Low Resource Sequence Tagging with Weak Labels

Edwin Simpson, Jonas Pfeiffer, Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP)
Department of Computer Science, Technische Universität Darmstadt.
{simpson, pfeiffer, gurevych}@ukp.informatik.tu-darmstadt.de

Abstract

Current methods for sequence tagging depend on large quantities of domain-specific training data, limiting their use in new, user-defined tasks with few or no annotations. While crowdsourcing can be a cheap source of labels, it often introduces errors that degrade the performance of models trained on such crowdsourced data. Another solution is to use transfer learning to tackle low resource sequence labelling, but current approaches rely heavily on similar high resource datasets in different languages. In this paper, we propose a domain adaptation method using Bayesian sequence combination to exploit pre-trained models and unreliable crowdsourced data that does not require high resource data in a different language. Our method boosts performance by learning the relationship between each labeller and the target task and trains a sequence labeller on the target domain with little or no gold-standard data. We apply our approach to labelling diagnostic classes in medical and educational case studies, showing that the model achieves strong performance though zero-shot transfer learning and is more effective than alternative ensemble methods. Using NER and information extraction tasks, we show how our approach can train a model directly from crowdsourced labels, outperforming pipeline approaches that first aggregate the crowdsourced data, then train on the aggregated labels.

1 Introduction

NLP systems are often needed for bespoke tasks, such as extracting a particular type of information in the form of text spans from documents in a specific domain. The lack of training data for each task motivates the reuse of pre-trained models and crowdsourcing. Current methods for crowdsourcing often result in errors, and agreement even between expert annotators is far from perfect for many NLP tasks. This leads to the following questions: How can existing NLP models be adapted to new domains or tasks with minimal training data? How can we make the most of small amounts of unreliable, crowdsourced annotations? In this paper, we focus on the generic task of sequence labelling, which has broad applications to information extraction, named entity recognition, question answering, and other span annotation tasks, and consider how to learn a sequence labeller using pre-trained models and noisy crowdsourced data.

Transfer learning can be used to adapt pre-trained models to a target task, however, typical methods require either labelled training examples or large unlabelled datasets in the target domain, or parallel examples in each domain (Ruder 2019). Fine-tuning techniques can also take several hours on a GPU, which renders these methods unsuitable for adapting rapidly to a user's request or in realtime. Instead, we need a solution that can perform transfer learning quickly with small amounts of noisy annotations.

When faced with a new sequence labelling task for which there is very little training data, one possibility is to apply several pre-trained models from different domains to the new task and combine their predictions. This forms an ensemble of weak labellers that often performs better than any available individual labeller. Various methods have been introduced to combine sequence labels from multiple annotators, including a recent method, *Bayesian sequence combination (BSC)* (Simpson and Gurevych 2019), which models sequential dependencies between labels. The Bayesian approach reduces over-fitting and handles uncertainty in the model when learning from sparse, unreliable data, such as that obtained by current crowdsourcing methods.

We propose to combine multiple sequence labellers using BSC to learn the correlations between a model's labels and the target variable. While the idea is closely related to ensemble learning, typical ensemble methods such as boosting, bagging or random forests are concerned with generating base classifiers, whereas we wish to use existing models. BSC uses variational Bayes (VB) to approximately infer the target labels in a Bayesian manner without the computational cost of sampling methods (Simpson and Gurevych 2019). Here, we propose a technique that extends the VB algorithm to learn a sequence labeller as part of an ensemble with human annotators, in order to reduce the workload of a crowd. Our variational technique lets us train the sequence labeller when gold labels are unavailable, while still providing an approximate Bayesian treatment to the BSC model.

Our key contributions are: (1) a transfer learning method for pre-trained sequence labellers using Bayesian sequence combination; (2) a variational inference technique that trains

models on a target domain with sparse, noisy annotations. We evaluate the approach empirically using two real-world datasets, showing that our transfer learning and training approaches improve sequence labelling in new domains with very few labels or tasks with only noisy, crowdsourced data. We make all of our code and data publicly available[1].

2 Background and Related Work

Transfer learning with small data. Recent advances in transfer learning leverage transformer-based models trained on large amounts of data (Vaswani et al. 2017; Devlin et al. 2019), which continuously outperform GLUE benchmarks (Wang et al. 2018), underlining the effectiveness of transfer learning in NLP. Transfer learning for sequence labelling has primarily been realised by fine-tuning: Yang, Salakhutdinov, and Cohen (2017) fine-tune different subsets of parameters in a source model for different tasks; Peters et al. (2017) fine-tune a language model to transfer the contextual representation to sequence labelling tasks. While fine-tuning works well for tasks with sufficient data, it is considerably more difficult for small datasets. Further, for many tasks it is necessary to fine-tune the entire model, not just the final layer (Peters, Ruder, and Smith 2019), which increases computational overheads and causes updates to take several hours. In this paper, we consider scenarios where these data and computational demands make fine-tuning impractical.

In the low-resource setting, most work has focused on the transfer of sequence labelling models from a high resource domain of one language to a low resource domain of another language (Feng et al. 2018; Mayhew, Tsai, and Roth 2017). Lin et al. (2018) combine the different approaches of multi-task learning and cross-lingual training to leverage the signals for a low-resource task. A major challenge for multi-task learning in low resource settings is that the models tend to overfit the majority task when the data is unbalanced. Another direction is zero-shot transfer learning: for instance, Rei and Søgaard (2018) use information from a sentence classification task together with weak attention to infer token level labels, hence they do not require labelled data at the token level. The methods we propose here do not rely on cross-lingual resources or a hierarchical structure, such as the relationship between sentences and tokens.

