 1$, draw $\mathbf{y}_i^r$ from $p_{crf}(\mathbf{y}_i^r|\mathbf{x}_i, \boldsymbol{\lambda})$
      (ii) if $z^r = 0$, draw $\mathbf{y}_i^r$ from $p_{rand}(\mathbf{y}_i^r|\mathbf{x}_i)$

Figure 1 shows a plate representation of the proposed model.

For the sake of simplicity, we assume the random model $p_{rand}(\mathbf{y}_i^r|\mathbf{x}_i)$ to be uniformly distributed, hence

$$p_{rand}\big(\mathbf{y}_i^r|\mathbf{x}_i\big) = \prod_{t=1}^{T} \frac{1}{\mathcal{C}} \tag{5}$$

where T denotes the length of the sequence and $\mathcal{C}$ is the number of possible classes/labels for a sequence element.

Although it might seem too restrictive to assume that only one annotator provides the correct label sequences, it is important to note that the model can still capture the uncertainty

[2]These constraints are required for the Jensen's inequality to apply and for the EM algorithm presented in Sect. 3.4 to be valid.

in who the correct annotator should be. In alternative to this approach, one could replace the multinomial random variable with multiple Bernoullis (one for each annotator). From a generative perspective, this would allow for multiple annotators to be correct. However, this places too much emphasis on the form of $p_{rand}(\mathbf{y}_i^r|\mathbf{x}_i)$, since it would be crucial for deciding whether the annotator is likely to be correct. One the other hand, as we shall see later, by using a multinomial, the probabilities $p_{rand}(\mathbf{y}_i^r|\mathbf{x}_i)$ cancel out from the updates of the annotators reliabilities, thus forcing the annotators to "compete" with each other for the best label sequences.

Following the generative process described above, we can now define

$$p\left(\mathbf{y}_i^1,\ldots,\mathbf{y}_i^R|\mathbf{x}_i,\mathbf{z},\boldsymbol{\lambda}\right)=\prod_{r=1}^{R}\left\{p_{crf}\left(\mathbf{y}_i^r|\mathbf{x}_i,\boldsymbol{\lambda}\right)\right\}^{(z^r)}\left\{p_{rand}\left(\mathbf{y}_i^r|\mathbf{x}_i\right)\right\}^{(1-z^r)} \tag{6}$$

where we made use of the assumption that the annotators make their decisions independently of each other.

Including the vector $\mathbf{z}$ in our model as observed would yield the following expression for the likelihood

$$p(\mathcal{D},\mathbf{z}|\theta)=p(\mathbf{z}|\boldsymbol{\pi})\prod_{i=1}^{N}p\left(\mathbf{y}_i^1,\ldots,\mathbf{y}_i^R|\mathbf{x}_i,\mathbf{z},\boldsymbol{\lambda}\right) \tag{7}$$

where $\theta=\{\boldsymbol{\pi},\boldsymbol{\lambda}\}$ are the model parameters.

Since we do not actually observe $\mathbf{z}$, we must treat it as *latent* and marginalize over it by summing over all its possible values. The likelihood of our model then becomes

$$p(\mathcal{D}|\theta)=\sum_{\mathbf{z}}p(\mathbf{z}|\boldsymbol{\pi})\prod_{i=1}^{N}p\left(\mathbf{y}_i^1,\ldots,\mathbf{y}_i^R|\mathbf{x}_i,\mathbf{z},\boldsymbol{\lambda}\right). \tag{8}$$

The choice of explicitly including the reliability of the annotators (which we represent through the vector $\mathbf{z}$) as latent variables and marginalizing over it, contrasts with typical approaches in learning from multiple annotators (e.g. Raykar et al. 2009, 2010; Dredze et al. 2009; Yan et al. 2011), where the unobserved ground truth labels are treated as latent variables. Since these variables are not observed (i.e. latent), they must be marginalized over. For sequence labeling problems, this marginalization can be problematic due to the combinatorial explosion of possible label sequences over which we would have to marginalize. Instead, by explicitly handling the annotators reliabilities as latent variables this problem can be completely avoided.

Making use of Eqs. (3) and (6), the likelihood can be further simplified giving

$$p(\mathcal{D}|\theta)=\sum_{r=1}^{R}\pi_r\prod_{i=1}^{N}\left\{p_{crf}\left(\mathbf{y}_i^r|\mathbf{x}_i,\boldsymbol{\lambda}\right)\prod_{\substack{j=1\\j\neq r}}^{R}p_{rand}\left(\mathbf{y}_i^j|\mathbf{x}_i\right)\right\}. \tag{9}$$

The maximum likelihood estimator is then found by determining the parameters θ_{MLE} that maximize

$$\theta_{MLE}=\arg\max_{\theta}\ \ln\sum_{r=1}^{R}\pi_r\prod_{i=1}^{N}\left\{p_{crf}\left(\mathbf{y}_i^r|\mathbf{x}_i,\boldsymbol{\lambda}\right)\prod_{\substack{j=1\\j\neq r}}^{R}p_{rand}\left(\mathbf{y}_i^j|\mathbf{x}_i\right)\right\}. \tag{10}$$

3.4 EM algorithm

As with other latent variable models, we rely on the Expectation-Maximization algorithm (Dempster et al. 1977) to find a maximum likelihood parameters of the proposed model.

