Crowdsourcing Learning as Domain Adaptation: A Case Study on Named Entity Recognition

Xin Zhang[1], **Guangwei Xu, Yueheng Sun**[2], **Meishan Zhang** [1*], **Pengjun Xie**
[1]School of New Media and Communication, Tianjin University, China
[2]College of Intelligence and Computing, Tianjin University, China
{hsinz,yhs,zhangmeishan}@tju.edu.cn
{ahxgwOnePiece,xpjandy}@gmail.com

Abstract

Crowdsourcing is regarded as one prospective solution for effective supervised learning, aiming to build large-scale annotated training data by crowd workers. Previous studies focus on reducing the influences from the noises of the crowdsourced annotations for supervised models. We take a different point in this work, regarding all crowdsourced annotations as gold-standard with respect to the individual annotators. In this way, we find that crowdsourcing could be highly similar to domain adaptation, and then the recent advances of cross-domain methods can be almost directly applied to crowdsourcing. Here we take named entity recognition (NER) as a study case, suggesting an annotator-aware representation learning model that inspired by the domain adaptation methods which attempt to capture effective domain-aware features. We investigate both unsupervised and supervised crowdsourcing learning, assuming that no or only small-scale expert annotations are available. Experimental results on a benchmark crowdsourced NER dataset show that our method is highly effective, leading to a new state-of-the-art performance. In addition, under the supervised setting, we can achieve impressive performance gains with only a very small scale of expert annotations.

1 Introduction

Crowdsourcing has gained a growing interest in the natural language processing (NLP) community, which helps hard NLP tasks such as named entity recognition (Finin et al., 2010; Derczynski et al., 2016), part-of-speech tagging (Hovy et al., 2014), relation extraction (Abad et al., 2017), translation (Zaidan and Callison-Burch, 2011), argument retrieval (Mayhew et al., 2020), and others (Snow et al., 2008; Callison-Burch and Dredze, 2010) to collect a large scale dataset for supervised model training. In contrast to the gold-standard annotations labeled by experts, the crowdsourced annotations can be constructed quickly at a low cost with masses of crowd annotators (Snow et al., 2008; Nye et al., 2018). However, these annotations are relatively lower-quality with much-unexpected noise since the crowd annotators are not professional enough, which can make errors in complex and ambiguous contexts (Sheng et al., 2008).

text	*Andrea*	*Ferrigato*	*sprinted*	*to*	*his*	*World*	*Cup*	*win*
A-1	B-PER	I-PER	O	O	O	B-ORG	I-ORG	O
A-2	O	B-PER	O	O	O	B-MISC	I-MISC	O
A-3	B-PER	I-PER	O	O	O	B-ORG	I-ORG	O
EXP	B-PER	I-PER	O	O	O	B-MISC	I-MISC	O

Figure 1: A NER example with crowdsourced labels, A and EXP denote annotator and expert, respectively.

Previous crowdsourcing learning models struggle to reduce the influences of noises of the crowdsourced annotations (Hsueh et al., 2009; Raykar and Yu, 2012a; Hovy et al., 2013; Jamison and Gurevych, 2015). Majority voting (MV) is one straightforward way to aggregate high-quality annotations, which has been widely adopted (Snow et al., 2008; Fernandes and Brefeld, 2011; Rodrigues et al., 2014), but it requires multiple annotations for a given input. Recently, the majority of models concentrate on monitoring the distances between crowdsourced and gold-standard annotations, obtaining better performances than MV by considering the annotator information together (Nguyen et al., 2017; Simpson and Gurevych, 2019; Li et al., 2020). Most of these studies assume the crowdsourced annotations as untrustworthy answers, proposing sophisticated strategies to recover the golden answers from crowdsourced labels.

In this work, we take a different view for crowdsourcing learning, regarding the crowdsourced annotations as the gold standard in terms of individual

*Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5558–5570
August 1–6, 2021.

annotators. In other words, we assume that all annotators (including experts) own their specialized understandings towards a specific task, and they annotate the task consistently according to their individual principles by the understandings, where the experts can reach an oracle principle by consensus. The above view indicates that crowdsourcing learning aims to train a model based on the understandings of crowd annotators, and then test the model by the oracle understanding from experts.

Based on the assumption, we find that crowdsourcing learning is highly similar to domain adaptation, which is one important topic that has been investigated extensively for decades (Ben-David et al., 2006; Daumé III, 2007; Chu and Wang, 2018; Jia and Zhang, 2020). We treat each annotator as one domain specifically, and then crowdsourcing learning is essentially almost a multi-source domain adaptation problem. Thus, one natural question arises: What is the performance when a state-of-the-art domain adaptation model is applied directly to crowdsourcing learning.

Here we take NER as a study case to investigate crowdsourcing learning as domain adaptation, considering that NER has been one popular task for crowdsourcing learning in the NLP community (Finin et al., 2010; Rodrigues et al., 2014; Derczynski et al., 2016). We suggest a state-of-the-art representation learning model that can effectively capture annotator(domain)-aware features. Also, we investigate two settings of crowdsourcing learning, one being the unsupervised setting with no expert annotation, which has been widely studied before, and the other being the supervised setting where a certain scale of expert annotations exists, which is inspired by domain adaptation.

Finally, we conduct experiments on a benchmark crowdsourcing NER dataset (Tjong Kim Sang and De Meulder, 2003; Rodrigues et al., 2014) to evaluate our methods. We take a standard BiLSTM-CRF (Lample et al., 2016) model with BERT (Devlin et al., 2019) word representations as the baseline, and adapt it to our representation learning model. Experimental results show that our method is able to model crowdsourced annotations effectively. Under the unsupervised setting, our model can give a strong performance, outperforming previous work significantly. In addition, the model performance can be greatly boosted by feeding with small-scale expert annotations, which can be a prospective direction for low-resource scenarios.

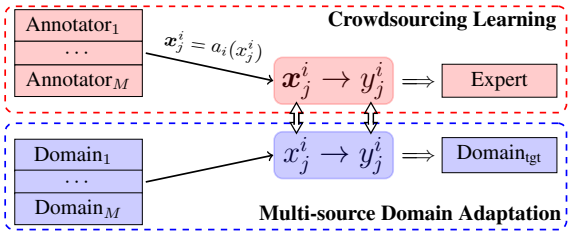


Figure 2: Illustration of the connection between multi-source domain adaptation and crowdsourcing learning.

In summary, we make the following three major contributions:

(1) We present a different view of crowdsourcing learning, and propose to treat crowdsourcing learning as domain adaptation, which naturally connects the two important topics of machine learning for NLP.