While most work in transfer learning has either leveraged labelled data for the target task for fine-tuning, or, in the low-resource setting, relied on cross-lingual high-resource data, there has been little work on single language, low-resource transfer learning. Zhou et al. (2019) tackle this problem through adversarial training to mitigate overfitting on the target domain. However, in contrast to our work, all approaches introduce a proprietary model architecture which is designed for the specific task. This introduces the additional constraint that the capacity of the source model needs to be able to adapt to the new domain, which excludes rule-based models, for example. While our approach can fine-tune a parametric model, this model can be treated as a black box with a simple training and prediction interface. Our method can integrate sequence labelling models of arbitrary nature into an ensemble, using Bayesian combination methods to address their varying informativeness, making our approach universally applicable.

Ensemble methods. Rahimi, Li, and Cohn (2019) propose to use an ensemble technique to transfer a large set of named entity recognition models trained on different languages to a low-resource target language. The ensembling approach avoids the need to hand-pick suitable models for transfer, as it learns the relevance of source models to the target domain in both a supervised and unsupervised manner. In contrast to our method, their approach does not consider sequential dependencies between labels, so requires an additional span-level aggregation step to resolve labelling inconsistencies. For example, if spans are annotated with inside-outside-beginning (IOB2) encoding, an I tag may not immediately follow an O tag, as a B tag is required at the start of the span. Furthermore, their work performs cross-lingual transfer learning with a large ensemble, whereas our approach exploits a small number of models trained on different domains in the same language. Rehbein and Ruppenhofer (2017) also combine automatic sequence labellers using MACE (Hovy et al. 2013), but again, they do not model sequential dependencies between labels nor train sequence labellers from noisy labels.

The method used by Rahimi, Li, and Cohn (2019) to combine different models is based on IBCC (Kim and Ghahramani 2012), which was originally used for classifier combination, then for aggregating crowdsourced data (Simpson et al. 2013). IBCC is based on earlier work by Dawid and Skene (1979), which has been shown to outperform various other weighted voting methods (Sheshadri and Lease 2013; Simpson et al. 2013). Recent methods extend this type of model to sequence labels (Nguyen et al. 2017), including BSC (Simpson and Gurevych 2019), which was shown to outperform both IBCC and MACE at sequence labelling tasks. Unlike this paper, Simpson and Gurevych (2019) did not apply BSC to transfer learning nor integrate the training process for a sequence labeller. However, previous work demonstrates the benefits of approximate inference using VB over techniques such as maximum likelihood expectation maximisation (Hovy et al. 2013; Simpson et al. 2013; Simpson and Gurevych 2019), as VB accounts for uncertainty in model parameters, for example, when there is insufficient data to learn the parameters with high confidence.

Several previous works have proposed methods for learning a sequence tagger from crowdsourced data. Plank, Hovy, and Søgaard (2014) modify the loss function of the sequence tagger to account for disagreement between annotators. Nguyen et al. (2017) and Rodrigues and Pereira (2018) train neural sequence taggers directly on crowdsourced data by adding a *crowd layer* to model the reliability of each annotator. However, these approaches did not outperform probabilistic models in their experiments on sequence labelling tasks. Unlike these approaches, we do not need to adapt the sequence tagger itself to learn from noisy labels.

Another approach by Albarqouni et al. (2016) integrates a CNN classifier for image classification into an aggrega-

[1] https://github.com/UKPLab/arxiv2018-bayesian-ensembles

tion method based on expectation maximization (EM), while Yang et al. (2018) adapt a Bayesian neural network so that it can be trained concurrently with an annotator model, also using EM. Our approach also uses EM to train a neural network (or any off-the-shelf tagger), but, in contrast to previous work, we do not assume that the tagger's predictions are reliable, instead treating it as another member of the ensemble and accounting for its unreliability.

3 Combining Sequence Labellers

We propose to apply multiple pre-trained sequence labellers or noisy crowd annotators to a target domain using Bayesian sequence combination (*BSC*) (Simpson and Gurevych 2019). We use BSC to combine multiple automated taggers or human annotators – collectively referred to as *annotators* – by treating each annotator as a weak labeller in the target domain. Combining multiple such weak annotators allows us to estimate the target labels with higher accuracy, following the principles of ensemble methods: uncorrelated errors made by individual annotators tend to cancel each other out so that the ensemble is more accurate than the average individual (Brown et al. 2005).

The simplest way to combine multiple annotators is to take a majority vote: for each token, choose the label that was selected by most annotators. However, this ignores the fact that some annotators may be better at choosing the correct label than others. Therefore, models such as BSC learn an *annotator model* for each individual annotator that describes the relationship between their labels and the target labels. When we transfer weak sequence taggers from one domain to another, the annotator model puts more weight on annotators that are more accurate in the target domain and effectively ignores those that do not correlate with the target task at all. Similarly, when combining crowd workers, the annotator model will ignore spammers and give accurate labellers high weights. Combining existing taggers using BSC allows us to exploit the information they provide without modifying or re-training the taggers themselves.