If we observed the complete dataset $\{\mathcal{D}, \mathbf{z}\}$ then the loglikelihood function would simply take the form $\ln p(\mathcal{D}, \mathbf{z}|\theta)$. Since we only have the "incomplete" dataset $\mathcal{D}$, our state of the knowledge of the values of the latent variable $\mathbf{z}$ (the reliabilities of the annotators) can be given by the posterior distribution $p(\mathbf{z}|\mathcal{D}, \theta)$. Therefore, instead of the complete-data loglikelihood, we consider its expected value under the posterior distribution of the latent variable $p(\mathbf{z}|\mathcal{D}, \theta)$, which corresponds to the E-step of the EM algorithm. Hence, in the E-step we use the current parameter values θ^{old} to find the posterior distribution of the latent variable $\mathbf{z}$. We then use this posterior distribution to find the expectation of the complete-data loglikelihood evaluated for some general parameter values θ. This expectation is given by

$$\begin{aligned}\mathbb{E}_{p(\mathbf{z}|\mathcal{D},\theta^{old})}\big\{\ln p(\mathcal{D}, \mathbf{z}|\theta)\big\} &= \sum_{\mathbf{z}} p\big(\mathbf{z}|\mathcal{D}, \theta^{old}\big) \ln p(\mathcal{D}, \mathbf{z}|\theta) \\ &= \sum_{\mathbf{z}} p\big(\mathbf{z}|\mathcal{D}, \theta^{old}\big) \ln \left\{ p(\mathbf{z}|\boldsymbol{\pi}) \prod_{i=1}^{N} p\big(\mathbf{y}_i^1, \ldots, \mathbf{y}_i^R|\mathbf{x}_i, \mathbf{z}, \boldsymbol{\lambda}\big) \right\}. \quad (11)\end{aligned}$$

The posterior distribution of the latent variable $\mathbf{z}$ can be estimated using the Bayes theorem, giving

$$\begin{aligned}\gamma\big(z^r\big) &= p\big(z^r = 1|\mathcal{D}, \theta^{old}\big) \\ &= \frac{p(z^r = 1|\boldsymbol{\pi}^{old})\ p(\mathcal{Y}^1, \ldots, \mathcal{Y}^R|\mathcal{X}, z^r = 1, \boldsymbol{\lambda}^{old})}{\sum_{j=1}^{R} p(z^j = 1|\boldsymbol{\pi}^{old})\ p(\mathcal{Y}^1, \ldots, \mathcal{Y}^R|\mathcal{X}, z^j = 1, \boldsymbol{\lambda}^{old})} \\ &= \frac{\pi_r^{old} \prod_{i=1}^{N} \{p_{crf}(\mathbf{y}_i^r|\mathbf{x}_i, \boldsymbol{\lambda}^{old}) \prod_{\substack{k=1 \\ k\neq r}}^{R} p_{rand}(\mathbf{y}_i^k|\mathbf{x}_i)\}}{\sum_{j=1}^{R} \pi_j^{old} \prod_{i=1}^{N} \{p_{crf}(\mathbf{y}_i^j|\mathbf{x}_i, \boldsymbol{\lambda}^{old}) \prod_{\substack{k=1 \\ k\neq j}}^{R} p_{rand}(\mathbf{y}_i^k|\mathbf{x}_i)\}}. \quad (12)\end{aligned}$$

As long as we are assuming a uniform model for $p_{rand}(\mathbf{y}_i^r|\mathbf{x}_i)$, this expression can be further simplified, giving

$$\gamma\big(z^r\big) = \frac{\pi_r^{old} \prod_{i=1}^{N} p_{crf}(\mathbf{y}_i^r|\mathbf{x}_i, \boldsymbol{\lambda}^{old})}{\sum_{j=1}^{R} \pi_j^{old} \prod_{i=1}^{N} p_{crf}(\mathbf{y}_i^j|\mathbf{x}_i, \boldsymbol{\lambda}^{old})}. \quad (13)$$

Making use of Eqs. (3), (6) and (11) the expected value of the complete-data loglikelihood becomes

$$\begin{aligned}&\mathbb{E}_{p(\mathbf{z}|\mathcal{D},\theta^{old})}\big\{\ln p(\mathcal{D}, \mathbf{z}|\theta)\big\} \\ &\quad = \sum_{r=1}^{R} \gamma\big(z^r\big) \left\{ \ln \pi_r + \sum_{i=1}^{N} \ln p_{crf}\big(\mathbf{y}_i^r|\mathbf{x}_i, \boldsymbol{\lambda}\big) + \sum_{\substack{j=1 \\ j\neq r}}^{R} \ln p_{rand}\big(\mathbf{y}_i^j|\mathbf{x}_i\big) \right\}. \quad (14)\end{aligned}$$

In the M-step of the EM algorithm we maximize this expectation with respect to the model parameters θ, obtaining new parameter values θ^{new}.