(2) We propose a novel method for crowdsourcing learning. Although the method is of a limited novelty for domain adaptation, it is the first work to crowdsourcing learning, and can achieve state-of-the-art performance on NER.

(3) We introduce supervised crowdsourcing learning for the first time, which is borrowed from domain adaptation and would be a prospective solution for hard NLP tasks in practice.

We will release the code and detailed experimental settings at github.com/izhx/CLasDA under the Apache License 2.0 to facilitate future research.

2 The Basic Idea

Here we describe the concepts of the domain adaptation and crowdsourcing learning in detail, and show how they are connected together.

2.1 Domain Adaptation

Domain adaptation happens when a supervised model trained on a fixed set of training corpus, including several specific domains, is required to test on a different domain (Ben-David et al., 2006; Mansour et al., 2009). The scenario is quite frequent in practice, and thus has received extensive attention with massive investigations (Csurka, 2017; Ramponi and Plank, 2020). The major problem lies in the different input distributions between source and target domains, leading to biased predictions over the inputs with a large gap to the source domains.

Here we focus on multi-source cross-domain adaptation, which would suit our next corresponding mostly. Following Mansour et al. (2009); Zhao et al. (2019), the multi-source domain adaptation assumes a set of labeled examples from M domains available, denoted by $D_{\text{src}} = \{(X_i, Y_i)\}_{i=1}^{M}$,[1] where $X_i = \{x_j^i\}_{j=1}^{N_i}$ and $Y_i = \{y_j^i\}_{j=1}^{N_i}$,[2] and we aim to train a model on D_{src} to adapt to a specific target domain with the help of a large scale raw corpus $X_{\text{tgt}} = \{x_i\}_{i=1}^{N_t}$ of the target domain.

Note that under this setting, all Xs, including source and target domains, are generated individually according to their unknown distributions, thus the abstract representations learned from the source domain dataset D_{src} would inevitably be biased to the target domain, which is the primary reason for the degraded performance of the target domain (Huang and Yates, 2010; Ganin et al., 2016). A number of domain adaptation models have struggled for better transferable high-level representations as domain shifts (Ramponi and Plank, 2020).

2.2 Crowdsourcing Learning

Crowdsourcing aims to produce a set of large-scale annotated examples created by crowd annotators, which is used to train supervised models for a given task (Raykar et al., 2010). As the majority of NLP models assume that gold-standard high-quality training corpora are already available (Manning and Schutze, 1999), crowdsourcing learning has received much less interest than cross-domain adaptation, although the availability of these corpora is always not the truth.

Formally, under the crowdsourcing setting, we usually assume that there are a number of crowd annotators $A = \{a_i\}_{i=1}^{M}$ (here we use the same M as well as later superscripts in order to align with the domain adaptation), and all annotators should have a sufficient number of training examples by their different understandings for a given task, which are referred to as $D_{\text{crowd}} = \{(X_i, Y_i)\}_{i=1}^{M}$ where $X_i = \{x_j^i\}_{j=1}^{N_i}$ and $Y_i = \{y_j^i\}_{j=1}^{N_i}$. We aim to train a model on D_{crowd} and adapt it to predict the expert outputs. Note that all Xs do not have significant differences in their distributions in this paradigm.

[1] A domain is commonly defined as a distribution on the input data in many works, e.g., Ben-David et al. (2006). To make domain adaptation and crowdsourcing learning highly similar in formula, we follow Zhao et al. (2019), defining a domain as a joint distribution on the input space $\mathcal{X}$ and the label space $\mathcal{Y}$. Section 4.5 gives a discussion of their connection.

[2] N_* indicates the number of instances.

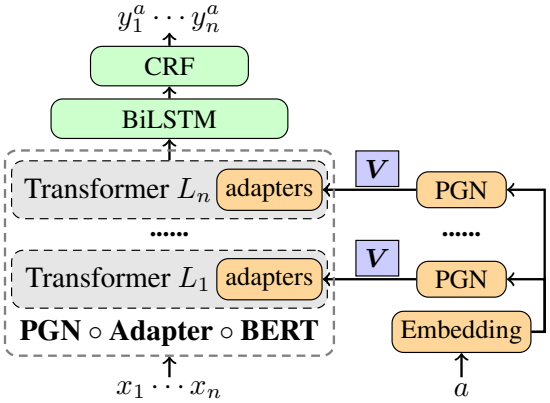


Figure 3: The structure of our representation learning model, where the right orange part denotes the annotator switcher, and $\boldsymbol{V}$ denotes the generated adapter parameters by PGN. The transformer layers in gray are kept frozen in training, and other modules are trainable.

Crowdsourcing Learning as Domain adaptation By scrutinizing the above formalization, when we set all Xs jointly with the annotators by using $\boldsymbol{x}_j^i = a_i(x_j^i)$, which indicates the contextualized understanding (a vectorial form is desirable here of the neural representations) of x_j^i by the annotator a_i, then we would regard that $\boldsymbol{X}_i = \{a_i(x_j^i)\}_{j=1}^{N_i}$ is generated from different distributions as well. In this way, we are able to connect crowdsourcing learning and domain adaptation together, as shown in Figure 2, based on the assumption that all Ys are gold-standard for crowdsourced annotations when crowd annotators are united as joint inputs. And finally, we need to perform predictions by regarding $\boldsymbol{x}_{\text{expert}} = \text{expert}(x)$, and in particular, the learning of expert differs from that of the target domain in domain adaptation.

3 A Case Study On NER

In this section, we take NER as a case study, which has been investigated most frequently in NLP (Yadav and Bethard, 2018), and propose a representation learning model mainly inspired by the domain adaptation model of (Jia et al., 2019) to perform crowdsourcing learning. In addition, we introduce the unsupervised and supervised settings for crowdsourcing learning which are directly borrowed from the domain adaptation.

3.1 The Representation Learning Model

We convert NER into a standard sequence labeling problem by using the BIO schema, following the majority of previous works, and extend a state-of-the-art BERT-BiLSTM-CRF model (Mayhew

et al., 2020) to our crowdsourcing learning. Figure 3 shows the overall network structure of our representation learning model. By using a sophisticated parameter generator module (Platanios et al., 2018), it can capture annotator-aware features. Following, we introduce the proposed model by four components: (1) word representation, (2) annotator switcher, (3) BiLSTM Encoding, and (4) CRF inference and training.