The BSC-seq Model The goal of BSC is to infer the sequence of target labels $\boldsymbol{t}_n = \{t_{n,1}, .., t_{n,L_n}\}$ for each document n in a set of N documents, where L_n is the length of the nth document. Each target label takes a value from $\{1, .., J\}$. For document n, we observe a sequence of tokens $\boldsymbol{x}_n$ and a set of annotations, $\boldsymbol{c}_n^{(k)} = \{c_{n,1}^{(k)}, .., c_{n,L_n}^{(k)}\}$, from each annotator, k, that has labelled document n. The annotations take values in $\{1, .., J_k\}$, where J_k is the number of class labels that annotator k can assign, which may differ from the set of target labels.

Similar to a *hidden Markov model (HMM)*, BSC assumes that the probability of a target label $t_{n,\tau}$ depends on its predecessor: $p(t_{n,\tau} = i | t_{n,\tau-1} = j, \boldsymbol{T}) = T_{j,i}$, where $\boldsymbol{T}$ is a matrix of transition probabilities between target labels. BSC also assumes that the probability of observing a token $x_{n,\tau}$ is given by a categorical distribution with a probability vector $\boldsymbol{\rho}_j$, which depends on the target label $t_{n,\tau} = j$.

Similarly, each annotation, $c_{n,\tau}^{(k)}$, depends on the target label, and BSC can use different annotator models to model this likelihood. Here, we use *BSC-seq*, which models sequential dependencies between annotations as follows:

$$p(c_{n,\tau}^{(k)} = l | c_{n,\tau-1} = \iota, t_{n,\tau} = j, \boldsymbol{\pi}^{(k)}) = \pi_{j,\iota,l}^{(k)}. \quad (1)$$

The parameters $\boldsymbol{\pi}^{(k)}$ of the annotator models and target labels $\boldsymbol{t}_n$ can be inferred using VB as in Simpson and Gurevych (2019). We now extend the VB procedure to train a sequence tagger from weak labels in the target domain.

Tuning Black-box Sequence Taggers using Weak Labels We now address the problem of training or fine-tuning a model for the target domain given only weak labels. By including a model tuned to the target domain, we aim to boost the performance of the complete ensemble. Ensembles of human annotators can benefit from including automated taggers to provide additional, cheap sources of labels across large numbers of documents. However, the scenarios we consider lack reliable data for training on the target domain. We therefore propose *variational combined supervision (VCS)*, a method for training a tagger given weak labels from an ensemble of annotators, which allows us to use a tagger without modification, treating it as a black box. The approach integrates the tagger into BSC as follows.

We choose one annotator, s, to be trained using variational combined supervision. We modify the BSC-seq model so that the labels $\boldsymbol{c}^{(s)}$ produced by s are now latent, rather than observed variables. The posterior distribution and its variational approximation are given as follows:

$$p(\boldsymbol{t}, \boldsymbol{\pi}, \boldsymbol{T}, \boldsymbol{\rho}, \boldsymbol{c}^{(s)} | \boldsymbol{c}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\kappa})$$
$$\approx \prod_{k=1}^{K} q(\boldsymbol{\pi}^{(k)}) \prod_{j=1}^{J} \{q(\boldsymbol{T}_j) q(\boldsymbol{\rho}_j)\} \prod_{n=1}^{N} q(\boldsymbol{t}_n) q(\boldsymbol{c}^{(s)}), \quad (2)$$

where $\boldsymbol{c}$ contains the labels from all annotators except s, $\boldsymbol{x}$ is the set of token sequences, and $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\kappa}$ are hyperparameters. For detailed definitions of the terms in Equation 2, we refer the reader to Simpson and Gurevych (2019). The variational approximation in Equation 2 differs from that of standard BSC-seq in the inclusion of $q(c^{(s)})$, defined as:

$$\ln q(\boldsymbol{c}^{(s)}) = \mathbb{E}_{\boldsymbol{t}, \boldsymbol{\pi}^{(s)}}[\ln p(\boldsymbol{c}^{(s)} | \boldsymbol{x}, \boldsymbol{t}, \boldsymbol{\pi}^{(s)})] \quad (3)$$

We do not need to know the form of this distribution, which is defined by sequence tagger s, as we interface with s to estimate the required terms through two functions: $train(\boldsymbol{x}, \hat{\boldsymbol{t}})$ and $predict(\boldsymbol{x}_n)$. We assume that training s either finds the optimal internal parameters $\hat{\boldsymbol{\theta}}$ of s that locally maximise the likelihood of $\hat{\boldsymbol{t}}$ given the inputs $\boldsymbol{x}$, or marginalises $\boldsymbol{\theta}$. The $predict(\boldsymbol{x}_n)$ function then returns the approximate posterior probabilities of each label as a vector, $\hat{\boldsymbol{c}}_{n,\tau}^{(s)} = \left[\hat{c}_{n,\tau,1}^{(s)}, ..., \hat{c}_{n,\tau,J_s}^{(s)}\right]$, with entries $\hat{c}_{n,\tau,j}^{(s)} = \mathbb{E}_{\boldsymbol{\pi}^{(s)}}[p(c_{n,\tau}^{(s)} = j | \hat{\boldsymbol{t}}, \boldsymbol{x}_n, \boldsymbol{\pi}) \approx p(c_{n,\tau}^{(s)} = j | \hat{\boldsymbol{\theta}}, \boldsymbol{x}_n)$, where $\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t}} q(\boldsymbol{t})$ is the most likely sequence of target labels.

