Multi-Cell Compositional LSTM for NER Domain Adaptation

Chen Jia[†‡] and **Yue Zhang**[‡§]
[†]Fudan University, China
[‡]School of Engineering, Westlake University, China
[§]Institute of Advanced Technology, Westlake Institute for Advanced Study, China
{jiachen,zhangyue}@westlake.edu.cn

Abstract

Cross-domain NER is a challenging yet practical problem. Entity mentions can be highly different across domains. However, the correlations between entity types can be relatively more stable across domains. We investigate a multi-cell compositional LSTM structure for multi-task learning, modeling each entity type using a separate cell state. With the help of entity typed units, cross-domain knowledge transfer can be made in an entity type level. Theoretically, the resulting distinct feature distributions for each entity type make it more powerful for cross-domain transfer. Empirically, experiments on four few-shot and zero-shot datasets show our method significantly outperforms a series of multi-task learning methods and achieves the best results.

1 Introduction

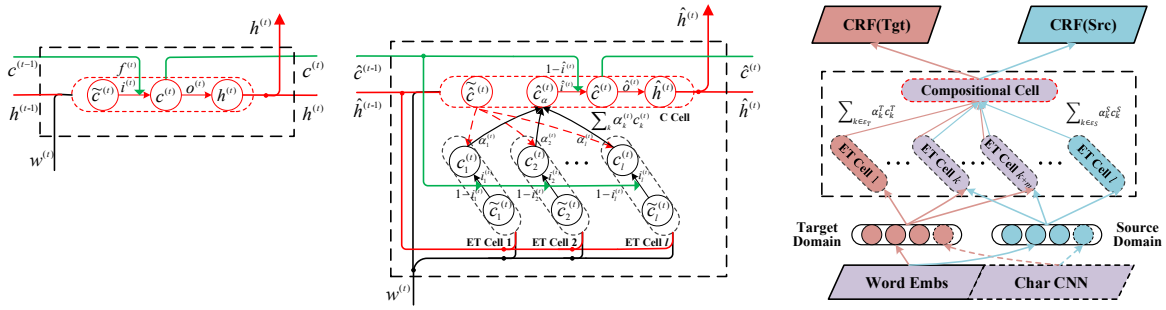
Named entity recognition (NER) is a fundamental task in information extraction, providing necessary information for relation classification (Mooney and Bunescu, 2006), event detection (Popescu et al., 2011), sentiment classification (Mitchell et al., 2013), etc. NER is challenging because entity mentions are an open set and can be ambiguous in the context of a sentence. Due to relatively high cost in manual labeling, cross-domain NER has received increasing research attention. Recently, multi-task learning methods (Yang et al., 2017; Wang et al., 2018, 2019; Zhou et al., 2019; Jia et al., 2019) have achieved great success for cross-domain NER. Other methods such as fine-tuning (Rodriguez et al., 2018), share-private (Cao et al., 2018; Lin and Lu, 2018) and knowledge distill (Yang et al., 2019) also show effectivenesses for cross-domain NER.

There are three main source of challenges in cross-domain NER. First, instances of the same type entities can be different across domains. For example, typical person names can include "Trump" and "Clinton" in the political news domain, but "James" and "Trout" in the sports domain. Second, different types of entities can exhibit different degrees of dissimilarities across domains. For example, a large number of location names are shared in the political news domain and the sports domain, such as "Barcelona" and "Los Angeles", but the case is very different for organization names across these domains. Third, even types of entities can be different across domains. For example, while disease names are a type of entities in the medical domain, it is not so in the biochemistry domain.