Word Representation Given a sentence of n words $x = w_1 \cdots w_n$, we first convert it to vectorial representations by BERT. Different from the standard BERT exploration, here we use Adapter∘BERT (Houlsby et al., 2019), where two extra adapter modules are inside each transformer layer. The process can be simply formalized as:

$$\boldsymbol{e}_1 \cdots \boldsymbol{e}_n = \text{Adapter} \circ \text{BERT}(w_1 \cdots w_n) \quad (1)$$

where $\circ$ indicates an injection operation. The detailed structure of the transformer with adapters is described in Appendix A.

Noticeably, the Adapter $\circ$ BERT method no longer needs fine-tuning the huge BERT parameters and can obtain comparable performance by adjusting the much lightweight adapter parameters instead. Thus the representation can be more parameter efficient, and in this way we can easily extend the word representations to annotator-aware representations.

Annotator Switcher Our goal is to efficiently learn annotator-aware word representations, which can be regarded as contextualized understandings of individual annotators. Hence, we introduce an annotator switcher to support Adapter∘BERT with annotator input as well, which is inspired by Üstün et al. (2020). The key idea is to use Parameter Generation Network (PGN) (Platanios et al., 2018; Jia et al., 2019) to produce adapter parameters dynamically by input annotators. In this way, our model can flexibly switch among different annotators.

Concretely, assuming that $\boldsymbol{V}$ is the vectorial form of all adapter parameters by a pack operation, which can also be unpacked to recover all adapter parameters as well, the PGN module is to generate $\boldsymbol{V}$ for Adapter $\circ$ BERT dynamically according the annotator inputs, as shown in Figure 3 by the right orange part. The switcher can be formalized as:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{r}'_1 \cdots \boldsymbol{r}'_n \\ &= \text{PGN} \circ \text{Adapter} \circ \text{BERT}(x, a) \\ &= \text{Adapter} \circ \text{BERT}(x, \boldsymbol{V} = \boldsymbol{\Theta} \times \boldsymbol{e}^a), \end{aligned} \quad (2)$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{|\boldsymbol{V}| \times |\boldsymbol{e}^a|}$, $\boldsymbol{x} = \boldsymbol{r}'_1 \cdots \boldsymbol{r}'_n$ is the annotator-aware representations of annotator a for $x = w_1 \cdots w_n$, and $\boldsymbol{e}^a$ is the annotator embedding.

BiLSTM Encoding Adapter∘BERT requires an additional task-oriented module for high-level feature extraction. Here we exploit a single BiLSTM layer to achieve it: $\boldsymbol{h}_1 \cdots \boldsymbol{h}_n = \text{BiLSTM}(\boldsymbol{x})$, which is used for next-step inference and training.

CRF Inference and Training We use CRF to calculate the score of a candidate sequential output $y = l_1 \cdots l_n$ globally:

$$\begin{aligned} \boldsymbol{o}_i &= \boldsymbol{W}^{\text{crf}} \boldsymbol{h}_i + \boldsymbol{b}^{\text{crf}} \\ \text{score}(y|x, a) &= \sum_{i=1}^{n} (\boldsymbol{T}[l_{i-1}, l_i] + \boldsymbol{o}_i[l_i]) \end{aligned} \quad (3)$$

where $\boldsymbol{W}^{\text{crf}}$, $\boldsymbol{b}^{\text{crf}}$ and $\boldsymbol{T}$ are model parameters.

Given an input (x, a), we perform inference by the Viterbi algorithm. For training, we define a sentence-level cross-entropy objective:

$$\begin{aligned} p(y^a|x, a) &= \frac{\exp\big(\text{score}(y^a|x, a)\big)}{\sum_y \exp\big(\text{score}(y|x, a)\big)} \\ \mathcal{L} &= -\log p(y^a|x, a) \end{aligned} \quad (4)$$

where y^a is the gold-standard output of x from a, y belongs to all possible candidates, and $p(y^a|x, a)$ indicates the sentence-level probability.

3.2 The Unsupervised Setting

Here we introduce unsupervised crowdsourcing learning in alignment with unsupervised domain adaptation, assuming that no expert annotation is available, which is the widely-adopted setting of previous work of crowdsourcing learning (Sheng et al., 2008; Zhang et al., 2016; Sheng and Zhang, 2019). This setting has a large divergence with domain adaptation in target learning. In the unsupervised domain adaptation, the information of the target domain can be learned through a large-scale raw corpus (Ramponi and Plank, 2020), where there is no correspondence in the unsupervised crowdsourcing learning to learn information of experts.

To this end, here we suggest a simple and heuristic method to model experts by the specialty of crowdsourcing learning. Intuitively, we expect that experts should approve the knowledge of the common consensus for a given task, and meanwhile, our model needs the embedding representation of experts for inference. Thus, we can estimate the

expert embedding by using the centroid point of all annotator embeddings:

$$\boldsymbol{e}^{\text{expert}} = \frac{1}{|A|} \sum_{a \in A} \boldsymbol{e}^{a} \quad (5)$$

where A represents all annotators contributed to the training corpus. This expert can be interpreted as the elected outcome by annotator voting with equal importance. In this way, we perform the inference in unsupervised crowdsourcing learning by feeding $\boldsymbol{e}^{\text{expert}}$ as the annotator input.

3.3 The Supervised Setting

Inspired by the supervised domain adaptation, we also present the supervised crowdsourcing learning, which has been seldom concerned. The setting is very simple, just by assuming that a certain scale of expert annotations is available. In this way, we can learn the expert representation directly by supervised learning with our proposed model.

The supervised setting could be a more practicable scenario in real applications. Intuitively, it should bring much better performance than the unsupervised setting with few shot expert annotations, which does not increase the overall annotation cost much. In fact, during or after the crowdsourcing annotation process, we usually have a quality control module, which can help to produce silvery quality pseudo-expert annotations (Kittur et al., 2008; Lease, 2011). Thus, the supervised setting can be highly valuable yet has been ignored mostly.

4 Experiments

4.1 Setting

Dataset We use the CoNLL-2003 NER English dataset (Tjong Kim Sang and De Meulder, 2003) with crowdsourced annotations provided by Rodrigues and Pereira (2018) to investigate our methods in both unsupervised and supervised settings. The crowdsourced annotations consume 400 new articles, involving 5,985 sentences in practice, which are labeled by a total of 47 crowd annotators. The total number of annotations is 16,878. Thus the averaged number of annotated sentences per annotator is 359, which covers 6% of the total sentences. The dataset includes golden/expert annotations on the training sentences and a standard CoNLL-2003 test set for NER evaluation.

