Cross-Domain NER using Cross-Domain Language Modeling

Chen Jia†‡ , **Xiaobo Liang**◇[*] and **Yue Zhang**‡§
†Fudan University, China
‡School of Engineering, Westlake University, China
◇Natural Language Processing Lab, Northeastern University, China
§Institute of Advanced Technology, Westlake Institute for Advanced Study
{jiachen,zhangyue}@westlake.edu.cn, liangxiaobo0309@gmail.com

Abstract

Due to limitation of labeled resources, cross-domain named entity recognition (NER) has been a challenging task. Most existing work considers a supervised setting, making use of labeled data for both the source and target domains. A disadvantage of such methods is that they cannot train for domains without NER data. To address this issue, we consider using cross-domain LM as a bridge cross-domains for NER domain adaptation, performing cross-domain and cross-task knowledge transfer by designing a novel parameter generation network. Results show that our method can effectively extract domain differences from cross-domain LM contrast, allowing unsupervised domain adaptation while also giving state-of-the-art results among supervised domain adaptation methods.

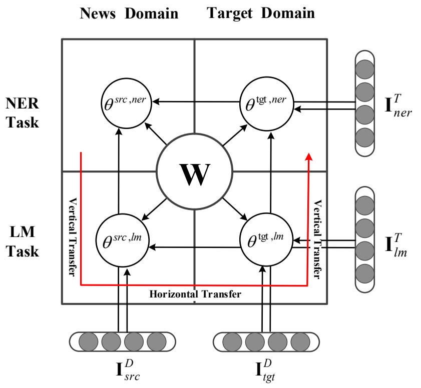

Figure 1: Overview of the proposed model.

1 Introduction

Named entity recognition (NER) is a fundamental task in information extraction and text understanding. Due to large variations in entity names and flexibility in entity mentions, NER has been a challenging task in NLP. Cross-domain NER adds to the difficulty of modeling due to the difference in text genre and entity names. Existing methods make use of feature transfer (Daumé III, 2009; Kim et al., 2015; Obeidat et al., 2016; Wang et al., 2018) and parameters sharing (Lee et al., 2017; Sachan et al., 2018; Yang et al., 2017; Lin and Lu, 2018) for supervised NER domain adaptation.

Language modeling (LM) has been shown useful for NER, both via multi-task learning (Rei, 2017) and via pre-training (Peters et al., 2018). Intuitively, both noun entities and context patterns can be captured during LM training, which benefits the recognition of named entities. A natural question that arises is whether cross-domain LM training can benefit cross-domain NER. Figure 1 shows one example, where there are relatively large training data in the news domain but no data or a small amount of data in a target domain. We are interested in transferring NER knowledge from the news domain to the target domain by contrasting large raw data in both domains through cross-domain LM training.

Naive multi-task learning by parameter sharing (Collobert and Weston, 2008) does not work effectively in this multi-task, multi-domain setting due to potential conflict of information. To achieve cross-domain information transfer as shown in the red arrow, two types of connections must be made: (1) cross-task links between NER and LM (for vertical transfer) and (2) cross-domain links (for horizontal transfer). We investigate a novel parameter generator network to this end, by decomposing the parameters θ of the NER or LM task on the source or target text domain into the combination $\theta = f(\mathbf{W}, \mathbf{I}^D_d, \mathbf{I}^T_t)$ of a set of meta parameters $\mathbf{W}$, a task embedding vector $\mathbf{I}^T_t$ $(t \in \{ner, lm\})$ and a domain embedding vector $\mathbf{I}^D_d$ $(d \in \{src, tgt\})$, so that domain and task-correlations can be learned through similarities between the respective domain and task embedding vectors.

[*] Work done when visiting Westlake University.

In Figure 1, the values of $\mathbf{W}$, $\{\mathbf{I}_t^T\}$, $\{\mathbf{I}_d^D\}$ and the parameter generation network $f(\cdot,\cdot,\cdot)$ are all trained in a multi-task learning process optimizing NER and LM training objectives. Through the process, connections between the sets of parameters $\theta^{src,ner}$, $\theta^{src,lm}$, $\theta^{tgt,ner}$ and $\theta^{tgt,lm}$ are decomposed into two dimensions and distilled into two task embedding vectors $\mathbf{I}_{ner}^T$, $\mathbf{I}_{lm}^T$ and two domain embedding vectors $\mathbf{I}_{src}^D$, $\mathbf{I}_{tgt}^D$, respectively. Compared with traditional multi-task learning, our method has a modular control over cross-domain and cross-task knowledge transfer. In addition, the four embedding vectors $\mathbf{I}_{ner}^T$, $\mathbf{I}_{lm}^T$, $\mathbf{I}_{src}^D$ and $\mathbf{I}_{tgt}^D$ can also be trained by optimizing on only three datasets for $\theta^{src,ner}$, $\theta^{src,lm}$ and $\theta^{tgt,lm}$, therefore achieving zero-shot NER learning on the target domain by deriving $\theta^{tgt,ner}$ automatically.

Results on three different cross-domain datasets show that our method outperforms naive multi-task learning and a wide range of domain adaptation methods. To our knowledge, we are the first to consider unsupervised domain adaptation for NER via cross-domain LM tasks and the first to work on NER transfer learning between domains with completely different entity types (i.e. news vs. biomedical). We released our data and code at https://github.com/jiachenwestlake/Cross-Domain_NER.

2 Related Work

NER. Recently, neural networks have been used for NER and achieved state-of-the-art results. Hammerton (2003) use a unidirectional LSTM with a Softmax classifer. Collobert et al. (2011) use a CNN-CRF architecture. Santos and Guimarães (2015) extend the model by using character CNN. Most recent work uses LSTM-CRF (Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016; Yang et al., 2018). We choose BiLSTM-CRF as our method since it gives state-of-the-art resutls on standard benchmarks.