The variational Bayes algorithm for BSC-seq is modified to update $q(\boldsymbol{c}^{(s)})$, as shown in Algorithm 1. For the new fac-

Input: Annotations $\boldsymbol{c}$, tokens $\boldsymbol{x}$

Run standard BSC-seq excluding s to compute initial values of $\mathbb{E}\ln\pi^{(k)}, \forall k$, $\mathbb{E}\ln\rho_j, \forall j$, $\mathbb{E}\ln\boldsymbol{T}_j, \forall j$.

Set $\hat{\boldsymbol{c}}^{(s)}_{n,\tau}, \forall n, \forall\tau$ to uniform vectors.

while $\text{not\_converged}(q(\boldsymbol{t}))$ **do**

  Update $\ln\hat{\pi}^{(s)}_{j,\tau}$.

  Update $r_{j,n,\tau}, s_{t_{j,n,\tau-1},t_{\iota,n,\tau}}, \forall j, \forall\tau, \forall n, \forall\iota$ as in BSC, but substitute $\ln\hat{\pi}^{(s)}_{j,\tau}$ for $\pi^{(s)}_{j,c^{(s)}_{n,\tau},c^{(s)}_{n,\tau-1}}$.

  Update $\mathbb{E}\ln\pi^{(k)}, \forall k$, $\mathbb{E}\ln\rho_j, \forall j$, $\mathbb{E}\ln\boldsymbol{T}_j, \forall j$ as in a single iteration of standard BSC.

  Compute $\hat{\boldsymbol{t}}$.

  Run $train(\boldsymbol{x}, \hat{\boldsymbol{t}})$.

  Run $\hat{\boldsymbol{c}}^{(s)}_n = predict(\boldsymbol{x})$.

end

Output: Label posteriors, $r_{n,\tau,j}, \forall n, \forall\tau, \forall j$; most probable sequence of labels, $\hat{\boldsymbol{t}}_n, \forall n$ computed using the Viterbi algorithm

Algorithm 1: The VB algorithm for BSC with variational combined supervision.

tor in the extended model, $q(\boldsymbol{c}^{(s)})$, the algorithm must compute the expected log likelihood of the labels from s by taking an expectation over the values of $c^{(s)}_{n,\tau}$:

$$\ln\hat{\pi}^{(s)}_{j,\tau} = \mathbb{E}\left[\ln\pi^{(s)}_{j,c^{(s)}_{n,\tau},c^{(s)}_{n,\tau-1}}\right] = \sum_{l=1}^{J_s}\sum_{m=1}^{J_s}\hat{c}^{(s)}_{n,\tau,l}\,\hat{c}^{(s)}_{n,\tau-1,m}\mathbb{E}[\ln\pi^{(s)}_{j,m,l}]. \quad (4)$$

VCS can be seen as a form of expectation-maximisation (*EM*) (Dawid and Skene 1979): running $train(\boldsymbol{x}, \hat{\boldsymbol{t}})$ and $predict(\boldsymbol{x})$ performs an expectation step (over the values of $\boldsymbol{c}^{(s)}$), and computing $\hat{\boldsymbol{t}}$ performs a maximisation step (of the likelihood of $\boldsymbol{c}$ given $\boldsymbol{t}$). The complete algorithm for BSC with VCS is therefore a hybrid between VB for learning the BSC parameters and maximum likelihood EM for training the sequence tagger s. The algorithm optimises the evidence lower bound (*ELBO*), which is given as follows:

$$\begin{aligned}\mathcal{L} = {} & \mathbb{E}_{\boldsymbol{t},\boldsymbol{\pi},\boldsymbol{c}^{(s)}}\Big[\ln p\Big(\boldsymbol{c}|\boldsymbol{t},\boldsymbol{\pi}^{(1)},..,\boldsymbol{\pi}^{(K)}\Big)\Big] - \mathbb{E}_{\boldsymbol{c}^{(s)}}\Big[\ln q(\boldsymbol{c}^{(s)})\Big] \\ & + \mathbb{E}_{\boldsymbol{t},\boldsymbol{\rho}}\left[\ln p\left(\boldsymbol{x}|\boldsymbol{t},\boldsymbol{\rho}_1,..,\boldsymbol{\rho}_J\right)\right] + \mathbb{E}_{\boldsymbol{t},\boldsymbol{T}}\left[p(\boldsymbol{t}|\boldsymbol{T}) - q(\boldsymbol{t})\right] \\ & + \sum_{j=1}^{J}\Big\{\mathbb{E}_{\boldsymbol{\rho}}\big[p(\boldsymbol{\rho}_j|\boldsymbol{\kappa}_j) - q(\rho_j)\big] + \mathbb{E}_{\boldsymbol{T}_j}\big[p(\boldsymbol{T}_j|\boldsymbol{\gamma}_j) - q(\boldsymbol{T}_j)\big] \\ & + \sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{\pi}_j}\left[p(\boldsymbol{\pi}^{(k)}_j|\boldsymbol{\alpha}^{(k)}) - q(\boldsymbol{\pi}^{(k)}_j)\right]\Big\}. \quad (5)\end{aligned}$$

4 Experiments

4.1 Datasets

We test two different scenarios for learning a model in a new domain from weak labels: firstly, by combining models trained on other domains using the FAMULUS German datasets (Schulz et al. 2018); secondly, by combining noisy crowdsourced labels on the NER (Rodrigues, Pereira, and Ribeiro 2014) and PICO (Nguyen et al. 2017) English datasets, using VCS to learn an automated sequence tagger. The datasets each have different properties, detailed below, which represent a range of NLP sequence labelling tasks. For example, each contains a different set of class labels and NER contains much shorter spans than FAMULUS or PICO.