Evaluation The standard CoNLL-2003 evaluation metric is used to calculate the NER performance, reporting the entity-level precision (P), recall (R), and their F1 value. All experiments of the same setting are conducted by five times, and the median outputs are used for performance reporting. We exploit the pair-wise t-test for significance test, regarding two results significantly different when the p-value is below 10^{-5}.

Model	P	R	F1
Annotator-Agnostic			
ALL	76.35	**72.47**	74.36
MV	**83.61**	68.47	**75.28**
Annotator-Aware			
LC	78.59	74.54	76.51
LC-cat	74.34	**79.41**	76.79
This Work	**78.84**	75.67	**77.95**
Previous Work			
(Rodrigues et al., 2014)	49.40	85.60	62.60
LC (Nguyen et al., 2017)	**82.38**	62.10	70.82
LC-cat (Nguyen et al., 2017)	79.61	62.87	70.26
(Rodrigues and Pereira, 2018)	66.00	59.30	62.40
(Simpson and Gurevych, 2019)†	80.30	**74.80**	**77.40**

Table 1: The test results of the unsupervised setting, where the superscript † indicates that there exist differences in the test corpus.

Baselines We re-implement several methods of previous work as baselines, and all the methods are based on Adapter∘BERT-BiLSTM-CRF (no annotator switcher inside) for fair comparisons.

For both the unsupervised and supervised settings, we consider the following baseline models:

- ALL: which treats all annotations equally, ignoring the annotator information no matter crowd or expert.
- MV: which is borrowed from Rodrigues et al. (2014), where aggregated labels are produced by token level majority voting. In particular, the gold-standard labels are used instead if they are available for a specific sentence during the supervised crowdsourcing learning.
- LC: which is proposed by Nguyen et al. (2017), where the annotator bias to the gold-standard labels is explicitly modeled at the CRF layer for each crowd annotator, and specifically, the expert is with zero bias.
- LC-cat: which is also presented by Nguyen et al. (2017) as a baseline to LC, where the annotator bias is modeled at the BiLSTM layer instead and also the expert bias is set to zero.[3]

[3]Note that although LC-cat is not as expected as LC in (Nguyen et al., 2017), our results show that LC-cat is slightly better based on Adapter∘BERT-BiLSTM-CRF.

Model	1%			5%			25%			100%		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Annotator-Agnostic												
ALL	75.08	**74.82**	**74.95**	76.18	75.71	75.94	78.64	78.93	78.78	86.65	82.29	84.42
MV	**83.87**	67.37	74.72	**83.49**	69.32	75.75	**84.77**	79.43	82.01	**89.28**	**89.77**	**89.52**
Gold	69.52	75.41	72.35	76.70	**82.14**	**79.33**	81.32	**85.39**	**83.31**			
Annotator-Aware												
LC	78.09	74.10	76.04	79.98	77.18	78.55	77.72	81.06	79.36	87.42	85.64	86.52
LC-cat	75.37	78.54	76.92	74.24	81.32	77.62	76.88	81.37	78.96	88.25	86.03	87.13
This Work	**80.06**	**81.91**	**80.97**	**83.25**	**85.36**	**84.29**	**85.19**	**87.46**	**86.31**	**89.62**	**90.51**	**90.06**

Table 2: The test results of the supervised setting, where we add different proportions of the most informative gold-standard (expert) annotations incrementally. Note that MV at 100% is equivalent to the gold model, because all voted labels are substituted with gold-standard labels.

Notice that ALL and MV are annotator-agnostic models, which exploit no information specific to the individual annotators, while the other three models are all annotator-aware models, where the annotator information is used by different ways.

Hyper-parameters We offer all detailed settings of Hyper-parameters in Appendix B.

4.2 Unsupervised Results

Table 1 shows the test results of the unsupervised setting. As a whole, we can see that our representation learning model (i.e., This Work) borrowed from domain adaptation can achieve the best performance, resulting in an F1 score of 77.95, significantly better than the second-best model LC-cat (i.e., $77.95 - 76.79 = 1.16$). The result indicates the advantage of our method over the other models.

By examining the results in-depth, we can find that the annotator-aware model is significantly better than the annotator-agnostic models, demonstrating that the annotator information is highly helpful for crowdsourcing learning. The observation further shows the reasonableness by aligning annotators to domains, since domain information is also useful for domain adaptation. In addition, the better performance of our representation learning method among the annotator-aware models indicates that our model can capture annotator-aware information more effectively because our start point is totally different. We do not attempt to model the expert labels based on crowdsourcing annotations.

Further, we observe that several models show better precision values, while others give better recall values. A high precision but low recall indicates that the model is conservative in detecting named entities, and vice the reverse. Our proposed model is able to balance the two directions better, with the least gap between them. Also, the results imply that there is still much space for future development, and the recent advances of domain adaptation might offer good avenues.

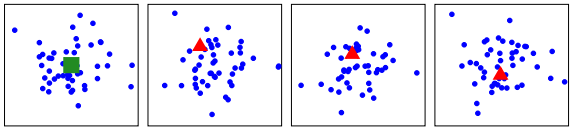
Finally, we compare our results with previous studies. As shown, our model can obtain the best performance in the literature. In particular, by comparing our results with the original performances reported in Nguyen et al. (2017), we can see that our re-implementation is much better than theirs. The major difference lies in the exploration of BERT in our model, which brings improvements closed to 6% for both LC and LC-cat.

4.3 Supervised Results

To investigate the supervised setting, we assume that expert annotations (ground truths) of all crowdsourcing sentences are available. Besides exploring the full expert annotations, we study another three different scenarios by incrementally adding the expert annotations into the unsupervised setting, aiming to study the effectiveness of our model with small expert annotations as well. Concretely, we assume proportions of 1%, 5%, 25%, and 100% of the expert annotations available.[4] Table 2 shows all the results, including our four baselines and an gold model based on only expert annotations for comparisons. Overall, we can see that our representation learning model can bring the best performances for all scenarios, demonstrating its effectiveness in the supervised learning as well.

Next, by comparing annotator-agnostic and annotator-aware models, we can see that annotator-aware models are better, which is consistent with

[4]Intuitively, if expert annotations are involved, we should intentionally choose the more informative inputs for annotations, which can reduce the overall cost to meet a certain performance standard. Thus, we can fully demonstrate the effectiveness of crowdsourced annotations under the semi-supervised setting. Here we try to choose the most informative labeled instances for the 1%, 5%, and 25% settings.