We investigate a multi-cell compositional LSTM structure to deal with the above challenges by separately and simultaneously considering the possibilities of all entity types for each word when processing a sentence. As shown in Figure 1, the main idea is to extend a standard LSTM structure by using a separate LSTM cell to model the state for each entity type in a recurrent step. Intuitively, the model differs from the baseline LSTM by *simultaneously* considering all possible entity types. A compositional cell (C cell) combines the entity typed cells (ET cells) for the next recurrent state transition by calculating a weighted sum of each ET cell, where the weight of each ET cell corresponds to the probability of its corresponding entity type. Different from naive parameter sharing on LSTM (Yang et al., 2017), source domain and target domain in our multi-task learning framework share only the ET cells corresponding to the same entity types and the same C cell, but not for the domain-specific ET cells. In this way, our model learns domain-invariant in the entity level.

Intuitively, our model addresses the above challenges by modeling entity type sequences more explicity, which are relatively more robust across domains compared with entity instances. For example, the pattern "PER O PER O LOC" can exist in both the political and sports domains, despite

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5906–5917
July 5 - 10, 2020.



(a) Baseline LSTM unit. (b) Multi-cell compositional LSTM unit. (c) Multi-task learning framework.

Figure 1: Overall structures. The red, blue and purple in (c) represent target, source and shared parts, respectively.

that the specific PER instances can be different. In addition, thanks to the merging operation at each step, our method effectively encodes multiple entity type sequences in linear time by having a sausage shaped multi-cell LSTM. Thus it allows us to learn distributional differences between entity type chains across domains. This effectively reduces the confusions of different entities when source domain and target domain have different entity types in few-shot transfer, where the target domain has a few training data. In zero-shot transfer where the target domain has no training data, a target-domain LM transfers source-domain knowledge. This knowledge transfer is also in the entity level thanks to the compositional weights which are supervised by gold-standard entity type knowledge in source-domain training.

Theoretically, our method creates distinct feature distributions for each entity type across domains, which can give better transfer learning power compared to representation networks that do not explicitly differentiate entity types (§3.4). Empirically, experiments on four few-shot and zero-shot datasets show that our method gives significantly better results compared to standard BiLSTM baselines with the same numbers of parameters. In addition, we obtain the best resutls on four cross-domain NER datasets. The code is released at https://github.com/jiachenwestlake/Multi-Cell_LSTM.

2 Method

Given a sentence $\boldsymbol{x} = [x_1, \ldots, x_m]$, the vector representation $\mathbf{w}_t$ for each word x_t is the concatenation of its word embedding and the output of a character level CNN, following Yang et al. (2018). A bi-directional LSTM encoder is used to obtain sequence level features $\mathbf{h} = [\mathbf{h}_1, \ldots, \mathbf{h}_m]$. We use the forward LSTM component to explain the details in the following subsections. Finally, a CRF layer outputs the label sequence $\boldsymbol{y} = l_1, \ldots, l_m$.

2.1 Baseline LSTM

We adopt the standard LSTM (Graves and Schmidhuber, 2005) for the baseline. At each time step t ($t \in [1, ..., m]$), the baseline calculates a current hidden vector $\mathbf{h}^{(t)}$ based on a memory cell $\mathbf{c}^{(t)}$. In particular, a set of input gate $\mathbf{i}^{(t)}$, output gate $\mathbf{o}^{(t)}$ and forget gate $\mathbf{f}^{(t)}$ are calculated as follows:

$$\begin{bmatrix} \mathbf{i}^{(t)} \\ \mathbf{o}^{(t)} \\ \mathbf{f}^{(t)} \\ \widetilde{\mathbf{c}}^{(t)} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(\mathbf{W}[\mathbf{h}^{(t-1)}; \mathbf{w}^{(t)}] + \mathbf{b} \right) \quad (1)$$
$$\mathbf{c}^{(t)} = \mathbf{i}^{(t)} \odot \widetilde{\mathbf{c}}^{(t)} + \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)}$$
$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot \tanh(\mathbf{c}^{(t)}),$$

where $[\mathbf{W}; \mathbf{b}]$ are trainable parameters. σ represents the sigmoid activation function.

2.2 Multi-Cell Compositional LSTM

As shown in Figure 1 (b), we split cell computation in the baseline LSTM unit into l copies, each corresponding to one entity type. These cells are shown in black. A compositional cell (shown in red) is used to merge the entity typed LSTM cells into one cell state for calculating the final hidden vector. In this process, a weight is assigned to each entity type according to the local context.