The EM algorithm can then be summarized as follows:

E-step Evaluate

$$\gamma\left(z^r\right) \propto \pi_r^{old} \prod_{i=1}^{N} p_{crf}\left(\mathbf{y}_i^r | \mathbf{x}_i, \boldsymbol{\lambda}^{old}\right). \tag{15}$$

M-step Estimate the new ground truth labels sequences $\widehat{\mathcal{Y}}^{new}$ and the new parameters $\theta^{new} = \{\boldsymbol{\pi}^{new}, \boldsymbol{\lambda}^{new}\}$ given by

$$\boldsymbol{\lambda}^{new} = \arg\max_{\boldsymbol{\lambda}} \sum_{i=1}^{N} \sum_{r=1}^{R} \gamma\left(z^r\right) \left\{\ln p_{crf}\left(\mathbf{y}_i^r | \mathbf{x}_i, \boldsymbol{\lambda}\right)\right\} \tag{16}$$

$$\widehat{\mathcal{Y}}^{new} = \arg\max_{\widehat{\mathcal{Y}}} p_{crf}\left(\widehat{\mathcal{Y}} | \mathcal{X}, \boldsymbol{\lambda}^{new}\right) \tag{17}$$

$$\pi_r^{new} = \frac{F1\text{-}measure_r}{\sum_{j=1}^{R} F1\text{-}measure_j} \tag{18}$$

where in Eq. (17) the new ground truth estimate is efficiently determined using the *Viterbi* algorithm,[3] and in Eq. (16) the new CRF model parameters $\boldsymbol{\lambda}^{new}$ are determined using limited-memory BFGS similarly to normal CRF training (Sutton and McCallum 2006). However, the loglikelihood function now includes a weighting factor: $\gamma(z^r)$. From this perspective, when learning from the label sequences of the different annotators, the proposed approach is weighting the latter by how much we expect them to be right, considering also how likely the other annotators are to be correct. If, for example, there are only two "good" annotators among a group of five, they will share the responsibility in "teaching" the CRF model.

The initialization of the EM algorithm can be simply done by assigning random values to the annotators reliabilities or by estimating the ground truth label sequences $\widehat{\mathcal{Y}}$ using majority voting. The algorithm stops when the expectation in equation 11 converges or when the updates to the annotators reliabilities fall below a given threshold.

3.5 Maximum-a-posteriori

Sometimes we know a priori that some annotators are better or more trustworthy than others. This knowledge can be incorporated in the model by imposing a Dirichlet prior with parameters $\{\alpha_1, \ldots, \alpha_R\}$ over the annotators reliabilities $\mathbf{z}$. Similarly, it is also useful to add a zero-mean Gaussian prior with σ^2 variance over the CRF parameters $\boldsymbol{\lambda}$ to enforce regularization (Sutton and McCallum 2006). The maximum-a-posteriori (MAP) estimator is found by determining

$$\theta_{MAP} = \arg\max_{\theta} \left\{\ln p(\mathcal{D}|\theta) + \ln p(\theta)\right\}. \tag{19}$$

An EM algorithm can then be derived in a similar fashion.

When no prior knowledge about the annotators reliabilities is given, the Dirichlet prior can also be used as non-informative prior with all parameters α_r equal. This prior would act as a regularization term preventing the model to overfit the data provided by a few annotators. The strength of the regularization would depend on the parameter α.

[3] Note that the ground truth estimate is required to compute the F1-scores of the annotators and estimate the multinomial parameters $\boldsymbol{\pi}$.

Table 1 Summary of CRF features

Features
Word identity features
Capitalization patterns
Numeric patterns
Other morphologic features (e.g. prefixes and suffixes)
Part-of-Speech tags
Bi-gram and tri-gram features
Window features (window size = 3)

4 Experiments

The proposed approach is evaluated in the field of Natural Language Processing (NLP) for the particular tasks of Named Entity Recognition (NER) and Noun Phrase (NP) chunking. NER refers to the Information Retrieval subtask of identifying and classifying atomic elements in text into predefined categories such as the names of persons, organizations, locations and others, while NP chunking consists of recognizing chunks of sentences that correspond to noun phrases. Because of their many applications these tasks are considered very important in the field of NLP and other related areas.