Cross-domain NER. Most existing work on cross-domain NER investigates the supervised setting, where both source and target domains have labeled data. Daumé III (2009) maps entity label space between the source and target domains. Kim et al. (2015) and Obeidat et al. (2016) use label embeddings instead of entities themselves as the features for cross-domain transfer. Wang et al. (2018) perform label-aware feature representation transfer based on text representation learned by BiLSTM networks.

Recently, parameters transfer approaches have seen increasing popularity for cross-domain NER. Such approaches first initialize a target model with parameters learned from source-domain NER (Lee et al., 2017) or LM (Sachan et al., 2018), and then fine-tune the model using labeled NER data from the target domain. Yang et al. (2017) jointly train source- and target-domain models with shared parameters, Lin and Lu (2018) add adaptation layers on top of existing networks. Except for Sachan et al. (2018), all the above methods use cross-domain NER data only. In contrast, we leverage both NER data and raw data for both domains. In addition, our method can deal with a zero-shot learning setting for unsupervised NER domain adaptation, which no existing work considers.

Learning task embedding vectors. There has been related work using task vector representations for multi-task learning. Ammar et al. (2016) learn language embeddings for multi-lingual parsing. Stymne et al. (2018) learn treebank embeddings for cross-annotation-style parsing. These methods use "task" embeddings to augment word embedding inputs, distilling "task" characteristics into these vectors for preserving word embeddings. Liu et al. (2018) learn domain embeddings for multi-domain sentiment classification. They combine domain vectors with domain-independent representation of the input sentences to obtain a domain-specific input representation. A salient difference between our work and the methods above is that we use domain and task embeddings to obtain domain and task-specific parameters, rather than input representations.

Closer in spirit to our work, Platanios et al. (2018) learn language vectors, using them to generate parameters for multi-lingual machine translation. While one of their main motivation is to save the parameter space when the number of langauges grows, our main goal is to investigate the modularization of transferable knowledge in a cross-domain and cross-task setting. To our knowledge, we are the first to study "task" embeddings in a multi-dimensional parameter decomposition setting (e.g. domain + task).

3 Methods

The overall structure of our proposed model is shown in Figure 2. The bottom shows the com-

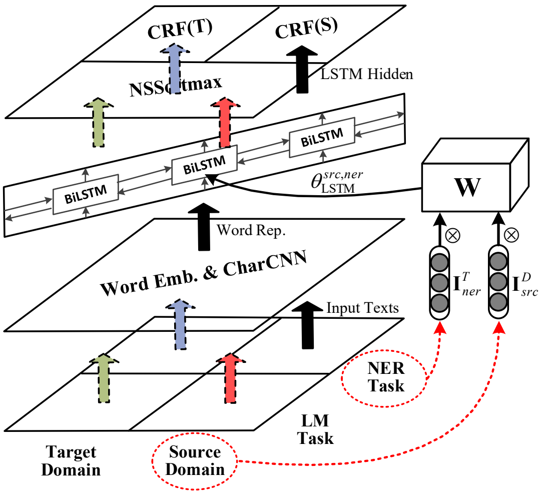

Figure 2: Model architecture.

bination of two domains and two tasks. Given an input sentence, word representations are first calculated through a shared embedding layer (Subsection 3.1). Then a set of task- and domain-specific BiLSTM parameters is calculated through a novel parameter generation network (Subsection 3.2), for encoding the input sequence. Finally, respective output layers are used for different tasks and domains (Subsection 3.3).

3.1 Input Layer

Following Yang et al. (2018), given an input $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ from a source-domain NER training set $\mathcal{S}_{ner} = \{(x_i, y_i)\}_{i=1}^{m}$ or target-domain NER training set $\mathcal{T}_{ner} = \{(x_i, y_i)\}_{i=1}^{n}$, a source-domain raw text set $\mathcal{S}_{lm} = \{(x_i)\}_{i=1}^{p}$ or target-domain raw text set $\mathcal{T}_{lm} = \{(x_i)\}_{i=1}^{q}$, each word x_i is represented as the concatenation of its word embedding and the output of a character level CNN :

$$\mathbf{v}_i = [\mathbf{e}^w(x_i) \oplus \text{CNN}(\mathbf{e}^c(x_i))], \tag{1}$$

where $\mathbf{e}^w$ represents a shared word embedding lookup table and $\mathbf{e}^c$ represents a shared character embedding lookup table. $\text{CNN}(\cdot)$ represents a standard CNN acting on a character embedding sequence $\mathbf{e}^c(x_i)$ of a word x_i. $\oplus$ represents vector concatenation.

3.2 Parameter Generation Network

A bi-directional LSTM layer is applied to $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$.

To transfer knowledge across domains and tasks, we dynamically generate the parameters of BiLSTM using a **Parameter Generation Network** ($f(\cdot,\cdot,\cdot)$). The resulting parameters are denoted as $\theta_{\text{LSTM}}^{d,t}$, where $d \in \{src, tgt\}$ and $t \in \{ner, lm\}$ represent domain label and task label, respectively. More specifically:

$$\theta_{\text{LSTM}}^{d,t} = \mathbf{W} \otimes \mathbf{I}_d^D \otimes \mathbf{I}_t^T, \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{P^{(\text{LSTM})} \times V \times U}$ represents a set of meta parameters in the form of a 3rd-order tensor and $\mathbf{I}_d^D \in \mathbb{R}^U$, $\mathbf{I}_t^T \in \mathbb{R}^V$ represent domain embedding and task embedding, respectively. U, V represent domain and task embedding sizes, respectively. $P^{(\text{LSTM})}$ is the number of BiLSTM parameters. $\otimes$ refers to tensor contraction.