(a) 0% (b) 5% (c) 25% (d) 100%

Figure 4: The visualization of annotator embeddings by dimensionality reduction with PCA. Out designed unsupervised (0%) expert is consistent with the well-learned one (100%). With the expert annotations increases, the learned expert becomes more accurate.

the unsupervised setting. More interestingly, the results show that `All` is better than `gold` with very small-scale expert annotations (1% and 5%), and the tendency is reversed only when there are sufficient expert annotations (25% and 100%). The observation indicates that crowdsourced annotations are always helpful when golden annotations are not enough. In addition, it is easy to understand that `MV` is worse than `gold` since the latter has a higher-quality of the training corpus.

Further, we can find that even the annotator-aware `LC` and `LC-cat` models are unable to obtain any positive influence compared with `gold`, which demonstrates that distilling ground-truths from the crowdsourcing annotations might not be the most promising solution. While our representation learning model can give consistently better results than `gold`, indicating that crowdsourced annotations are always helpful by our method. By regarding crowdsourcing learning as domain adaptation, we no longer take crowdsourced annotations as noise, and on the contrary, they are treated as transferable knowledge, similar to the relationship between the source domains and the target domain. Thus they could always be useful in this way.

4.4 Analysis

To better understand our idea and model in-depth, we conducted the following fine-grained analyses.[5]

Visualization of Annotator Embeddings Our representation learning model is able to learn annotator embeddings through the task objective. It is interesting to visualize these embeddings to check their distributions, which can reflect the relationships between the individual annotators. Figure 4 shows the visualization results after Principal Component Analysis (PCA) dimensionality reduction,

[5]In addition, we could not perform the ablation study of our model because it is not an incremental work.

Model	P	R	F1	Gold(5%)
`ALL`	67.02	69.31	68.15	
MV	72.24	69.49	70.88	
`LC`	72.34	70.48	71.35	**79.33**
`LC-cat`	**72.76**	**71.78**	**72.26**	
This Work	**80.78**	**73.78**	**77.12**	

Table 3: The performance of training on 85% and testing on 15% of the crowdsourced annotations.

where the unsupervised and three supervised scenarios are investigated.[6] As shown, we can see that most crowd annotators are distributed in a concentrated area for all scenarios, indicating that they are able to share certain common characteristics of task understanding.

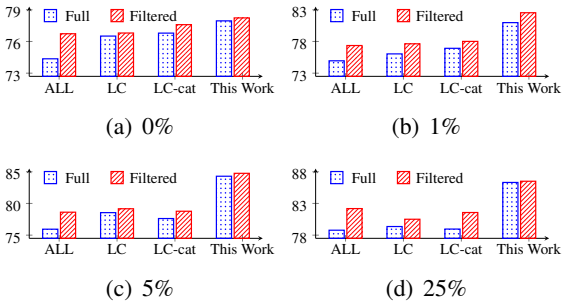
Further, we focus on the relationship between expert and crowd annotators, and the results show two interesting findings. First, the heuristic expert of our unsupervised learning is almost consistent with that of the supervised learning of the whole expert annotations (100%), which indicates that our unsupervised expert estimation is perfectly good. Second, the visualization shows that the relationship between expert and crowd annotators could be biased when expert annotations are not enough. As the size of expert annotations increases, their connection might be more accurate gradually.

The Predictability of Crowdsourcing Annotations Our primary assumption is based on that all crowdsourced annotations are regarded as the gold-standard with respect to the crowd annotators, which naturally indicates that these annotations are predictable. Here we conduct analysis to verify the assumption by a new task to predicate the crowdsourced annotations, Concretely, we divide the annotations into two sections, where 85% of them are used as the training and the remaining are used for testing, and then we apply our baseline and proposed models to learn and evaluate.

Table 3 shows the results. As shown, our model can achieve the best performance by an F1 score of 77.12%, and the other models are significantly worse (at least 4.86 drops by F1). Considering that the proportion of the averaged training examples per annotator over the full 5,985 sentences is only 5%,[7] we exploit the `gold` model of the 5% expert annotations for reference. We can see that the gap between them is small (77.12% v.s. 79.33%),

[6]The 1% setting is excluded for its incapability to capture the relationship between the expert and crowd annotators with such small expert annotations.

[7]The value can be directly calculated ($0.06 * 0.85 \approx 0.05$).



(a) 0% (b) 1%

(c) 5% (d) 25%

Figure 5: Comparisons by F1 scores between full and filtered crowdsourced annotations (i.e., excluding unreliable annotators). We compute F1 values of each annotator with respect to the gold-standard labels, and filter out 10 annotators with lowest scores.

which indicates that our assumption is acceptable as a whole. The other models could be unsuitable for our assumption due to the poor performance induced by their modeling strategies.

The Impact of Unreliable Annotators Handling unreliable annotators, such as spammers, is a practical and common issue in Crowdsourcing (Raykar and Yu, 2012b). Obviously, regarding crowd annotations as untrustworthy answers is more considerate to this problem. In contrast, our assumption might be challenged because these unreliable annotators are discrepant in their own annotations. To show the influence of unreliable annotators, we filter out several unreliable annotators in the corpus, and reevaluate the performance for the low-resource supervised and unsupervised scenarios on the remaining annotations.

Figure 5 shows the comparison results of the original corpus and the filtered corpus.[8] First, we can find that improved performance can be achieved in all cases, indicating excluding these unreliable annotations is helpful for crowdsourcing. Second, the `LC` and `LC-cat` model give smaller score differences compared with the `ALL` model between these two kinds of results, which verified that they are considerate to unreliable annotators. Third, our model also performs robustly, it can cope with this practical issue in a certain degree as well.

Results on The Sampled Annotators and Annotations The above analysis shows the benefit of removing unreliable annotators, which reduces a small number of annotators and annotations. A problem arises naturally: will the performance be consistent if we sample a small proportion of annotators? To verify it, we sampled two sub-set from the crowdsourced training corpus and re-train our model as well as baselines. Table 4 shows the evaluation results of re-trained models on the standard test set in unsupervised setting. We also add our main result for the comparison. As shown, all sampled datasets demonstrate similar trends with the main result (denoted as `Full`). The supervised results are consistent with our main result as well, which are not listed due to space reasons.

[8] `MV` is not included because a proportion of instances are unable to obtain aggregated answers.