We make our experiments using two types of data: artificial data generated by simulating multiple annotators, and real data obtained using Amazon's Mechanical Turk (AMT). In both cases, the label sequences are represented using the traditional BIO (Begin, Inside, Outside) scheme as introduced by Ramshaw and Marcus (1995).

The proposed approach (henceforward referred to as "CRF-MA")[4] is compared with four baselines:

- MVseq: majority voting at sequence level (i.e., the label sequence with more votes wins);
- MVtoken: majority voting at token level (i.e., the BIO label with more votes for a given token wins);
- MVseg: this corresponds to a two-step majority voting performed over the BIO labels of the tokens. First, a majority voting is used for the segmentation process (i.e. to decide whether the token should be considered as part of a segment—a named entity for example), then a second majority voting is used to decide the labels of the segments identified (e.g. what type of named entity it is).
- CRF-CONC: a CRF using all the data from all annotators concatenated for training.

The proposed model is also compared with the two variants of multi-label model proposed in Dredze et al. (2009): MultiCRF and MultiCRF-MAX. The latter differs from the former by training the CRF on the most likely (maximum) label instead of training on the (fuzzy) probabilistic labels (kindly see Dredze et al. (2009) for the details).

As an upper-bound, we also show the results of a CRF trained on ground truth (gold) data. We refer to this as "CRF-GOLD".

For all the experiments a simple set of features that is typical in NLP tasks was used. The features used are summarized in Table 1. In CRF-MA, the EM algorithm was initialized with

[4]Datasets available at: http://amilab.dei.uc.pt/fmpr/crf-ma-datasets.tar.gz. Source code available at: http://amilab.dei.uc.pt/fmpr/ma-crf.tar.gz.

token-level majority voting (MVtoken). The MultiCRF model was initialized with uniform label priors. All the results are reported using (strict) phrase-level F1-score.

During our experiments we found that using the square of the $F1$-*measure* when computing π^r gives the best results. This has the effect of emphasizing the differences between the reliabilities of the different annotators, and consequently their respective importances when learning the CRF model from the data. Hence, we use this version in all our experiments.

4.1 Artificial data

4.1.1 Named entity recognition

There are a few publicly available "golden" datasets for NER such as the 2003 CONLL English NER task dataset (Sang and Meulder 2003), which is a common benchmark for sequence labeling tasks in the NLP community. Using the 2003 CONLL English NER data we obtained a train set and a test set of 14987 and 3466 sentences respectively.

Since the 2003 CONLL shared NER dataset does not contain labels from multiple annotators, these are simulated for different reliabilities using the following method. First a CRF is trained for the complete training set. Then, random Gaussian noise (with zero mean and σ^2 variance) is applied to the CRF parameters and the modified CRF model is used to determine new sequences of labels for the training set texts. These label sequences differ more or less from the ground truth depending on σ^2. By repeating this procedure many times we can simulate multiple annotators with different levels of reliability.

An alternative approach would take the ground truth dataset and randomly flip the labels of each token with uniform probability p. However, this would result in simulated annotators that are inconsistent throughout the dataset, by labeling the data with a certain level of randomness. We believe that the scenario where the annotators are consistent but might not be as good as an "expert" is more realistic and challenging, and thus more interesting to investigate. Therefore we give preference to the CRF-based method in most of our experiments with artificial data. Nonetheless, we also make experiments using this alternative method of label-flipping to simulate annotators for the NP chunking task.

Using the CRF-based method described above, we simulated 5 artificial annotators with $\sigma^2 = [0.005, 0.05, 0.05, 0.1, 0.1]$. This choice of values intends to reproduce a scenario where there is a "good", two "bad" and two "average" annotators. The proposed approach (CRF-MA) and the four baselines were then evaluated against the test set. This process was repeated 30 times and the average results are presented in Table 2. We also report the results obtained on the training set. Note that, unlike for "typical" supervised learning tasks, in our case the F1 of the training set is important because it represents the estimation of the "unobserved" ground truth from the opinions of multiple annotators with different levels of expertise.

The results in Table 2 indicate that CRF-MA outperforms the four proposed baselines in both the train set and test set. In order to assess the statistical significance of this result, a paired t-test was used to compare the mean F1-score of CRF-MA in the test set against the MVseq, MVtoken, MVseg and CRF-CONC baselines. The obtained *p-values* were 4×10^{-25}, 7×10^{-10}, 4×2^{-8} and 1×10^{-14} respectively, which indicates that the differences are all highly significant.