Given the input $\mathbf{v}$ and the parameter $\theta_{\text{LSTM}}^{d,t}$, the hidden outputs of a task and domain-specific BiLSTM unit can be uniformly written as:

$$\begin{aligned} \overrightarrow{\mathbf{h}}_i^{d,t} &= \text{LSTM}(\overrightarrow{\mathbf{h}}_{i-1}^{d,t}, \mathbf{v}_i, \overrightarrow{\theta}_{\text{LSTM}}^{d,t}) \\ \overleftarrow{\mathbf{h}}_i^{d,t} &= \text{LSTM}(\overleftarrow{\mathbf{h}}_{i+1}^{d,t}, \mathbf{v}_i, \overleftarrow{\theta}_{\text{LSTM}}^{d,t}), \end{aligned} \tag{3}$$

for the forward and backward directions, respectively.

3.3 Output Layers

NER. Standard CRFs (Ma and Hovy, 2016) are used as output layers for NER. Given $\mathbf{h} = [\overrightarrow{\mathbf{h}}_1 \oplus \overleftarrow{\mathbf{h}}_1, \ldots, \overrightarrow{\mathbf{h}}_n \oplus \overleftarrow{\mathbf{h}}_n]$, the output probability $p(\boldsymbol{y}|\boldsymbol{x})$ over label sequence $\boldsymbol{y} = l_1, l_2, \ldots, l_i$ produced on input sentence $\boldsymbol{x}$ is:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp\{\sum_i (\mathbf{w}_{\text{CRF}}^{l_i} \cdot \mathbf{h}_i + b_{\text{CRF}}^{(l_{i-1}, l_i)})\}}{\sum_{\boldsymbol{y}'} \exp\{\sum_i (\mathbf{w}_{\text{CRF}}^{l'_i} \cdot \mathbf{h}_i + b_{\text{CRF}}^{(l'_{i-1}, l'_i)})\}}, \tag{4}$$

where $\boldsymbol{y}'$ represents an arbitary labal sequence, and $\mathbf{w}_{\text{CRF}}^{l_i}$ is a model parameter specific to l_i, and $b_{\text{CRF}}^{(l_{i-1}, l_i)}$ is a bias specific to l_{i-1} and l_i.

Considering that the NER label sets across domains can be different, we use CRF(S) and CRF(T) to represent CRFs for the source and target domains in Figure 2, respectively. We use the first-order Viterbi algorithm to find the highest scored label sequence.

Language modeling. A forward LM (LM_f) uses the forward LSTM hidden state $\overrightarrow{\mathbf{h}} = [\overrightarrow{\mathbf{h}}_1, \ldots, \overrightarrow{\mathbf{h}}_n]$ to compute the probability of next word x_{i+1} given $x_{1:i}$, represented as $p^f(x_{i+1}|x_{1:i})$. A backward LM (LM_b) computes $p^b(x_{i-1}|x_{i:n})$ based on backward LSTM hidden state $\overleftarrow{\mathbf{h}} = [\overleftarrow{\mathbf{h}}_1, \ldots, \overleftarrow{\mathbf{h}}_n]$ in a similar manner.

Considering the computational efficiency, Negative Sampling Softmax (NSSoftmax) (Mikolov et al., 2013; Jean et al., 2014) is used to compute forward and backward probabilities, respectively, as follows:

$$\begin{aligned} p^f(x_{i+1}|x_{1:i}) &= \frac{1}{Z}\exp\{\mathbf{w}^\top_{\#x_{i+1}}\overrightarrow{\mathbf{h}}_i + b_{\#x_{i+1}}\} \\ p^b(x_{i-1}|x_{i:n}) &= \frac{1}{Z}\exp\{\mathbf{w}^\top_{\#x_{i-1}}\overleftarrow{\mathbf{h}}_i + b_{\#x_{i-1}}\}, \end{aligned} \tag{5}$$

where $\#x$ represents the vocabulary index of the target word x. $\mathbf{w}_{\#x}$ and $b_{\#x}$ are the target word vector and the target word bias, respectively. Z is the normalization item computed by

$$Z = \sum_{k\in\{\#x\cup\mathcal{N}_x\}} \exp\{\mathbf{w}_k^\top\overline{\mathbf{h}}_i + b_k\}, \tag{6}$$

where $\mathcal{N}_x$ represents the nagative sample set of the target word x. Each element in the set is a random number from 1 to the cross-domain vocabulary size. $\overline{\mathbf{h}}_i$ represents $\overrightarrow{\mathbf{h}}_i$ in LM_f and $\overleftarrow{\mathbf{h}}_i$ in LM_b, respectively.