Transfer learning data. The FAMULUS datasets comprise diagnostic reasoning annotations in the Medical (Med) and Teacher Education (TEd) domains. Each dataset contains summaries written by students of 8 virtual patients (*cases*), in which the students reason over possible symptomatic diagnoses. The argumentative structure of the diagnoses is categorised into *epistemic activities* (Fischer et al. 2014), covered by sub-spans of the text.

The dataset consists of 4 epistemic activity classes: *hypothesis generation* (HG; the derivation of possible answers to the problem), *evidence generation* (EG; the derivation of evidence, e.g., through deductive reasoning or observing phenomena), *evidence evaluation* (EE; the assessment of whether and to which degree evidence supports an answer to the problem), and *drawing conclusions* (DC; the aggregation and weighing of evidence and knowledge to derive a final answer to the problem) discussed in detail by Schulz et al. (2018). A labelled example is shown in Figure 1.

While the diagnostic texts of the two domains are inherently different, the cases within each domain are also disparate, covering different symptomatic nuances. This fundamentally increases the complexity of the task while radically reducing the data size for each case. Table 1 exhibits the average number of class labels across the two domains. This emphasises, that as well as the limited data set size, the label distribution is highly skewed (e.g. EE vs. EG).

		HG	EG	EE	DC	#Docs
Med	Train	97	41	349	87	106
	Dev	15	5	59	14	16
	Test	16	4	50	11	16
TEd	Train	53	78	452	75	91
	Dev	7	8	67	11	14
	Test	8	10	67	12	14

Table 1: Mean numbers of instances for each class of epistemic activity and mean numbers of diagnostic texts (#Docs) across the 8 cases for the two FAMULUS datasets.

Crowdsourced data. We experiment with two crowdsourced datasets with different types of span annotations. Firstly, the CoNLL 2003 named-entity recognition dataset (*NER*) (Tjong Kim Sang and De Meulder 2003), with crowdsourced annotations from Rodrigues, Pereira, and Ribeiro (2014). Secondly, a dataset of medical paper abstracts (*PICO*) (Nguyen et al. 2017), with spans that identify the

First I wanted to see if the problem was new, so I checked the teachers observations. As it was the same back then, I ruled out a trauma or another dramatic event. I was then undecided between autism and ADHD, since his social behaviour seems to be problematic and that's a sign for both diagnoses. In the end, I settled on ADHD since his script seems chaotic and unorganised and because he seems to have some friends despite his difficult behaviour.

Figure 1: Example text from the TEd dataset, with highlighted spans for EG (green), EE (underlined), DC (yellow), HG (blue).

population enrolled in a clinical trial. While NER contains mainly short spans (on average 1.5 tokens) for four categories of named entity (PER, LOC, ORG, MISC), the spans in PICO belong to one class and are on average 7.7 tokens long. The statistics in Table 2 show how that NER is a much larger dataset than FAMULUS or PICO, albeit with fewer annotators than PICO. Crowdsourced labels are provided for all documents (i.e. the total figure in Table 2), while gold labels are provided for development and test subsets[2].

		#Spans	#Docs	#Annotators	
NER	Total	21,612	6,056	Total	47
	Dev	1,285	2,800	Per doc	4.9
	Test	1,516	3,256		
PICO	Total	n/a	9,480	Total	312
	Dev	351	191	Per doc	6.0
	Test	349	191		

Table 2: Crowdsourced dataset statistics. For PICO, gold labels are are not available for all documents, hence the correct number of spans is unknown.

Evaluation metrics. We use two span-level F1-scores: for NER, where matching the exact span is important to recognise the entity correctly, we use the CoNLL 2003 F1-score (Tjong Kim Sang and De Meulder 2003), which counts only exact span matches as true positives; for the other datasets, which involve longer spans with more ambiguous boundaries, we use a relaxed F1-score that counts fractions of partial span matches when computing true and false positives. For the crowdsourcing scenarios, we also evaluate the probabilities output by each method using cross entropy error (*CEE*), which penalises both over- and under-confident predictions and is therefore a useful metric of the quality of confidence estimates.

4.2 Sequence Taggers

We use the BiLSTM-LSTM-CRF model of Lample et al. (2016) with the Adam optimiser, dropout of 0.25, and character embedding size of 100, which was previously found to give strong performance (Reimers and Gurevych 2017). For FAMULUS, we use 300-dimensional German fastText embeddings (Grave et al. 2018), and for NER and PICO we use 300-dimensional English GloVe 3 embeddings trained on 840 billion tokens from Common Crawl. To reduce the effect of random initialisation, each tagger was trained with 10 different random seeds, then we selected the model with highest performance on the development set of the training domain. For brevity, we refer to this model as *DNN*.