Regarding the MultiCRF model, we can see that, at best, it performs almost as good as MVtoken. Not surprisingly, the "MAX" version of MultiCRF performs better than the standard version. This behavior is expected since the "hard" labels obtained from majority voting also perform better than the "soft" label effect obtained in CRF-CONC. Nonetheless,

Table 2 Results for the NER task with 5 simulated annotators (with $\sigma^2 = [0.005, 0.05, 0.05, 0.1, 0.1]$) with repeated labeling

Method	Train set			Test set		
	Precision	Recall	F1	Precision	Recall	F1
MVseq	24.1 %	50.5 %	32.6 ± 2.0 %	47.3 %	30.9 %	37.3 ± 3.1 %
MVtoken	56.0 %	69.1 %	61.8 ± 4.1 %	62.4 %	62.3 %	62.3 ± 3.4 %
MVseg	52.5 %	65.0 %	58.0 ± 6.9 %	60.6 %	57.1 %	58.7 ± 7.1 %
CRF-CONC	47.9 %	49.6 %	48.4 ± 8.8 %	47.8 %	47.1 %	47.1 ± 8.1 %
MultiCRF	39.8 %	22.6 %	28.7 ± 3.8 %	40.0 %	15.4 %	22.1 ± 5.0 %
MultiCRF-MAX	55.0 %	66.7 %	60.2 ± 4.1 %	63.2 %	58.4 %	60.5 ± 3.6 %
CRF-MA	72.9 %	81.7 %	77.0 ± 3.9 %	72.5 %	67.7 %	70.1 ± 2.5 %
CRF-GOLD	99.7 %	99.9 %	99.8 %	86.2 %	87.8 %	87.0 %

Table 3 Results for the NER task with 5 simulated annotators (with $\sigma^2 = [0.005, 0.05, 0.05, 0.1, 0.1]$) without repeated labeling

Method	Train set			Test set		
	Precision	Recall	F1	Precision	Recall	F1
CRF-CONC	52.1 %	56.5 %	54.0 ± 7.3 %	53.9 %	51.7 %	52.6 ± 6.4 %
CRF-MA	63.8 %	71.1 %	67.2 ± 1.7 %	65.7 %	62.7 %	64.2 ± 1.6 %
CRF-GOLD	99.7 %	99.9 %	99.8 %	86.2 %	87.8 %	87.0 %

neither version of MultiCRF performs as well as MA-CRF (test set *p-values* are 1×10^{-26} and 1×10^{-11} for the MultiCRF and MultiCRF-MAX respectively).

In order to empirically show that the proposed approach does not rely on repeated labeling (i.e., multiple annotators labeling the same data instances), the same "golden" NER dataset was split into 5 subsets, and for each subset an annotator was simulated with a different level of reliability σ^2 (namely $\sigma^2 = [0.005, 0.05, 0.05, 0.1, 0.1]$) according to the CRF-based procedure described above. This process was repeated 30 times and the average results for the provided test set can be found in Table 3. Since there was no repeated labeling, the majority voting baselines, as well as the multi-label models (MultiCRF and MultiCRF-MAX), did not apply. The obtained results indicate that, in a scenario without any repeated labeling, the proposed approach (CRF-MA) still outperforms the CRF-CONC baseline. The statistical significance of the difference between the F1-scores of the two methods in the test set was evaluated using a paired t-test, indicating that the difference of the means is highly significant ($p\text{-}value = 1.47 \times 10^{-11}$).

The comparison of both experiments (i.e. with and without repeated labeling) indicates that, in this setting, having less repeated labeling hurts the performance of CRF-MA. Since this model differentiates between annotators with different levels of expertise, its performance is best when the more reliable ones have annotated more sequences, which is more likely to happen with more repeated labeling. Naturally, the opposite occurs with CRF-CONC. Since in this setting the less reliable annotators dominate, more repeated labeling translates in even more predominance of lower quality annotations, which affects the performance of CRF-CONC.

Table 4 Results for the NP chunking task with 5 simulated annotators (with $p = [0.01, 0.1, 0.3, 0.5, 0.7]$) with repeated labeling

Method	Train set			Test set		
	Precision	Recall	F1	Precision	Recall	F1
MVseq	50.6 %	55.6 %	53.0 ± 0.4 %	66.1 %	63.1 %	64.6 ± 2.4 %
MVtoken	83.6 %	76.1 %	79.7 ± 0.2 %	83.3 %	86.9 %	85.0 ± 0.7 %
CRF-CONC	84.3 %	84.7 %	84.5 ± 1.8 %	83.8 %	82.9 %	83.3 ± 1.9 %
MultiCRF	76.6 %	65.6 %	70.7 ± 0.4 %	75.6 %	64.9 %	69.8 ± 0.4 %
MultiCRF-MAX	83.6 %	81.3 %	82.5 ± 1.0 %	81.2 %	79.0 %	80.1 ± 1.0 %
CRF-MA	92.0 %	91.8 %	91.9 ± 1.9 %	89.7 %	89.7 %	89.7 ± 0.8 %
CRF-GOLD	99.9 %	99.9 %	99.9 %	95.9 %	91.1 %	91.0 %

4.1.2 Noun phrase chunking

For the NP chunking task, the 2003 CONLL English NER dataset was also used. Besides named entities, this dataset also provides part-of-speech tags and syntactic tags (i.e. noun phrases, verbal phrases, prepositional phrases, etc.). The latter were used to generate a train and a test set for NP chunking with the same sizes of the corresponding NER datasets.