3.4 Training Objectives

NER. Given a manually labeled dataset $\mathcal{D}_{ner} = \{(\boldsymbol{x}^n, \boldsymbol{y}^n)\}_{n=1}^N$, the sentence-level negative log-likelihood loss is used for training:

$$\mathcal{L}_{ner} = -\frac{1}{|\mathcal{D}_{ner}|}\sum_{n=1}^{N}\log(p(\boldsymbol{y}^n|\boldsymbol{x}^n)) \tag{7}$$

Language modeling. Given a raw data set $\mathcal{D}_{lm} = \{(\boldsymbol{x}^n)\}_{n=1}^N$, LM_f and LM_b are trained jointly using Negative Sampling Softmax. Negative samples are drawn based on word frequency distribution in $\mathcal{D}_{lm}$. The loss function is:

$$\begin{aligned} \mathcal{L}_{lm} = -\frac{1}{2\,|\mathcal{D}_{lm}|}\sum_{n=1}^{N}\sum_{t=1}^{T}\{\, &\log(p^f(\boldsymbol{x}^n_{t+1}|\boldsymbol{x}^n_{1:t})) \\ &+ \log(p^b(\boldsymbol{x}^n_{t-1}|\boldsymbol{x}^n_{t:T}))\,\} \end{aligned} \tag{8}$$

Joint training. To perform joint training for NER and language modeling on both the source and target domains, we minimize the overall loss:

$$\mathcal{L} = \sum_{d\in\{src,tgt\}} \lambda^d(\mathcal{L}^d_{ner} + \lambda^t\mathcal{L}^d_{lm}) + \frac{\lambda}{2}\|\Theta\|^2, \tag{9}$$

where λ^d is a domain weight and λ^t is a task weight. λ is the L_2 regularization parameters and Θ represents the parameters set.

Algorithm 1 Multi-task learning

Input: training data $\{\mathcal{S}_{ner}, \mathcal{T}^*_{ner}\}$ and $\{\mathcal{S}_{lm}, \mathcal{T}_{lm}\}$
Parameters:
- Parameters Generator: $\mathbf{W}$, $\{\mathbf{I}^D_d\}$, $\{\mathbf{I}^T_t\}$
- Output layers: θ_{crf_s},$\theta_{crf_t}{}^*$,θ_{nss}

Output: Target model

```
 1: while training steps not end do
 2:     split training data into minibatches:
        $B^{ner_s}$, $B^{ner_t*}$, $B^{lm_s}$, $B^{lm_t}$
 3:     # source-domain NER
 4:     $\theta^{src,ner}_{\text{LSTM}} \leftarrow f(\mathbf{W}, \mathbf{I}^D_{src}, \mathbf{I}^T_{ner})$
 5:     $\Delta\mathbf{W}, \Delta\mathbf{I}^D_{src}, \Delta\mathbf{I}^T_{ner}, \Delta\theta_{crf_s} \leftarrow train(B^{ner_s})$
 6:     # source-domain LM
 7:     $\theta^{src,lm}_{\text{LSTM}} \leftarrow f(\mathbf{W}, \mathbf{I}^D_{src}, \mathbf{I}^T_{lm})$
 8:     $\Delta\mathbf{W}, \Delta\mathbf{I}^D_{src}, \Delta\mathbf{I}^T_{lm}, \Delta\theta_{nss} \leftarrow train(B^{lm_s})$
 9:     if do supervised learning then
10:         # target-domain NER
11:         $\theta^{tgt,ner}_{\text{LSTM}} \leftarrow f(\mathbf{W}, \mathbf{I}^D_{tgt}, \mathbf{I}^T_{ner})$
12:         $\Delta\mathbf{W}, \Delta\mathbf{I}^D_{tgt}, \Delta\mathbf{I}^T_{ner}, \Delta\theta_{crf_t} \leftarrow train(B^{ner_t})$
13:     end if
14:     # target-domain LM
15:     $\theta^{tgt,lm}_{\text{LSTM}} \leftarrow f(\mathbf{W}, \mathbf{I}^D_{tgt}, \mathbf{I}^T_{lm})$
16:     $\Delta\mathbf{W}, \Delta\mathbf{I}^D_{tgt}, \Delta\mathbf{I}^T_{lm}, \Delta\theta_{nss} \leftarrow train(B^{lm_t})$
17:     Update $\mathbf{W}$, $\{\mathbf{I}^D\}$, $\{\mathbf{I}^T\}$, $\theta_{crf_s}$, $\theta_{crf_t}{}^*$, $\theta_{nss}$
18: end while
```

Note: * means none in unsupervised learning

3.5 Multi-Task Learning Algorithm

We propose a cross-task and cross-domain joint training method for multi-task learning. Algorithm 1 provides the training procedure. In each training step (line 1 to 18), minibatches of the 4 tasks in Figure 1 take turns to train (lines 4-5, 7-8, 11-12 and 15-16, respectively). Each task first generates the parameters $\theta^{d,t}_{\text{LSTM}}$ using $\mathbf{W}$ and their respective $\mathbf{I}^D_d$, $\mathbf{I}^T_t$, and then compute gradients for $f(\mathbf{W}, \mathbf{I}^D_d, \mathbf{I}^T_t)$ and domain-specific output layer (θ_{crf_s}, θ_{crf_t} or θ_{nss}). In the scenario of unsupervised learning, there is no training data of the target-domain NER, and lines 11-12 will not be executed. At the end of each training step, parameters of $f(\cdot,\cdot,\cdot)$ and private output layers are updated together in line 17.

4 Experiments

We conduct experiments on three cross-domain datasets, comparing our method with a range of transfer learning baselines under both the supervised domain adaptation and the unsupervised domain adaptation settings.

4.1 Experimental Settings

Data. We take the CoNLL-2003 English NER data (Sang and Meulder, 2003) as our source-domain data. In addition, 377,592 sentences from the Reuters are used for source-domain LM training in unsupervised domain adaptation. Three sets of target-domain data are used, including two publicly available biomedical NER datasets, BioNLP13PC (13PC) and BioNLP13CG (13CG) [1] and a science and technology dataset we collected and labeled. Statistics of the datasets are shown in Table 1.