As the FAMULUS datasets are small, we also evaluate a shallow model, namely the CRF, using the Sklearn-crfsuite implementation[3] with L-BFGS-B optimiser and L1 and L2 regularisation coefficients set to 0.1.

4.3 Transfer Learning by Combining Models

The first scenario investigates transfer learning between the cases in each of the FAMULUS datasets. We hypothesise that combining several models trained on different cases using BSC will outperform any individual out-of-domain sequence tagger and alternative combination methods. To test this, we first train separate DNNs and CRFs on each of the cases. We train separate models to predict each class of epistemic activity, providing a set of 16 base models per class. We then combine the models from different cases for each class with the following methods: majority vote (*MV*), which assigns, for each token, the label assigned by the most base models; IBCC (Kim and Ghahramani 2012), which models the likelihood of annotations given the ground truth but ignores the sequence of tokens; and BSC-seq, which models sequential dependencies as described in Section 3. To provide a target performance level for the ensembles, we also train a DNN and a CRF on the combined training set for all cases (labelled *All-domain* in the results). Simple grid searches are used to set the hyperparameters of each method using development set performance. The results here show performance on the test sets.

Table 3 reports the results for the TEd and MEd datasets. For Med, the DNN base models mostly outperform the CRFs, while for TEd, this is reversed. This may relate to the quantity of training data, as the DNN requires more data. However, for EE, the CRF performs best on TEd, which has the largest training set, suggesting that a shallow model may still be sufficient in some cases. In both datasets, the strongest performance is mostly achieved by the all-domain model, which has access to training data from the target domain. However, we are interested in scenarios where data in the target domain is unavailable. The out-of-domain DNNs and CRFs are consistently out-performed by ensembles using IBCC and BSC-seq. Majority vote is less consistent: for example, it performs worse than a single out-of-domain model in the HG and DC classes with both Med and TEd. This highlights the importance of learning annotator reliability. BSC-seq has the best performance in general, showing the benefits of a sequential model.

[2]Data splits for NER are from Nguyen et al. (2017); for PICO we make a new random split (see our Github repository), as splits from prior work were not available.

[3]https://sklearn-crfsuite.readthedocs.io/en/latest/index.html

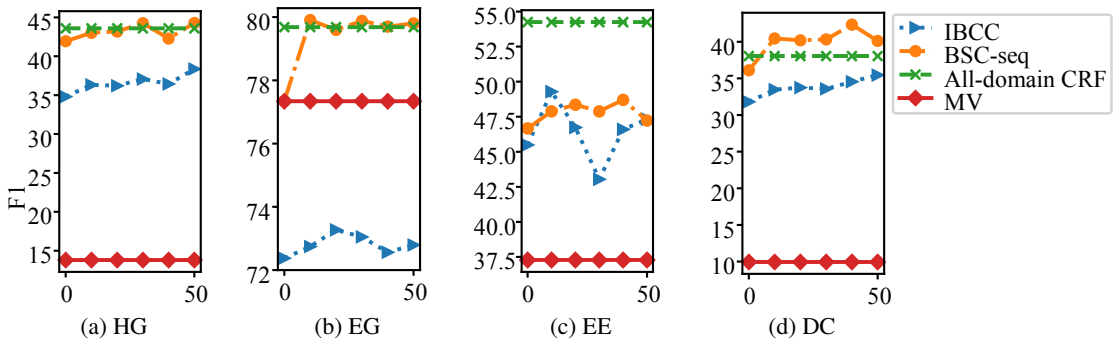

Figure 2: Semi-supervised learning on FAMULUS: x-axis shows the increasing number of training sentences in the target domain used to train BSC-seq and IBCC. MV is not trainable. All-domain was trained using the full target domain training set.

			Med					**TEd**				
	Setup	Base model	HG	EG	EE	DC	Mean	HG	EG	EE	DC	Mean
In-domain	Single	DNN	47.5	73.9	28.9	36.1	46.6	23.8	74.9	44.9	14.9	39.6
	Single	CRF	43.8	77.6	21.5	33.3	44.1	25.4	76.7	43.5	14.6	40.0
	All-domain	DNN	***50.1***	77.5	***43.7***	**47.0**	***54.6***	39.1	***82.7***	58.1	16.3	49.1
	All-domain	CRF	46.3	***80.0***	28.1	43.1	49.4	***41.6***	79.0	***65.2***	**27.2**	***53.3***
Out-of-domain, base models trained on one domain	Single	DNN	39.4	72.8	20.3	25.8	39.6	18.9	69.1	32.6	8.6	32.4
	Single	CRF	20.7	70.1	19.3	22.5	33.2	15.5	69.9	43.4	11.0	34.9
	MV	DNN,CRF	25.5	76.6	32.8	19.9	38.7	2.1	**78.0**	41.7	0.0	30.5
	IBCC	DNN,CRF	45.7	74.0	38.1	43.8	50.4	23.9	70.7	52.9	19.8	41.8
	BSC-seq	DNN	**49.9**	78.4	**42.8**	41.3	53.1	31.2	77.0	50.2	15.8	43.6
	BSC-seq	CRF	23.3	**78.6**	34.8	38.3	43.7	30.2	77.1	**63.6**	14.5	46.3
	BSC-seq	DNN,CRF	49.0	78.5	40.0	***49.2***	**54.2**	**32.6**	75.3	51.4	**27.1**	**46.6**

Table 3: Relaxed span-level F1 scores (counting proportions of matches) on the FAMULUS datasets by class and mean over classes. 'Single' refers to a single base model trained on a single source domain; 'All-domain' refers to a single base model trained on all domains, including the target domain. Bold indicates best in-domain or best out-of-domain performance and bold-italic indicates best overall performance.