Entity typed LSTM cells (ET cells). Given $\mathbf{w}^{(t)}$ and $\hat{\mathbf{h}}^{(t-1)}$, the input gate $\mathbf{i}_k^{(t)}$ and the temporary memory cell state $\widetilde{\mathbf{c}}_k^{(t)}$ of the k-th ($k \in [1, \ldots, l]$) entity typed cells (ET cells) are computed as:

$$\begin{bmatrix} \mathbf{i}_k^{(t)} \\ \widetilde{\mathbf{c}}_k^{(t)} \end{bmatrix} = \begin{bmatrix} \sigma \\ \tanh \end{bmatrix} \left(\mathbf{W}_k[\hat{\mathbf{h}}^{(t-1)}; \mathbf{w}^{(t)}] + \mathbf{b}_k \right), \quad (2)$$

where the $[\mathbf{W}_k; \mathbf{b}_k]$ represent the trainable parameters specific to the k-th ET cell.

Then a copy of the compositional memory cell state $\hat{\mathbf{c}}^{(t-1)}$ of the previous time step $(t-1)$ is used to update the temporary memory cell state.

$$\mathbf{c}_k^{(t)} = \mathbf{i}_k^{(t)} \odot \widetilde{\mathbf{c}}_k^{(t)} + (\mathbf{1} - \mathbf{i}_k^{(t)}) \odot \hat{\mathbf{c}}^{(t-1)} \quad (3)$$

The above operations are repeated for l ET cells with the same $\hat{\mathbf{c}}^{(t-1)}$. We finally acquire a list of ET cell states $[\mathbf{c}_1^{(t)}, \ldots, \mathbf{c}_l^{(t)}]$.

Compositional LSTM cell (C cell). For facilitating integration of ET cells, a input gate $\hat{\mathbf{i}}^{(t)}$ and a temporary cell state $\widetilde{\hat{\mathbf{c}}}^{(t)}$ of the compositional cell (C cell) are computed similarly to those of the ET cells, but another output gate $\hat{\mathbf{o}}^{(t)}$ is added, which are computed as follows:

$$\begin{bmatrix} \hat{\mathbf{i}}^{(t)} \\ \hat{\mathbf{o}}^{(t)} \\ \widetilde{\hat{\mathbf{c}}}^{(t)} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(\hat{\mathbf{W}} [\hat{\mathbf{h}}^{(t-1)}; \mathbf{w}^{(t)}] + \hat{\mathbf{b}} \right), \quad (4)$$

where $[\hat{\mathbf{W}}; \hat{\mathbf{b}}]$ are trainable parameters of the C cell.

Merging. We use the temporary cell state of the C cell $\widetilde{\hat{\mathbf{c}}}^{(t)}$ to weigh the internal representations of ET cells $[\mathbf{c}_1^{(t)}, \ldots, \mathbf{c}_l^{(t)}]$ for obtaining a compositional representation. To this end, additive attention (Dzmitry et al., 2015) is used, which achieves better results in our development compared with other attention mechanism (Vaswani et al., 2017). The temporary memory cell state of the C cell $\hat{\mathbf{c}}_\alpha^{(t)}$ is a weighted sum of $[\mathbf{c}_1^{(t)}, \ldots, \mathbf{c}_l^{(t)}]$:

$$\hat{\mathbf{c}}_\alpha^{(t)} = \sum_{k=1}^{l} \alpha_k^{(t)} \mathbf{c}_k^{(t)} \quad s.t. \sum_{k=1}^{l} \alpha_k^{(t)} = 1 \quad (5)$$