CoNLL-2003 contains four types of entities, namely PER (person), LOC (location), ORG (organization) and MISC (miscellaneous). BioNLP13CG consists of five types, namely CHEM (Chemical), CC (cellular component), G/P (gene/protein), SPE (species) and CELL (cell), BioNLP13PC consists of three types of those entities: CHEM, CC and G/P. We use text of their training sets for language modeling training [2].

For the science and technology dataset, we collect 620 articles from CBS SciTech News[3], manually labeling them as a test set for unsupervised domain adaptation. It consists of four types of entities following the CoNLL-2003 standard. The numbers of each entity type are comparable to the CoNLL test set, as listed in Table 2. The main difference is that a great number of entities in the CBS News dataset are closely related to the domain of science and technology. In particular, for the MISC category, more technology terms such as Space X, bitcoin and IP are included, as compared with the CoNLL data set. Lack of such entities in the CoNLL training set and the difference of text genre cause the main difficulty in domain transfer. To address this difference, 398,990 unlabeled sentences from CBS SciTech News are used for LM training. We released this dataset as one contribution of this paper.

Hyperparameters. We choose NCRF++ (Yang and Zhang, 2018) for developing the models. Our hyperparameter settings largely follow (Yang et al., 2018), with the following exceptions: (1) The batch size is set to 30 instead of 10 for shorter training time in multi-task learning; (2) RMSprop with a learning rate of 0.001 is used for our Single Task Model (STM-TARGET) for the strongest baseline according to development experiments, while the multi-task models use SGD with a learning rate of 0.015 as (Yang et al., 2018). We use domain embeddings and task embeddings of size 8 to fit the model in one GPU of 8GB memory. The word embeddings for all models are initialized with GloVe 100-dimension vectors (Pennington et al., 2014) and fine-tuned during training. Character embeddings are randomly initialized.

Dataset	Type	Train	Dev	Test
CoNLL	Sentence Entity	15.0K 23.5K	3.5K 5.9K	3.7K 5.6K
BioNLP13PC	Sentence Entity	2.5K 7.9K	0.9K 2.7K	1.7K 5.3K
BioNLP13CG	Sentence Entity	3.0K 10.8K	1.0K 3.6K	1.9K 6.9K
CBS News	Sentence Entity	- -	- -	2.0K 4.1K

Table 1: Statistic of datasets.

Dataset		PER	LOC	ORG	MISC
CoNLL	Train	6,600	7,140	6,321	3,438
	Dev	1,842	1,837	1,341	922
	Test	1,617	1,668	1,661	702
CBS News	Test	1,660	629	1,352	497

Table 2: Entity numbers of the CoNLL dataset and the CBS SciTech News dataset.

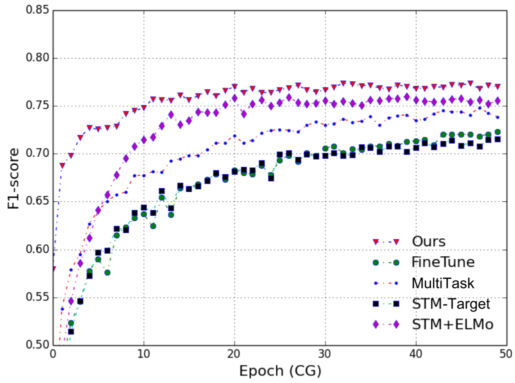

Figure 3: Development results on 13CG.

4.2 Development Experiments

We report a set of development experiments on the biomedical datasets 13PC and 13CG.

Learning curves. Figure 3 shows the F1-scores against the number of training iterations on the 13CG development set. STM-TARGET is our single task model trained on the target-domain training set $\mathcal{T}_{ner}$; FINETUNE is a model pre-trained

[1] https://github.com/cambridgeltl/MTL-Bioinformatics-2016

[2] We tried to use a larger number of raw data from the PubMed, but this did not improve the performances.

[3] https://www.cbsnews.com/

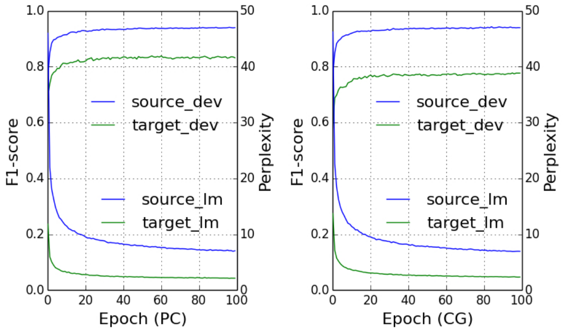

Figure 4: Joint training in multi-task learning.

using the source-domain training data $\mathcal{S}_{ner}$ and then fine-tuned using the target-domain data $\mathcal{T}_{ner}$; MULTITASK simultaneously trains source-domain NER and target-domain NER following Yang et al. (2017). For STM+ELMO, we mix the source- and target-domain raw data for training a contextualized ELMo representation (Peters et al., 2018), which is then used as inputs to an STM-TARGET model. This model shows a different way of transfer by using raw data, which is different from FINETUNE and MULTITASK. Note that due to differences in the label sets, FINETUNE and MULTITASK both share parameters between the two models except for the CRF layers.

As can be seen from Figure 3, the F1 of all models increase as the number of training iteration increases from 1 to 50, with only small fluctuations. All of the models converge to a plateau range when the iteration number increases to 100. All transfer learning methods outperform the STM-TARGET method, showing the usefulness of using source data to enhance target labeling. The strong performance of STM+ELMO over FINETUNE and MULTITASK shows the usefulness of raw text. By simultaneously using source-domain raw text and target-domain raw text, our model gives the best F1 over all iterations.