Although the DNN is stronger than the CRF on Med, including CRFs as well as DNNs in the ensemble does improve performance. We see the same pattern in TEd. Note the strong performance on the Med DC class, where the ensemble using BSC-seq to combine DNNs and CRFs outperforms even the strongest in-domain model. The CRFs and DNNs appear to provide complementary information that BSC-seq is able to exploit.

Both IBCC and BSC-seq can be trained using gold labels: the values of $\boldsymbol{t}$ with gold labels are fixed to these known values and are not updated, while the inference algorithm otherwise proceeds as normal. We hypothesise that by including small amounts of labelled data in the target case, we can improve the performance of IBCC and BSC-seq by learning more accurate annotator models. To test this, we form ensembles of DNNs and CRFs using IBCC and BSC-seq, as in the previous experiment, then iteratively introduce an increasing number of labelled sentences, selected at random.

Figure 2 plots the increasing F1-scores of BSC-seq and IBCC, averaged over Med and TEd, along with the majority vote baseline, which cannot take advantage of any in-domain data, and the all-domain model, which was trained on labelled data from all domains, including all training data from the target domain. The results show that performance of IBCC and BSC-seq improves with small numbers of labelled documents. Across all the classes, BSC-seq outperforms both IBCC and MV. IBCC, outperforms MV in three classes, but is worse on the EG class, while BSC-seq improves quickly on EG as training labels are added. BSC-seq is competitive with the all-domain CRF on three of the four classes, slightly out-performing it in the DC class, despite having very little training data in the target domain.

4.4 Enhancing a Crowd with an Automated Sequence Tagger

In our second scenario, we are presented with noisy labels from a crowd of human annotators. Different documents are labelled by different combinations of annotators, meaning that some documents may be annotated only by weaker annotators. To overcome this, we use VCS with BSC-seq to introduce an automated sequence tagger into the ensemble as an additional annotator. We perform the following exper-

	NER		PICO	
	F1	CEE	F1	CEE
Best worker	67.3	17.1	58.5	17.0
MV	65.4	6.24	64.3	2.55
IBCC	74.4	**0.49**	68.9	**0.27**
HMM-crowd	74.6	1.04	71.0	0.79
HMM-crowd→DNN	75.2	12.2	71.2	13.0
BSC-seq	77.4	0.65	72.8	0.53
BSC-seq→DNN	77.7	11.0	75.5	25.5
BSC-seq+VCS	**78.0**	0.99	**77.5**	1.15

Table 4: Performance of aggregation methods on crowdsourced data, including training a DNN on the output of an aggregation method (→DNN) and by using VCS.

iment on the NER and PICO datasets: train several rival aggregation methods on the crowdsourced data in each dataset; use the development sets to tune hyperparameters by grid search; evaluate the aggregated labels on the test sets.

Besides MV, IBCC and BSC-seq, we also test HMM-crowd (Nguyen et al. 2017). HMM-crowd is a probabilistic method that captures sequential dependencies between class labels, but uses a simpler model of annotator labelling bias that does not consider sequential dependencies between the annotator's labels. We compare BSC-seq with VCS (*BSC-seq+VCS*) against a pipeline method that first aggregates the labels with HMM-crowd or BSC-seq, then uses the aggregated labels to train a DNN, which then predicts the labels for the entire dataset. In the results, HMM-crowd→DNN and BSC-seq→DNN show the performance of the trained DNN. The advantage of VCS over the pipeline is that errors detected by the DNN are used to correct the ensemble, which in turn leads to better training data when training the DNN in the next epoch. Furthermore, the annotator model learns the reliability of the DNN and provides posterior probabilities that take into account its noise and bias.

Table 4 shows a clear advantage in terms of F1-score to BSC-seq+VCS. CEE is much lower than the pipeline method (BSC-seq→DNN), although IBCC performs best in this regard. CEE is a token-level metric, however, so may be biased toward the dominant 'O' (outside) class, which IBCC predicts with greater confidence in a majority of cases. The BSC-seq variants outperform the simpler sequential model, HMM-crowd, showing the advantage of the sequential annotator model. The results suggest that VCS could reduce the number of workers required to annotate each document by acting as an additional annotator, thereby reducing the crowdsourcing costs.

5 Conclusions

We proposed to use Bayesian sequence combination (BSC) to address two different scenarios in which reliable training data is unavailable: transferring models between domains and combining crowds of human classifiers. Our experiments showed that BSC can be an effective way to transfer sequence taggers between domains when given small amounts or no training data in the target domain. We also introduced variational combined supervision (VCS), a novel method for training a sequence tagger directly from the ensemble and integrating it back into the ensemble. Experiments on two very different crowdsourced datasets showed that VCS can improve the performance of an ensemble of human labellers. Despite the differences between the datasets and tasks tested, we see a common pattern supporting the use of BSC for learning from weak labels.