The weight $\alpha_k^{(t)}$ reflects the similarity between $\widetilde{\hat{\mathbf{c}}}^{(t)}$ and the k-th ET cell state $\mathbf{c}_k^{(t)}$. $\alpha_k^{(t)}$ is computed as:

$$\begin{aligned} I_k^{(t)} &= \mathbf{v}^\top \tanh(\mathbf{P} \widetilde{\hat{\mathbf{c}}}^{(t)} + \mathbf{Q} \mathbf{c}_k^{(t)}) \\ \alpha_k^{(t)} &= \frac{\exp(I_k^{(t)})}{\sum_{j=1}^{l} \exp(I_j^{(t)})}, \end{aligned} \quad (6)$$

where $[\mathbf{P}; \mathbf{Q}; \mathbf{v}]$ are trainable parameters. The memory cell state of the C cell is updated as:

$$\hat{\mathbf{c}}^{(t)} = \hat{\mathbf{i}}^{(t)} \odot \hat{\mathbf{c}}_\alpha^{(t)} + (\mathbf{1} - \hat{\mathbf{i}}^{(t)}) \odot \hat{\mathbf{c}}^{(t-1)} \quad (7)$$

Finally, we obtain the hidden state $\hat{\mathbf{h}}^{(t)}$:

$$\hat{\mathbf{h}}^{(t)} = \hat{\mathbf{o}}^{(t)} \odot \tanh(\hat{\mathbf{c}}^{(t)}) \quad (8)$$

2.3 Training Tasks

Below we discuss the two auxiliary tasks before introducing the main NER task. The auxiliary tasks are designed in addition to the main NER task in order to better extract entity type knowledge from a set of labeled training data for training ET cells and C cell. Formally, denote a training set as $\mathcal{D}_{ent} = \{(\boldsymbol{x}^n, \boldsymbol{e}^n)\}_{n=1}^N$, where each training instance consists of word sequence $\boldsymbol{x} = [x_1, \ldots, x_m]$ and its corresponding entity types $\boldsymbol{e} = [e_1, \ldots, e_m]$. Here each entity type e_t is a label such as [PER, O, LOC,...] without segmentation tags (e.g., B/I/E).

Entity type prediction. Given the ET cell states of x_t: $\mathbf{c}^{(t)} = [\overrightarrow{\mathbf{c}}_1^{(t)} \oplus \overleftarrow{\mathbf{c}}_1^{(t)}, \ldots, \overrightarrow{\mathbf{c}}_l^{(t)} \oplus \overleftarrow{\mathbf{c}}_l^{(t)}]$, we define the aligned entity distribution for x_t:

$$p(e_k|x_t) = \frac{\exp\{\mathbf{w}_k^\top \mathbf{c}_k^{(t)} + b_k\}}{\sum_{j=1}^{l} \exp\{\mathbf{w}_j^\top \mathbf{c}_j^{(t)} + b_j\}}, \quad (9)$$

Where $[\mathbf{w}_k; b_k]$ are parameters specific to the k-th entity type e_k. The negative log-likehood loss is used for training on $\mathcal{D}_{ent}$:

$$\mathcal{L}_{ent} = -\frac{1}{|\mathcal{D}_{ent}|} \sum_{n=1}^{N} \sum_{t=1}^{m} \log(p(\boldsymbol{e}_t^n|\boldsymbol{x}_t^n)) \quad (10)$$

Attention scoring. Similar to the entity type prediction task, given the attention scores between the temporary C cell and ET cells in Equation 6: $\mathbf{I}^{(t)} = [(\overrightarrow{I}_1^{(t)} + \overleftarrow{I}_1^{(t)})/2, \ldots, (\overrightarrow{I}_l^{(t)} + \overleftarrow{I}_l^{(t)})/2]$, we convert the attention scores to entity aligned distributions for x_t using $\mathrm{softmax}$:

$$p(e_k|x_t) = \frac{\exp(\mathbf{I}_k^{(t)})}{\sum_{j=1}^{l} \exp(\mathbf{I}_j^{(t)})} \quad (11)$$

Similar to the loss of entity type prediction:

$$\mathcal{L}_{atten} = -\frac{1}{|\mathcal{D}_{ent}|} \sum_{n=1}^{N} \sum_{t=1}^{m} \log(p(\boldsymbol{e}_t^n|\boldsymbol{x}_t^n)) \quad (12)$$

While entity type prediction brings supervised information to guide the ET cells, attention scoring introduces supervision to guide the C cell.