Effect of language model for transfer. Figure 4 shows the results of source language modeling, target language modeling, source NER and target NER for both development datasets when the number of training iterations increases. As can be seen, multi-task learning under our framework brings benefit to all tasks, without being negatively influenced by potential conflicts between tasks (Bingel and Søgaard, 2017; Mou et al., 2016).

Methods	Datasets	
	13PC	13CG
Crichton et al. (2017)	81.92	78.90
STM-TARGET	82.59	76.55
MULTITASK(NER+LM)	81.33	75.27
MULTITASK(NER)	83.09	77.73
FINETUNE	82.55	76.73
STM+ELMO	82.76	78.24
CO-LM	84.43	78.60
CO-NER	83.87	78.43
MIX-DATA	83.88	78.70
FINAL	**85.54**†	**79.86**†

Table 3: F1-scores on 13PC and 13CG. † indicates that the FINAL results are statistically significant compared to all transfer baselines and ablation baselines with $p < 0.01$ by t-test.

4.3 Final Results on Supervised Domain Adaptation

We investigate supervised transfer from CoNLL to 13PC and 13CG, comparing our model with a range of baseline transfer approaches. In particular, three sets of comparisons are made, including (1) a comparison between our method with other supervised domain adaptation methods, such as MULTITASK(NER) [4] and ELMo, (2) a comparison between the use of different subsets of data for transfer under our own framework and (3) a comparison with the current state-of-the-art in the literature for these datasets.

(1) Comparison with other supervised transfer methods. We compare our method with STM-TARGET, MULTITASK(NER), FINETUNE and STM+ELMO. The observations are similar to those on the development set. Note that FINETUNE does not always improve over STM-TARGET, which shows that the difference between the two datasets can hurt naive transfer learning, without considering domain descriptor vectors.

ELMo. The ELMo methods use raw text via language model pre-training, which has been shown to benefit many NLP tasks (Peters et al., 2018). In our cross-domain setting, STM+ELMO gives a significant improvement over STM-TARGET on the 13CG dataset, but only a small improvement on the 13PC dataset. The overall improvements are comparable to that of MULTITASK only using the raw data. We also tried to use the ELMo model (Original) released by Peters

[4] Here MULTITASK(NER) is the same model as MULTITASK in the development experiments.

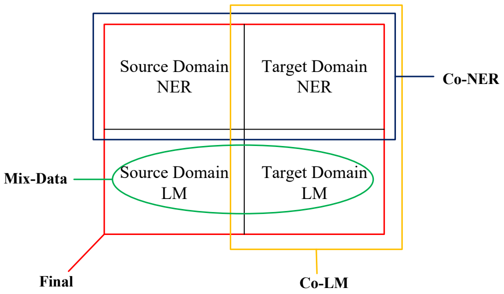

Figure 5: Ablations of the model.

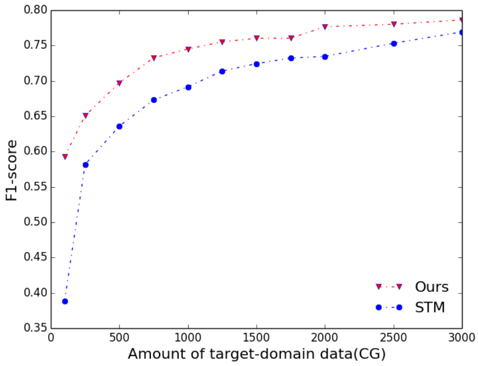

Figure 6: Influence of target-domain data.

et al. (2018) [5], which is trained over approximately 800M tokens. The results are 84.08% on 13PC and 79.57% on 13CG, respectively, which are lower compared to 85.54% and 79.86% by our method, respectively, despite the use of much larger external data. This shows the effectiveness of our model.

Multi-task of NER and LM. We additionally compare our method with the naive multi-task learning setting (Collobert and Weston, 2008), which uses shared parameters for the four tasks but use the exact same data conditions as the FINAL model. which is shown in the MULTI-TASK(NER+LM) method in Table 3. The method gives an 81.33% F1 on 13PC and 75.27% on 13CG, which is much lower compared with all baseline models. This demonstrates the challenge of the cross-domain and cross-task setting, which contains conflicting information from different text genres and task requirements.

(2) Ablation experiments. Now that we have compared our method with baselines utilizing similar data sources, we turn to investigate the influence of data sources on our own framework. As shown in Figure 5, we make novel use of 4 data sources for the combination of two tasks in two domains. If some sources are removed, our settings fall back to traditional transfer learning. For example, if the LM task is not considered, then the task setting is standard supervised domain adaptation.

The baselines include (1) CO-LM, which represents our model without source-domain tasks, joint training the target-domain NER and language modeling, transferring parameters as: $\theta^t_{\text{LSTM}} = \mathbf{W} \otimes \mathbf{I}^T_t, (t \in \{ner, lm\})$. (2) CO-NER, deleting tasks, jointly training source- and target-domain NER, transferring parameters as: $\theta^d_{\text{LSTM}} = \mathbf{W} \otimes \mathbf{I}^D_d, (d \in \{src, tgt\})$. (3) MIX-DATA, which uses the same NER data in source- and target-domain as FINAL, but also uses combined raw text to train source- and target-domain language models.

Our method outperforms all baselines significantly, which shows the importance of using rich data. A contrast between our method and MIX-DATA shows the effectiveness of using two different language models across domains. Even through MIX-DATA uses more data for training language models on both the source and target domains, it cannot learn a domain contrast since both sides use the same mixed data. In contrast, our model gives significantly better results by gleaning such contrast.

(3) Comparison with current state-of-the-art. Finally, Table 3 also shows a comparison with a state-of-the-art method on the 13PC and 13CG datasets (Crichton et al., 2017), which leverages POS tagging for multi-task learning by using co-training method. Our model outperforms their results, giving the best results in the literature.