Data	Full	Excluded	Part-1	Part-2
Model	F1			
`ALL`	74.36	76.73	74.66	75.92
`LC`	76.51	76.80	75.29	76.70
`LC-cat`	76.79	77.59	74.86	76.02
This Work	77.95	78.23	77.41	77.58

Table 4: The unsupervised test results of differently sampled datasets. The Full is original results in Table 1. The Excluded is the filtered corpus in Figure 5. The Part-1 and Part-2 are both consist of 13 annotators. Part-1 have 1800 texts with 6275 crowd annotations, each text is labeled by at least 3 annotators. These numbers of Part-2 are 2192, 5582, and 2, respectively.

4.5 The Discussion of Domain Definitions

The most widely used definition of a domain is the distribution on the input space $\mathcal{X}$. Zhao et al. (2019) define a domain D as the pair of a distribution $\mathcal{D}$ on the input space $\mathcal{X}$ and a labeling function $f : \mathcal{X} \to \mathcal{Y}$, i.e., domain $D = \langle \mathcal{D}, f \rangle$.

In this work, we assume each annotator is a unique labeling function $a : \mathcal{X} \to \mathcal{Y}$. Uniting each annotator and the instances he/she labeled, we can result in a number of domains $\{\langle \mathcal{D}_i, a_i \rangle\}_{i=1}^{|A|}$, where A represents all annotators. Then the crowdsourcing learning can be interpreted by the later definition, i.e., learning from these crowd annotators/domains and predicting the labels of raw inputs (sampled from the raw data distribution $\mathcal{D}_{\text{expert}}$) in expert annotator/domain $\langle \mathcal{D}_{\text{expert}}, \text{expert} \rangle$. To unify the definition in a single distribution, we directly define a domain as the joint distribution on the input space $\mathcal{X}$ and the label space $\mathcal{Y}$.

In addition, we can align to the former definition by using the representation outputs $\boldsymbol{x}^i = a_i(x)$ as the data input, which shows different distributions for the same sentence towards different annotators. Thus, each source domain D_i is the distibution of $\boldsymbol{x}^i$, and we need learn the expert representations $\boldsymbol{x}^{\text{expert}}$ to perform inference on the unlabled texts.

5 Related Work

5.1 Crowdsourcing Learning

Crowdsourcing is a cheap and popular way to collect large-scale labeled data, which can facilitate the model training for hard tasks that require supervised learning (Wang and Zhou, 2016; Sheng and Zhang, 2019). In particular, crowdsourced data is often regarded as low-quality, including much noise regarding expert annotations as the gold-standard. Initial studies of crowdsourcing learning try to arrive at a high-quality corpus by majority voting or control the quality by sophisticated strategies during the crowd annotation process (Khattak and Salleb-Aouissi, 2011; Liu et al., 2017; Tang and Lease, 2011).

Recently, the majority work focuses on full exploration of all annotated corpus by machine learning models, taking the information from crowd annotators into account including annotator reliability (Rodrigues et al., 2014), annotator accuracy (Huang et al., 2015), worker-label confusion matrix (Nguyen et al., 2017), and sequential confusion matrix (Simpson and Gurevych, 2019).

In this work, we present a totally different viewpoint for crowdsourcing, regarding all crowdsourced annotations as golden in terms of individual annotators, just like the primitive gold-standard labels corresponded to the experts, and further propose a domain adaptation paradigm for crowdsourcing learning.

5.2 Domain Adaptation

Domain adaptation has been studied extensively to reduce the performance gap between the resource-rich and resource-scarce domains (Ben-David et al., 2006; Mansour et al., 2009), which has also received great attention in the NLP community (Daumé III, 2007; Jiang and Zhai, 2007; Finkel and Manning, 2009; Glorot et al., 2011; Chu and Wang, 2018; Ramponi and Plank, 2020). Typical methods include self-training to produce pseudo training instances for the target domain (Yu et al., 2015) and representation learning to capture transferable features across the source and target domains (Sener et al., 2016).

In this work, we make correlations between domain adaptation and crowdsourcing learning, enabling crowdsourcing learning to benefit from the advances of domain adaptation, and then present a representation learning model borrowed from Jia et al. (2019) and Üstün et al. (2020).

5.3 Named Entity Recognition

NER is a fundamental and challenging task of NLP (Yadav and Bethard, 2018). The BiLSTM-CRF (Lample et al., 2016) architecture, as well as BERT (Devlin et al., 2019), are able to bring state-of-the-art performance in the literature (Jia et al., 2019; Wang et al., 2020; Jia and Zhang, 2020). Mayhew et al. (2020) exploits the BERT-BiLSTM-CRF model, achieving strong performance on NER.

In addition, NER has been widely adopted as crowdsourcing learning as well (Finin et al., 2010; Rodrigues et al., 2014; Derczynski et al., 2016; Yang et al., 2018). Thus, we exploit NER as a case study following these works, and take a BERT-BiLSTM-CRF model as the basic model for our annotator-aware extension.

6 Conclusion and Future Work

We studied the connection between crowdsourcing learning and domain adaptation, and then proposed to treat crowdsourcing learning as a domain adaptation problem. Following, we took NER as a case study, suggesting a representation learning model from recent advances of domain adaptation for crowdsourcing learning. By this case study, we introduced unsupervised and supervised crowdsourcing learning, where the former is a widely-studied setting while the latter has been seldom investigated. Finally, we conducted experiments on a widely-adopted benchmark dataset for crowdsourcing NER, and the results show that our representation learning model is highly effective in unsupervised learning, achieving the best performance in the literature. In addition, the supervised learning with a very small scale of expert annotations can boost the performance significantly.

Our work sheds light on the application of effective domain adaptation models on crowdsourcing learning. There are still many other sophisticated cross-domain models, such as adversarial learning (Ganin et al., 2016) and self-training (Yu et al., 2015). Future work may include how to apply these advances to crowdsourcing learning properly.

Acknowledgments

We thank all reviewers for their hard work. This research is supported by grants from the National Key Research and Development Program of China (No. 2018YFC0832101) and the fonds of Beijing Advanced Innovation Center for Language Resources under Grant TYZ19005.

Ethical Impact

We present a different view of crowdsourcing learning and propose to treat it as domain adaptation, showing the connection between these two topics of machine learning for NLP. In this view, many sophisticated cross-domain models could be applied to crowdsourcing learning. Moreover, the motivation that regarding all crowdsourced annotations as gold-standard to the corresponding annotators, also sheds light on introducing other transfer learning techniques in future work.

The above idea and our proposed representation learning model for crowdsourcing sequence labeling, are totally agnostic to any private information of annotators. And we do not use any sensitive information, bu only the ID of annotators, in problem modeling and learning. The crowdsourced CoNLL English NER data also anonymized annotators. There will be no privacy issues in the future.