In order to simulate multiple annotators in the NP chunking data, the alternative method of randomly flipping the label of each token with uniform probability p was used. Since for this task there are only two possible labels for each token (part of a noun phrase or not part of a noun phrase)[5] it is trivial to simulate multiple annotators by randomly flipping labels. This annotator simulation process reproduces situations where there is noise in the labels of the annotators. Using this method we simulated 5 annotators with label flip probabilities $p = [0.01, 0.1, 0.3, 0.5, 0.7]$. This process was repeated 30 times and the average results are presented in Table 4. Differently to NER, NP chunking is only a segmentation task, therefore the results for the MVseg baseline would be equal to the results for MVtoken. The experimental evidence shows that the proposed approach (CRF-MA) achieves a higher F1-score than the MVseq, MVtoken and CRF-CONC baselines. The statistical significance of the difference between the test set F1-scores of CRF-MA and all these three baselines (MVseq, MVtoken and CRF-CONC) was evaluated using a paired t-test, yielding *p-values* of 2×10^{-30}, 7×10^{-22} and 2×10^{-16} respectively. As with the NER task, the CRF-MA model also outperforms the MultiCRF and MultiCRF-MAX approaches (test set *p-values* are 6×10^{-32} and 2×10^{-21} respectively).

4.2 Real data

The use of *Crowdsourcing* platforms to annotate sequences is currently a very active research topic (Laws et al. 2011), with many different solutions being proposed to improve both the annotation and the learning processes at various levels like, for example, by evaluating annotators through the use of an expert (Voyer et al. 2010), by using a better annotation interface (Lawson et al. 2010), or by learning from partially annotated sequences thus reducing annotation costs (Fernandes and Brefeld 2011).

[5]In fact, since a BIO decomposition is being used, there are three possible labels: B-NP, I-NP and O, and these labels are the ones that were used in the random flipping process.

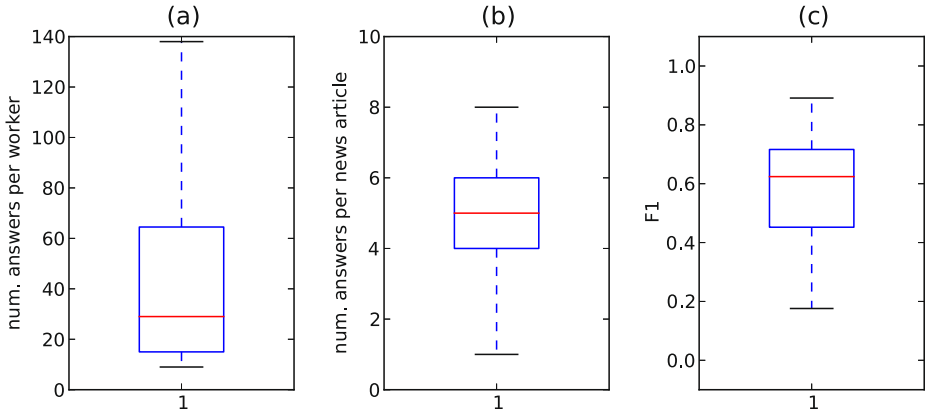

Fig. 2 Boxplots for (**a**) the number of answers per AMT worker, (**b**) the number of answers per news article, and (**c**) the F1-scores of the annotators

With the purpose of obtaining real data from multiple annotators, we put 400 news articles from the 2003 CONLL shared NER task (for which we had ground truth) on Amazon's *Mechanical Turk* for workers to label. In this experiment, the workers were then required to identify the named entities in the sentence and classify them as persons, locations, organizations or miscellaneous. Together with the named entity definition and the categories description, the workers were also provided with two exemplifying sentences. Workers with just a couple of answers were considered uninterested in the task and their answers were discarded, giving a total of 47 valid annotators. The average number of annotators per news article was 4.93, and each annotator labelled an average of 42 news articles (see Figs. 2a and 2b). In order to assess the "quality" of the annotators, we measured their F1-scores against the ground truth. Figure 2c shows a boxplot of the F1-scores obtained. It is interesting to notice that the quality of the AMT workers really varies enormously, with the lowest F1-score being 17.60 % (a very unreliable annotator), while the highest F1-score is 89.11 % (arguably almost an expert).