Discussion. When the number of target-domain NER sentences is 0, the transfer learning setting is unsupervised domain adaptation. As the number of target domain NER sentences increases, they will intuitively play an increasingly important role for target NER. Figure 6 compares the F1-scores of the baseline STM-TARGET and our multi-task model with varying numbers of target-domain NER training data under 100 training epochs. In the nearly unsupervised setting, our method gives the largest improvement of 20.5% F1-scores. As the number of training data increases, the gap between the two methods becomes smaller. But our method still gives a 3.3% F1 score gain when the number of training sentences reach 3,000, show-

[5] https://allennlp.org/elmo

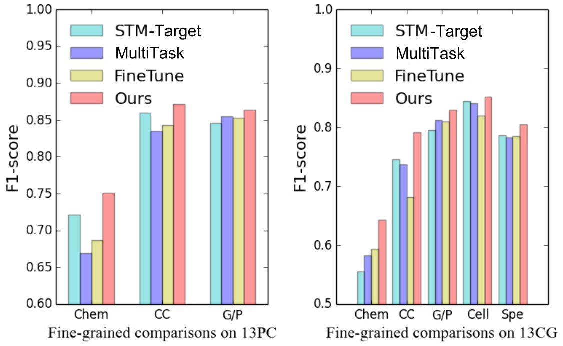

Figure 7: Fine-grained comparisons on 13PC and 13CG.

ing the effectiveness of LM in knowledge transfer.

Figure 7 shows fine-grained NER results of all available entity types. In comparison to Stm-Target, FineTune and MultiTask, our method outperforms all the baselines on each entity type, which is in accordance with the conclusion of development experiments.

4.4 Unsupervised Domain Adaptation

For unsupervised domain adaptation, many settings in Subsection 4.2 do not hold, including Stm-Target, FineTune, MultiTask, Co-Lm and Co-Ner. Instead, we add a naive baseline, Stm-Source, which directly applies a model trained on the source-domain CoNLL-2003 data to the target domain. In addition, we compare with models that make use of source NER, source LM and target LM data, including Self-Train, which improves a source NER model on target raw text (Daumé III, 2008). Stm-Elmo, which uses ELMo embeddings trained over combined source- and target-domain raw text for Stm-Source, Stm-Elmo(src), which uses only the source-domain raw data for training ELMo, Stm-Elmo(tgt), which uses only the target-domain raw text for training ELMo, and Dann (Ganin et al., 2016), which performs generative adversarial training over source- and target-domain raw data.

Final results. The final results are shown in Table 4. Self-Train gives better results compared with the Stm-Source baseline, which shows the effectiveness of target-domain raw data. Adversarial training brings significantly better improvements compared with naive self-training. Among ELMo methods, the model using both the source-domain raw data and target-domain raw data outperforms the model using only the source- or target-domain raw data. ELMo also outperforms Dann, which shows the strength of LM pre-training. Interestingly, ELMo with target-domain raw data gives similar accuracies to ELMo with mixed source- and target-domain data, which shows that target-domain LM is more useful for the pretraining method. It also indicates that our method makes better use of LMs over two different domains. Compared with all baseline models, our model gives a final F1 of 73.59, significantly better than the best result of 70.85 obtained by Stm+Elmo, demonstrating the effectiveness of parameter generation network for cross-task, cross-domain knowledge transfer.

Methods	**P**	**R**	**F1**
Stm-Source	63.87	71.28	67.37
Self-Train	62.56	75.04	68.24
Dann(Ganin et al., 2016)	65.14	73.84	69.22
Stm+Elmo(src)	65.43	70.14	67.70
Stm+Elmo(tgt)	67.78	72.73	70.17
Stm+Elmo	67.19	74.93	70.85
Ours	**68.48**	**79.52**	**73.59**†

Table 4: Three metrics on CBS SciTech News. We use the CoNLL dev set to select the hyperparameters of our models. ELMo and Ours are given the same overall raw data, Self-Train and Dann use the selected raw data from overall raw data for better performances. † indicates that our results are statistically significant compared to all baselines with $p < 0.01$ by t-test.

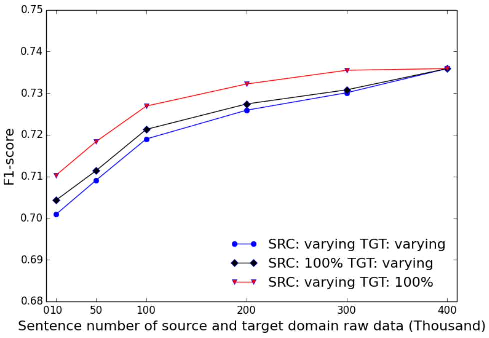

Figure 8: Amount of raw data.

Influence of raw text. For zero-shot learning, domain adaptation is achieved solely through LM channels. We thus compare the effectiveness of raw text from both the source domain and the target domain. Figure 8 shows the results. The line "SRC: varying; TGT: varying" shows the F1-scores against varying numbers of raw sentences in both source and target domains. Each number in the x-coordinate indicates an equal amount of source- and target-domain text. As can be seen, increasing raw text gives increased F1 for

Entity Type	Correct Num		Δ
	STM	Ours	
PER	1,501	1,569	+4.10%
LOC	469	512	+6.84%
ORG	941	1,050	+8.06%
MISC	134	193	+11.87%
Total	3,045	3,324	+6.74%

Table 5: Growth rate of correctly recognized enetity number in comparison with the STM-SOURCE. Δ represents the growth with respect to the total number of entities in the CBS SciTech News test set.