References

Azad Abad, Moin Nabi, and Alessandro Moschitti. 2017. Self-crowdsourcing training for relation extraction. In *Proceedings of the ACL: Short Papers*.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, pages 137–144. MIT Press.

Chris Callison-Burch and Mark Dredze. 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*, pages 1–12.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of COLING*, pages 1304–1319.

Gabriela Csurka. 2017. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of ACL*, pages 256–263.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad Twitter corpus: A diverse named entity recognition resource. In *Proceedings of the COLING: Technical Papers*, pages 1169–1179.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Eraldo R. Fernandes and Ulf Brefeld. 2011. Learning from partially annotated sequences. In *ECML-PKDD*, volume 6911 of *Lecture Notes in Computer Science*, pages 407–422. Springer.

Tim Finin, William Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in Twitter data with crowdsourcing. In *Proceedings of the NAACL-HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Jenny Rose Finkel and Christopher D. Manning. 2009. Hierarchical Bayesian domain adaptation. In *Proceedings of HLT-NAACL*, pages 602–610.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of ICML*, pages 513–520.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of ICML*, pages 2790–2799.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the NAACL-HLT*.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. Experiments with crowdsourced re-annotation of a POS tagging data set. In *Proceedings of ACL*.

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. 2009. Data quality from crowdsourcing: A study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35.

Fei Huang and Alexander Yates. 2010. Exploring representation-learning approaches to domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*.

Ziheng Huang, Jialu Zhong, and Rebecca J. Passonneau. 2015. Estimation of discourse segmentation labels from crowd data. In *Proceedings of the EMNLP*, pages 2190–2200.

Emily Jamison and Iryna Gurevych. 2015. Noise or additional information? leveraging crowdsource annotation item agreement for natural language tasks. In *Proceedings of the EMNLP*, pages 291–297.

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-domain NER using cross-domain language modeling. In *Proceedings of ACL*, pages 2464–2474.

Chen Jia and Yue Zhang. 2020. Multi-cell compositional LSTM for NER domain adaptation. In *Proceedings of ACL*, pages 5906–5917.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of ACL*, pages 264–271.

Faiza Khan Khattak and Ansaf Salleb-Aouissi. 2011. Quality control of crowd labeling through expert evaluation. In *Proceedings of the NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds*, volume 2, page 5.

Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, 2008, Florence, Italy, April 5-10, 2008*, pages 453–456. ACM.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Matthew Lease. 2011. On quality control and machine learning in crowdsourcing. In *Human Computation*, volume WS-11-11 of *AAAI Workshops*. AAAI.

Maolin Li, Hiroya Takamura, and Sophia Ananiadou. 2020. A neural model for aggregating coreference annotation in crowdsourcing. In *Proceedings of COLING*, pages 5760–5773.

Mengchen Liu, Liu Jiang, Junlin Liu, Xiting Wang, Jun Zhu, and Shixia Liu. 2017. Improving learning-from-crowds through expert validation. In *Proceedings of IJCAI*, pages 2329–2336.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, volume 21, pages 1041–1048.

Stephen Mayhew, Nitish Gupta, and Dan Roth. 2020. Robust named entity recognition with truecasing pretraining. In *AAAI 2020*, pages 8480–8487.

An Thanh Nguyen, Byron Wallace, Junyi Jessy Li, Ani Nenkova, and Matthew Lease. 2017. Aggregating and predicting sequence labels from crowd annotations. In *Proceedings of ACL*, pages 299–309.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain Marshall, Ani Nenkova, and Byron Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the ACL*, pages 197–207.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In *Proceedings of EMNLP*.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the COLING*, pages 6838–6855.

Vikas C. Raykar and Shipeng Yu. 2012a. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *J. Mach. Learn. Res.*, 13:491–518.

Vikas C. Raykar and Shipeng Yu. 2012b. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *J. Mach. Learn. Res.*, 13:491–518.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *J. Mach. Learn. Res.*, 11:1297–1322.

Filipe Rodrigues and Francisco C. Pereira. 2018. Deep learning from crowds. In *Proceedings of the AAAI*.

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

Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. 2016. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*.

Victor S. Sheng, Foster J. Provost, and Panagiotis G. Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the KDD*, pages 614–622.

Victor S. Sheng and Jing Zhang. 2019. Machine learning with crowdsourcing: A brief summary of the past research and future directions. *Proceedings of the AAAI*, 33(01):9837–9843.

Edwin Simpson and Iryna Gurevych. 2019. A Bayesian approach for sequence tagging with crowds. In *Proceedings of the EMNLP-IJCNLP*.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the EMNLP*.

Wei Tang and Matthew Lease. 2011. Semi-supervised consensus labeling for crowdsourcing. In *SIGIR 2011 workshop on crowdsourcing for information retrieval (CIR)*, pages 1–6.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the CoNLL at HLT-NAACL 2003*.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the EMNLP*, pages 2302–2315.

Jing Wang, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2020. Multi-domain named entity recognition with genre-aware and agnostic inference. In *Proceedings of the ACL*, pages 8476–8488.

Lu Wang and Zhi-Hua Zhou. 2016. Cost-saving effect of crowdsourcing learning. In *Proceedings of IJCAI*, IJCAI'16, pages 2111–2117.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the EMNLP: System Demonstrations*, pages 38–45.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the COLING*.

YaoSheng Yang, Meishan Zhang, Wenliang Chen, Wei Zhang, Haofen Wang, and Min Zhang. 2018. Adversarial learning for chinese NER from crowd annotations. In *Proceedings of the AAAI*.

Juntao Yu, Mohab Elkaref, and Bernd Bohnet. 2015. Domain adaptation for dependency parsing via self-training. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 1–10.

Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the ACL-HLT*, pages 1220–1229.

Jing Zhang, Xindong Wu, and Victor S. Sheng. 2016. Learning from crowdsourced labeled data: a survey. *Artif. Intell. Rev.*, 46(4):543–576.

Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. 2019. On learning invariant representations for domain adaptation. In *Proceedings of the ICML*, pages 7523–7532. PMLR.