In future work, we plan to test the approach in other entity annotation tasks besides NER, such as mention detection for entity linking, and other long-span annotation tasks besides FAMULUS and PICO, such as argument mining. We therefore intend this paper as a basis for future work that will make use of our software implementation. In future work, we will also investigate how VCS can be used to reduce the number of annotations required from human annotators, and how effectively it distils an ensemble down to a single sequence tagger to avoid obtaining predictions from a large number of base models in low resource settings.

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF) under the promotional references 16DHL1040 (FAMULUS) and 03VP02540 (ArgumenText).

References

Albarqouni, S.; Baur, C.; Achilles, F.; Belagiannis, V.; Demirci, S.; and Navab, N. 2016. Aggnet: deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE transactions on medical imaging* 35(5):1313–1321.

Brown, G.; Wyatt, J.; Harris, R.; and Yao, X. 2005. Diversity creation methods: a survey and categorisation. *Information Fusion* 6(1):5–20.

Dawid, A. P., and 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.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Feng, X.; Feng, X.; Qin, B.; Feng, Z.; and Liu, T. 2018. Improving low resource named entity recognition using cross-lingual knowledge transfer. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 4071–4077. International Joint Conferences on Artificial Intelligence Organization.

Fischer, F.; Kollar, I.; Ufer, S.; Sodian, B.; Hussmann, H.; Pekrun, R.; Neuhaus, B.; Dorner, B.; Pankofer, S.; Fischer, M. R.; Strijbos, J.-W.; Heene, M.; and Eberle, J. 2014. Scientific Reasoning and Argumentation: Advancing an Interdisciplinary Research Agenda in Education. *Frontline Learning Research* 4:28–45.

Grave, E.; Bojanowski, P.; Gupta, P.; Joulin, A.; and Mikolov, T. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language*

Resources and Evaluation (LREC-2018). Miyazaki, Japan: European Languages Resources Association (ELRA).

Hovy, D.; Berg-Kirkpatrick, T.; Vaswani, A.; and Hovy, E. 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*, 1120–1130. Atlanta, Georgia: Association for Computational Linguistics.

Kim, H.-c., and Ghahramani, Z. 2012. Bayesian classifier combination. In *International Conference on Artificial Intelligence and Statistics*, 619–627.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 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*, 260–270. San Diego, California: Association for Computational Linguistics.

Lin, Y.; Yang, S.; Stoyanov, V.; and Ji, H. 2018. A multi-lingual multi-task architecture for low-resource sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, 799–809.

Mayhew, S.; Tsai, C.-T.; and Roth, D. 2017. Cheap translation for cross-lingual named entity recognition. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2536–2545.

Nguyen, A. T.; Wallace, B. C.; Li, J. J.; Nenkova, A.; and Lease, M. 2017. Aggregating and predicting sequence labels from crowd annotations. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2017, 299. NIH Public Access.

Peters, M.; Ammar, W.; Bhagavatula, C.; and Power, R. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1756–1765.

Peters, M. E.; Ruder, S.; and Smith, N. A. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, 7–14. Florence, Italy: Association for Computational Linguistics.

Plank, B.; Hovy, D.; and Søgaard, A. 2014. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 742–751.

Rahimi, A.; Li, Y.; and Cohn, T. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 151–164. Florence, Italy: Association for Computational Linguistics.

Rehbein, I., and Ruppenhofer, J. 2017. Detecting annotation noise in automatically labelled data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1160–1170. Vancouver, Canada: Association for Computational Linguistics.

Rei, M., and Søgaard, A. 2018. Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 293–302.

Reimers, N., and Gurevych, I. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799, version 2*.

Rodrigues, F., and Pereira, F. C. 2018. Deep learning from crowds. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), 2018*.

Rodrigues, F.; Pereira, F.; and Ribeiro, B. 2014. Sequence labeling with multiple annotators. *Machine learning* 95(2):165–181.

Ruder, S. 2019. *Neural Transfer Learning for Natural Language Processing*. Ph.D. Dissertation, National University of Ireland, Galway.

Schulz, C.; Meyer, C. M.; Sailer, M.; Kiesewetter, J.; Bauer, E.; Fischer, F.; Fischer, M. R.; and Gurevych, I. 2018. Challenges in the automatic analysis of students' diagnostic reasoning. *arXiv preprint arXiv:1811.10550*.

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

Simpson, E. D., and Gurevych, I. 2019. A Bayesian approach for sequence tagging with crowds. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1093–1104. Hong Kong, China: Association for Computational Linguistics.

Simpson, E.; Roberts, S.; Psorakis, I.; and Smith, A. 2013. Dynamic Bayesian combination of multiple imperfect classifiers. *Intelligent Systems Reference Library series* Decision Making with Imperfect Decision Makers:1–35.

Tjong Kim Sang, E. F., and De Meulder, F. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 142–147. Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 353–355.

Yang, J.; Drake, T.; Damianou, A.; and Maarek, Y. 2018. Leveraging crowdsourcing data for deep active learning an application: Learning intents in Alexa. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 23–32. International World Wide Web Conferences Steering Committee.

Yang, Z.; Salakhutdinov, R.; and Cohen, W. W. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Zhou, J. T.; Zhang, H.; Jin, D.; Zhu, H.; Fang, M.; Goh, R. S. M.; and Kwok, K. 2019. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, 3461–3471.