As with the experiments with simulated annotators, the different approaches are evaluated in the provided test set, as well as in the ground truth labels for those 400 news articles. The obtained results are presented in Table 5. These results indicate that the proposed approach is better at uncovering the ground truth than all the other approaches tested. This in turn results in a better performance on the test set. Furthermore, the RMSE obtained between the true F1-scores of the annotators (measured against the actual ground truth) and their estimated F1-scores according to the CRF-MA approach (measured against the estimated ground truth) was 8.61 %, meaning that the reliability of the annotators is being approximated quite well. These results also indicate that *crowdsourcing* presents an interesting alternative solution for obtaining labeled data that could be used for training a NER system.

In order to evaluate the impact of repeated labeling, a random subsampling of the AMT data was performed. This experiment will allow us to reproduce a situation where each article is only labeled by one annotator, thus representing the minimum cost attainable with AMT (with the same price per task). For each of the 400 news articles, a single annotator was picked at random from the set of workers who labeled that article. This process was repeated 30 times to produce 30 subsampled datasets. The average precision, recall and

Table 5 Results for the NER task using real data obtained from Amazon's *Mechanical Turk*

Method	Train set			Test set		
	Precision	Recall	F1	Precision	Recall	F1
MVseq	79.0 %	55.2 %	65.0 %	44.3 %	81.0 %	57.3 %
MVtoken	79.0 %	54.2 %	64.3 %	45.5 %	80.9 %	58.2 %
MVseg	83.7 %	51.9 %	64.1 %	46.3 %	82.9 %	59.4 %
CRF-CONC	86.8 %	58.4 %	69.8 %	40.2 %	86.0 %	54.8 %
MultiCRF	67.8 %	15.4 %	25.1 %	74.8 %	3.7 %	7.0 %
MultiCRF-MAX	79.5 %	51.9 %	62.8 %	84.1 %	37.1 %	51.5 %
CRF-MA	86.0 %	65.6 %	74.4 %	49.4 %	85.6 %	62.6 %
CRF-GOLD	99.2 %	99.4 %	99.3 %	79.1 %	80.4 %	74.8 %

Table 6 Results for the NER task using data from Amazon's *Mechanical Turk* without repeated labelling (subsampled data from the original dataset)

Method	Train set			Test set		
	Precision	Recall	F1	Precision	Recall	F1
CRF-CONC	71.1 %	42.8 %	53.1 ± 10.5 %	35.9 %	70.1 %	47.2 ± 8.7 %
CRF-MA	76.2 %	54.2 %	63.3 ± 1.6 %	46.0 %	78.2 %	57.9 ± 1.8 %
CRF-GOLD	99.2 %	99.4 %	99.3 %	79.1 %	80.4 %	74.8 %

F1-scores of the different methods are shown in Table 6. Notice that, since there is no repeated labeling, both the majority voting baselines and the multi-label models (MultiCRF and MultiCRF-MAX) do not apply. The obtained results show that CRF-MA also outperforms CRF-CONC in this setting ($p\text{-}value = 3.56 \times 10^{-7}$). Interestingly, when compared to the results in Table 5, this experiment also shows how much could be gained by repeated labeling, thus providing a perspective on the trade-off between repeated labeling and cost.

5 Conclusion

This paper presented a probabilistic approach for sequence labeling using CRFs with data from multiple annotators which relies on a latent variable model where the reliability of the annotators are handled as latent variables. The EM algorithm is then used to find maximum likelihood estimates for the CRF model parameters, the reliability of the annotators and the ground truth label sequences. The proposed approach is empirically shown to significantly outperform traditional approaches, such as majority voting and using the labeled data from all the annotators concatenated for training, even in situations of high levels of noise in the labels of the annotators and when the less "trustworthy" annotators dominate. This approach also has the advantage of not requiring the repeated labeling of the same input sequences by the different annotators (unlike majority voting, for example). Although we presented a formulation using CRFs, it could be easily modified to work with other sequence labeling models such as HMMs.

Future work intends to explore dependencies of the reliabilities of the annotators on the input sequences they are labeling, which can be challenging due to the high dimensionality of the feature space, and the inclusion of a Dirichlet prior over the qualities of the annotators.

Furthermore, the extension of the proposed model to an active learning setting will also be considered. Since the annotators reliabilities are being estimated by the EM algorithm, this information can be used to, for example, decide who are the most trustworthy annotators. Requesting new labels from those annotators will eventually improve the models performance and reduce annotation cost.

Acknowledgements The Fundação para a Ciência e Tecnologia (FCT) is gratefully acknowledged for founding this work with the grants SFRH/BD/78396/2011 and PTDC/EIA-EIA/115014/2009 (CROWDS). We would also like to thank Mark Dredze and Partha Talukdar for kindly providing the code for their implementation of the MultiCRF model (Dredze et al. 2009).