NER. This is the main task across domains. Standard CRFs (Ma and Hovy, 2016) are used. Given $\mathbf{h} = [\overrightarrow{\mathbf{h}}_1 \oplus \overleftarrow{\mathbf{h}}_1, \ldots, \overrightarrow{\mathbf{h}}_m \oplus \overleftarrow{\mathbf{h}}_m]$, the output probability $p(\boldsymbol{y}|\boldsymbol{x})$ over labels $\boldsymbol{y}$=$l_1, \ldots, l_m$ is:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp\{\sum_t (\mathbf{w}_{\mathrm{CRF}}^{l_t} \cdot \mathbf{h}_t + b_{\mathrm{CRF}}^{(l_{t-1}, l_t)})\}}{\sum_{\boldsymbol{y}'} \exp\{\sum_t (\mathbf{w}_{\mathrm{CRF}}^{l'_t} \cdot \mathbf{h}_t + b_{\mathrm{CRF}}^{(l'_{t-1}, l'_t)})\}}, \quad (13)$$

where $\boldsymbol{y}'$ represents an arbitary labal sequence, and $\mathbf{w}_{\mathrm{CRF}}^{l_t}$ is a model parameter specific to l_t, and $b_{\mathrm{CRF}}^{(l_{t-1}, l_t)}$ is a bias specific to l_{t-1} and l_t.

Algorithm 1 Transfer learning

Input: Source-domain NER dataset $\mathcal{S}_{ner}$, target-domain NER dataset $\mathcal{T}_{ner}$ or raw data $\mathcal{T}_{lm}$ and entity dictionary D_e
Output: Target-domain model

1: **while** training steps not end **do**
2: **for** d in { Source, Target } **do**
3: **for** $\mathbf{w}^{(t)}$ in $[\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(m)}]$ **do**
4: $\{\mathbf{c}_k^{(t)}\}_{k\in\mathcal{E}_d} \leftarrow \{C_k(\hat{\mathbf{h}}^{(t-1)}, \mathbf{w}^{(t)}, \hat{\mathbf{c}}^{(t-1)})\}_{k\in\mathcal{E}_d}$
$\widetilde{\mathbf{c}}^{(t)} \leftarrow \hat{C}(\hat{\mathbf{h}}^{(t-1)}, \mathbf{w}^{(t)})$
$\{\hat{\mathbf{h}}^{(t)}, \hat{\mathbf{c}}^{(t)}\} \leftarrow \mathrm{Atten}\Big(\widetilde{\mathbf{c}}^{(t)}, \{\mathbf{c}_k^{(t)}\}_{k\in\mathcal{E}_d}\Big)$ (eq.2-8)
5: **end for**
6: Compute $\mathcal{L}_a^d \leftarrow \lambda_{ent}\mathcal{L}_{ent} + \lambda_{atten}\mathcal{L}_{atten}$
7: **if** d = Source **then**
8: Compute $\mathcal{L}_m^S \leftarrow \mathcal{L}_{ner}^S$
9: **else if** d = Target **then**
10: **if** do SDA **then**
11: Compute $\mathcal{L}_m^T \leftarrow \mathcal{L}_{ner}^T$
12: **else if** do UDA **then**
13: Compute $\mathcal{L}_m^T \leftarrow \mathcal{L}_{lm}^T$
14: **end if**
15: **end if**
16: $\mathcal{L} \leftarrow \mathcal{L} + \lambda^d\mathcal{L}_m^d + \mathcal{L}_a^d$
17: **end for**
18: Update paremeters of networks based on $\mathcal{L}$.
19: **end while**

A sentence-level negative log-likehood loss is used for training on $\mathcal{D}_{ner}$=$\{(\boldsymbol{x}^n, \boldsymbol{y}^n)\}_{n=1}^N$:

$$\mathcal{L}_{ner} = -\frac{1}{|\mathcal{D}_{ner}|}\sum_{n=1}^{N}\log(p(\boldsymbol{y}^n|\boldsymbol{x}^n)) \quad (14)$$

3 Transfer Learning

The multi-cell LSTM structure above is domain agnostic, and can therefore be used for in-domain NER too. However, the main goal of the model is to transfer entity sequence knowledge across domains, and therefore the ET cells and C cell play more significant roles in the transfer learning setting. Below we introduce the specific roles each cell is assigned in cross-domain settings.