A Transformer with Adapters

In our Adapter ∘ BERT word representation, we insert two adapter modules for each transformer layer inside BERT. Figure 6 shows the detailed network structure of transformer with adapters. More specifically, the forward operation of an adapter layer is computed as follows:

$$\begin{aligned} \boldsymbol{h}_{\text{mid}} &= \text{GELU}(\boldsymbol{W}_1^{\text{ap}}\boldsymbol{h}_{\text{in}} + \boldsymbol{b}_1^{\text{ap}}) \\ \boldsymbol{h}_{\text{out}} &= \boldsymbol{W}_2^{\text{ap}}\boldsymbol{h}_{\text{mid}} + \boldsymbol{b}_2^{\text{ap}} + \boldsymbol{h}_{\text{in}}, \end{aligned} \quad (6)$$

where $\boldsymbol{W}_1^{\text{ap}}$, $\boldsymbol{W}_2^{\text{ap}}$, $\boldsymbol{b}_1^{\text{ap}}$ and $\boldsymbol{b}_2^{\text{ap}}$ are adapter parameters, and the dimension size of $\boldsymbol{h}_{\text{mid}}$ is usually smaller than that of the corresponding transformer.

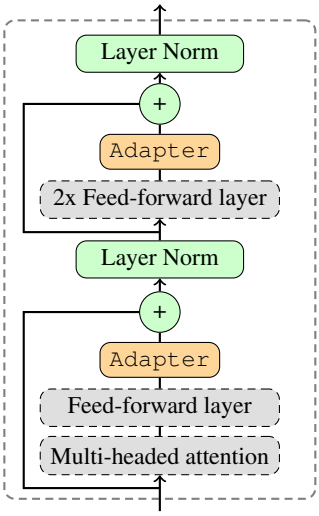


Figure 6: Transformer integrated with Adapters inside.

Model	ALL	MV	Gold	Trainable Params Size
FineTuning	74.12	74.96	89.32	**108M**
BERT with Adapter Inside				
2 layers	71.83	73.81	89.20	4.55M
4 layers	73.16	73.30	89.26	5.34M
6 layers	73.74	74.81	89.33	6.14M
8 layers	74.24	**75.31**	89.13	6.94M
10 layers	**74.56**	75.01	89.21	7.73M
All layers	74.36	75.28	**89.52**	8.53M

Table 5: The comparisons between BERT fine-tuning and Adapter ∘ BERT based on the standard NER without annotator as input.

Here we also give a supplement to illustrate the pack operation from all adapter parameters into a single vector $\boldsymbol{V}$:

$$\boldsymbol{V} = \bigoplus_{\text{Adapters}} \{\boldsymbol{W}_1^{\text{ap}} \oplus \boldsymbol{W}_2^{\text{ap}} \oplus \boldsymbol{b}_1^{\text{ap}} \oplus \boldsymbol{b}_2^{\text{ap}}\}, \quad (7)$$

where first all parameters of a single adapter are reshaped and concatenated and then a further concatenation is performed over all adapters.

B Hyper-parameters

We choose the BERT-base-cased[9], which is for English language and consists of 12-layer transformers with the hidden size 768 for all layers. We load the BERT weight and implement the adapter injection based on the transformers (Wolf et al., 2020) library. The sizes of the adapter middle hidden states are set to 128 constantly. The annotator embedding size is 8 to fit the model in one RTX-2080TI GPU of 11GB memory. The BiLSTM hidden size is

[9]https://github.com/google-research/bert

Model	Text and Entities
Unsupervised	
MV	Pace, a junior, helped [**Ohio State**]$_{LOC}$ to a 10-1 record and a berth in the Rose Bowl against [Arizona]$_{ORG}$ State.
LC-cat	Pace, a junior, helped [**Ohio State**]$_{ORG}$ to a 10-1 record and a berth in the [**Rose Bowl**]$_{MISC}$ against [Arizona]$_{ORG}$ State.
This Work	Pace, a junior, helped [**Ohio State**]$_{ORG}$ to a 10-1 record and a berth in the [**Rose Bowl**]$_{MISC}$ against [**Arizona State**]$_{ORG}$.
Supervised (25%)	
MV	Pace, a junior, helped [**Ohio State**]$_{LOC}$ to a 10-1 record and a berth in the [**Rose Bowl**]$_{MISC}$ against [**Arizona State**]$_{LOC}$.
Gold	[**Pace**]$_{PER}$, a junior, helped [**Ohio State**]$_{ORG}$ to a 10-1 record and a berth in the [**Rose Bowl**]$_{MISC}$ against [Arizona]$_{ORG}$ State.
LC-cat	Pace, a junior, helped [**Ohio State**]$_{ORG}$ to a 10-1 record and a berth in the [**Rose Bowl**]$_{MISC}$ against [**Arizona State**]$_{LOC}$.
This Work	[**Pace**]$_{PER}$, a junior, helped [**Ohio State**]$_{ORG}$ to a 10-1 record and a berth in the [**Rose Bowl**]$_{MISC}$ against [**Arizona State**]$_{ORG}$.
Ground-truth	[**Pace**]$_{PER}$, a junior, helped [**Ohio State**]$_{ORG}$ to a 10-1 record and a berth in the [**Rose Bowl**]$_{MISC}$ against [**Arizona State**]$_{ORG}$.

Table 6: A case study, where the text with underlines indicates errors.

set to 400. For all models, we inject adapters or switchers in all 12 layers of BERT. All experiments are run on the single GPU at an 8-GPU server with a 14 core CPU and 128GB memory.

We exploit the stochastic gradient-based online learning, with a batch size of 64, to optimize model parameters. We apply the time-step dropout, which randomly sets several representations in the sequence to zeros with a probability of 0.2, on the word representations to avoid overfitting. We use the Adam algorithm to update the parameters with a constant learning rate 1×10^{-3}, and apply the gradient clipping by a maximum value of 5.0 to avoid gradient explosion.

C The Advantage of Adapter ∘ BERT

Our models are all based on Adapter ∘ BERT as the basic representations, which is different from the widely-adopted BERT fine-tuning architecture. Here we compare the two strategies in detail. The results are shown in Table 5, where for Adapter ∘ BERT we consider gradually increasing the number of transformer layers (covering the last n layers) inside the BERT. As shown, it is apparently that Adapter ∘ BERT is much more parameter efficient, and when all layers are exploited, the model can be even better than BERT fine-tuning. Thus it is more desirable to use Adapter ∘ BERT covering all BERT transformers inside.

D Case Study

Here we also offer a case study to understand the performance in unsupervised and supervised crowdsourcing learning, as well as the different crowdsourcing models. We exploit one complex example in Table 6 which involves different outputs for various models. As shown, we can see that supervised models are able to recall the ambiguous entity (i.e., Pace, a single word with multiple senses) correctly, while unsupervised models fail, which may be due to the inconsistencies of the crowdsourced annotations. By comparing our model with other baselines, we can show that our representation learning model can capture the global text input understanding consistently, e.g., being able to connect Ohio State and Arizona State together.