References

Allen, J., & Salzberg, S. (2005). JIGSAW: integration of multiple sources of evidence for gene prediction. *Bioinformatics*, *21*(18), 3596–3603.

Allen, J., Pertea, M., & Salzberg, S. (2004). Computational gene prediction using multiple sources of evidence. *Genome Research*, *14*(1), 142–148.

Bellare, K., & Mccallum, A. (2007). Learning extractors from unlabeled text using relevant databases. In *Sixth international workshop on information integration on the web*.

Callison-Burch, C., & Dredze, M. (2010). Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with amazon's mechanical turk* (pp. 1–12).

Dawid, A. P., & Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C. Applied Statistics*, *28*(1), 20–28.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, *39*(1), 1–38.

Donmez, P., & Carbonell, J. (2008). Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on information and knowledge management* (pp. 619–628).

Donmez, P., Schneider, J., & Carbonell, J. (2010). A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *Proceedings of the SIAM international conference on data mining* (pp. 826–837).

Dredze, M., Talukdar, P., & Crammer, K. (2009). Sequence learning from data with multiple labels. In *ECML-PKDD 2009 workshop on learning from multi-label data*.

Fernandes, E., & Brefeld, U. (2011). Learning from partially annotated sequences. In *Proceedings of the 2011 European conference on machine learning and knowledge discovery in databases* (pp. 407–422).

Groot, P., Birlutiu, A., & Heskes, T. (2011). Learning from multiple annotators with Gaussian processes. In *Proceedings of the 21st international conference on artificial neural networks* (Vol. 6792, pp. 159–164).

Howe, J. (2008). *Crowdsourcing: why the power of the crowd is driving the future of business* (1st edn.). New York: Crown Publishing Group.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th international conference on machine learning* (pp. 282–289).

Laws, F., Scheible, C., & Schütze, M. (2011). Active learning with amazon mechanical turk. In *Proceedings of the conference on empirical methods in natural language processing*. Stroudsburg: Association for Computational Linguistics (pp. 1546–1556).

Lawson, N., Eustice, K., Perkowitz, M., & Yetisgen-Yildiz, M. (2010). Annotating large email datasets for named entity recognition with mechanical turk. In *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with amazon's mechanical turk*. Stroudsburg: Association for Computational Linguistics (pp. 71–79).

Liu, D., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, *45*, 503–528.

Novotney, S., & Callison-Burch, C. (2010). Cheap, fast and good enough: automatic speech recognition with non-expert transcription. In *Human language technologies, HLT '10*. Stroudsburg: Association for Computational Linguistics (pp. 207–215).

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE* (pp. 257–286).

Ramshaw, L., & Marcus, M. (1995). Text chunking using transformation-based learning. In *Proceedings of the third workshop on very large corpora*. Stroudsburg: Association for Computational Linguistics (pp. 82–94).

Raykar, V., Yu, S., Zhao, L., Jerebko, A., Florin, C., Valadez, G., Bogoni, L., & Moy, L. (2009). Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th international conference on machine learning* (pp. 889–896).

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

Sang, E., & Meulder, F. D. (2003). Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the 7th conference on natural language learning at HLT-NAACL* (Vol. 4, pp. 142–147).

Sheng, V., Provost, F., & Ipeirotis, P. (2008). Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 614–622).

Smyth, P., Fayyad, U., Burl, M., Perona, P., & Baldi, P. (1995). Inferring ground truth from subjective labelling of venus images. *Advances in Neural Information Processing Systems*, 1085–1092.

Snow, R., O'Connor, B., Jurafsky, D., & Ng, A. (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 254–263).

Surowiecki, J. (2004). *The wisdom of crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations*. New York: Doubleday.

Sutton, C., & McCallum, A. (2006). *Introduction to conditional random fields for relational learning*. Cambridge: MIT Press.

Voyer, R., Nygaard, V., Fitzgerald, W., & Copperman, H. (2010). A hybrid model for annotating named entity training corpora. In *Proceedings of the fourth linguistic annotation workshop*. Stroudsburg: Association for Computational Linguistics (pp. 243–246).

Wu, O., Hu, W., & Gao, J. (2011). Learning to rank under multiple annotators. In *Proceedings of the 22nd international joint conference on artificial intelligence* (pp. 1571–1576).

Yan, Y., Rosales, R., Fung, G., Schmidt, M., Valadez, G., Bogoni, L., Moy, L., & Dy, J. (2010). Modeling annotator expertise: learning when everybody knows a bit of something. *Journal of Machine Learning Research*, *9*, 932–939.

Yan, Y., Rosales, R., Fung, G., & Dy, J. (2011). Active learning from crowds. In *Proceedings of the 28th international conference on machine learning* (pp. 1161–1168).