Sentence	Brittany Kaiser spoke to "CBS This Morning" co-host John Dicherson for her first U.S. broadcast network interview.
STM-SRC	Brittany Kaiser ORG spoke to "CBS ORG This Morning" ...
DANN	Brittany Kaiser PER spoke to "CBS This Morning ORG" ...
Ours	Brittany Kaiser PER spoke to "CBS This Morning MISC" ...

Table 6: Example. Red and green represent incorrect and correct entities, respectively.

NER, which demonstrates effective use of raw data by our method. The lines "SRC: 100%; TGT: varying" and "SRC: varying; TGT: 100%" show to alternative measures by fixing the source- and target-domain raw text to 100% of our data, and then varying only the other domain text. A comparison between the two lines shows that the target-domain raw data gives more influence to the domain adaptation power, which conforms to intuition.

Discussion. Table 5 shows a breakdown for the improvement of our model over STM-SOURCE by different entity types. Compared with PER, LOC and ORG names, our method brings the most improvements over MISC entities, which are mostly types that are specific to the technology domain (see Subsection 4.1). Intuitively, the amount of overlap is the weakest for this type of entities between raw text from source and target domains. Therefore, the results show the effectiveness of our method in deriving domain contrast with respect to NER from cross-domain language modeling.

Table 6 shows a case study, where "Brittany Kaiser" is a personal name and "CBS This Morning" is a programme. Without using raw text, STM-SOURCE misclassifies "Brittany Kaiser" as ORG. Both DANN and our method give the correct results because the name is mentioned in raw text, from which connections between the pattern "PER spoke" can be drawn. With the help of raw text, DANN and our method can also recognize "CBS This Morning" as a entity, which has a common pattern of consecutive capital letters in both source and target domains.

DANN misclassifies "CBS This Morning" as ORG. In contrast, our model can classify it correctly as the category of MISC, in which most entities are specific to the target domain (see Subsection 4.1). This is likely because adversarial training in DANN aims to match feature distributions between source and target domains by mimicing the domain discriminator, which can lead to concentration on domain common features but confusion about such domain-specific features. This demonstrates the advantage of our method in deriving both domain common and domain-specific features.

5 Conclusion

We considered NER domain adaptation by extracting knowledge of domain differences from raw text. For this goal, cross-domain language modeling is conducted through a novel parameter generation network, which decomposes domain and task knowledge into two sets of embedding vectors. Experiments on three datasets show that our method is highly effective among supervised domain adaptation methods, while allowing zero-shot learning in unsupervised domain adaptation.

Acknowledgments

The three authors contributed equally to this work. Yue Zhang is the corresponding author. We gratefully acknowledge funding from NSFC (grant #61572245). We also thank the anonymous reviewers for their helpful comments and suggestions.

References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Short Papers)*, volume 2, pages 164–169. Association for Computational Linguistics.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(1):2493–2537.

Gamal Crichton, Sampo Pyysalo, Billy Chiu, and Anna Korhonen. 2017. A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics*, 18(1):368.

Hal Daumé III. 2009. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263. Association for Computational Linguistics.

Hal Daumé III. 2008. Cross-task knowledge-constrained self training. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 1, pages 680–688. Association for Computational Linguistics.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030.

James Hammerton. 2003. Named entity recognition with long short-term memory. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL*, volume 4, pages 172–175. Association for Computational Linguistics.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1–10. Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong. 2015. New transfer learning techniques for disparate label sets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Long Papers)*, volume 1, pages 473–482. Association for Computational Linguistics.

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

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2017. Transfer learning for named-entity recognition with neural networks. *Computing Research Repository*, arXiv:1705.06273. Version 1.

Bill Yuchen Lin and Wei Lu. 2018. Neural adaptation layers for cross-domain named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2012–2022. Association for Computational Linguistics.

Qi Liu, Yue Zhang, and Jiangming Liu. 2018. Learning domain representation for multi-domain sentiment classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long Papers)*, volume 1, pages 541–550. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, volume 1, pages 1064–1074. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489. Association for Computational Linguistics.

Rasha Obeidat, Xiaoli Fern, and Prasad Tadepalli. 2016. Label embedding approach for transfer learning. In *International Conference on Biomedical Ontology and BioCreative*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, volume 4, pages 1532–1543. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long Papers)*, volume 1, pages 2227–2237. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom M. Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 425–435. Association for Computational Linguistics.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, volume 1, pages 2121–2130. Association for Computational Linguistics.

Devendra Singh Sachan, Pengtao Xie, and Eric P. Xing. 2018. Effective use of bidirectional language modeling for medical named entity recognition. *Proceedings of Machine Learning Research*, 85:1–19.

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 Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147. Association for Computational Linguistics.

Cicero Nogueira dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entity Workshop, joint with 53rd ACL and the 7th IJCNLP*, pages 25–33. Association for Computational Linguistics.

Sara Stymne, Miryam de Lhoneux, Aaron Smith, and Joakim Nivre. 2018. Parser training with heterogeneous treebanks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 619–625. Association for Computational Linguistics.

Zhenghui Wang, Yanru Qu, Liheng Chen, Shen Jian, Weinan Zhang, Shaodian Zhang, Yimei Gao, Gen Gu, Ken Chen, and Yu Yong. 2018. Label-aware double transfer learning for cross-specialty medical named entity recognition. In *Proceedings of NAACL-HLT 2018*, pages 1–15. Association for Computational Linguistics.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3879–3889.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics-System Demonstrations*, pages 74–79. Association for Computational Linguistics.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In *International Conference on Learning Representations*.