3.1 Multi-Task Structure

Following the common cross-domain setting, we use source-domain NER dataset $\mathcal{S}_{ner}$ and the target-domain NER dataset $\mathcal{T}_{ner}$ or raw data $\mathcal{T}_{lm}$. The entity type sets of source and target domains are represented as $\mathcal{E}_d$, where $d \in \{S, T\}$, respectively.

As shown in Figure 1 (c), our multi-task learning structure follows Yang et al. (2017), which consists of shared embedding layer and shared BiLSTM layer, as well as domain-specific CRF layers. Our method replaces LSTM with multi-cell LSTM, following we introduce the multi-task parameter sharing mechanism in multi-cell LSTM.

ET cells. All ET cells $\{C_k\}_{k\in\mathcal{E}_S\cup\mathcal{E}_T}$ in multi-cell LSTM are a composion of entity-specific cells from both source and target domains. For each domain $d \in \{S, T\}$, the actually used ET cells are the domain-specific subset $\{C_k\}_{k\in\mathcal{E}_d}$, aiming to conserve domain-specific features.

C cell. In order to make the source and target domains share the same feature space in a word level, we use a shared C cell $\hat{C}$ across domains.

3.2 Unsupervised Domain Adaptation

To better leverage target-domain knowledge without target-domain NER labeled data, we conduct the auxiliary dictionary matching and language modeling tasks on target-domain raw data $\mathcal{T}_{lm} = \{(\boldsymbol{x}^n)\}_{n=1}^N$.

Auxiliary tasks. To better extract entity knowledge from raw data, we use a pre-collected named entity dictionary D_e by Peng et al. (2019) to label $\mathcal{T}_{lm}$ and obtain a set of entity words $\mathcal{D}_{ent}^+$, which are used to train entity prediction task and attention scoring task jointly.

Language modeling. Follwing Jia et al. (2019), we use sampling softmax to compute forward LM probability $p^f(\boldsymbol{x}_t|\boldsymbol{x}_{<t})$ and backward LM probability $p^b(\boldsymbol{x}_t|\boldsymbol{x}_{>t})$, respectively:

$$\begin{aligned} p^f(\boldsymbol{x}_t|\boldsymbol{x}_{<t}) &= \frac{1}{Z}\exp\left(\mathbf{w}_{x_t}^\top \overrightarrow{\mathbf{h}}_{t-1} + b_{x_t}\right) \\ p^b(\boldsymbol{x}_t|\boldsymbol{x}_{>t}) &= \frac{1}{Z}\exp\left(\mathbf{w}_{x_t}^\top \overleftarrow{\mathbf{h}}_{t+1} + b_{x_t}\right), \end{aligned} \quad (15)$$

where $\mathbf{w}_x$ and b_x are the target word vector and bias, respectively. Z is the normalization item computed by the target word and negative samples.

The LM loss function on $\mathcal{T}_{lm}$ is:

$$\begin{aligned} \mathcal{L}_{lm}^T = -\frac{1}{2\,|\mathcal{T}_{lm}|}\sum_{n,t=1}^{N,m}\Big\{ &\log(p^f(\boldsymbol{x}_t^n|\boldsymbol{x}_{<t}^n)) \\ &+ \log(p^b(\boldsymbol{x}_t^n|\boldsymbol{x}_{>t}^n))\Big\} \end{aligned} \quad (16)$$

3.3 Training Objective

Algorithm 1 is the transfer learning algorithm under both supervised and unsupervised domain adaptation settings. Both source- and target-domain training instances undertake auxiliary tasks and obtain the loss $\mathcal{L}_a$, which is a combination of $\mathcal{L}_{ent}$ and $\mathcal{L}_{atten}$ weighted by λ_{ent} and λ_{atten}, respectively (line 6).

Supervised domain adaptation. The auxiliary tasks as well as source- and target-domain NER tasks (line 8, 11) form the final training objective:

$$\mathcal{L}_{\mathrm{SDA}} = \sum_{d\in\{S,T\}}\Big\{\lambda_d\mathcal{L}_{ner}^d + \mathcal{L}_a^d\Big\} + \frac{\lambda}{2}\|\Theta\|^2, \quad (17)$$

where λ_d ($d \in \{S, T\}$) are the domain weights for NER tasks. λ is the L_2 regularization parameters and Θ represents the parameters set.

Unsupervised domain adaptation. The training objective for UDA is similar to that of SDA, except for using target-domain LM task (line 13) instead of target-domain NER task:

$$\mathcal{L}_{\text{UDA}} = \mathcal{L}_{ner}^{S} + \mathcal{L}_{lm}^{T} + \mathcal{L}_{a}^{S} + \mathcal{L}_{a}^{T} + \frac{\lambda}{2}\|\Theta\|^2 \quad (18)$$

3.4 Theoretical Discussion

Below we show theoretically that our method in §2.2 is stronger than the baseline method in §2.1 for domain adaptation. Following Ben-David et al. (2010), a domain is defined as a pair of input distribution $\mathcal{D}$ on $\mathcal{X}$ and a labeling function y: $\mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y}$ is a $(l-1)$-simplex[1]. According to this definition, $<\mathcal{D}_S, y_S>$ and $<\mathcal{D}_T, y_T>$ represent source and target domains, respectively. A hypothesis is a function h: $\mathcal{X} \to \{1, ..., l\}$, which can be a classification model.

Target-domain error is defined as the probability h_T disagrees with y_T, $\epsilon(h_T) = \epsilon(h_T, y_T) = \mathrm{E}_{x\sim\mathcal{D}_T}[|y_T - h_T(x)|]$. The training target for h is to minimize a convex weighted combination of source and target errors, $\epsilon_\alpha(h) = \alpha\epsilon_T(h) + (1-\alpha)\epsilon_S(h)$, where $\alpha \in [0, 1)$ is the domain weight, when $\alpha = 0$, it is the setting of UDA.

Theorem 1 Let h be a hypothesis in class $\mathcal{H}$, then:

$$\epsilon_T(h) \le \epsilon_\alpha(h) + (1-\alpha)\left(\tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_S, \mathcal{D}_T\right) + \lambda\right),$$

where

$$d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_S, \mathcal{D}_T\right) = 2 \sup_{h', h'' \in \mathcal{H}} \Big| \Pr_{x\sim\mathcal{D}_S}\left[h'(x) \neq h''(x)\right] - \Pr_{x\sim\mathcal{D}_T}\left[h'(x) \neq h''(x)\right] \Big|$$

Here λ is a constant that values the shared error of the ideal joint hypothesis. In $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T)$, sup denotes the supremum of the right term for $\forall h', h'' \in \mathcal{H}$. $\Pr_{x\sim\mathcal{D}_S}\left[h'(x) \neq h''(x)\right]$ denotes the probability according to the distribution $\mathcal{D}_S$ that h' disagrees with h'' and $\Pr_{x\sim\mathcal{D}_T}\left[h'(x) \neq h''(x)\right]$ is similar. Intuitively, the theorem states the upper bound of $\epsilon_T(h)$ based on $\epsilon_\alpha(h)$ and the distance between $\mathcal{D}_S$ and $\mathcal{D}_T$ in the $\mathcal{H}\Delta\mathcal{H}$ space, which is measured as the discrepancy between the two classifiers h' and h''.

[1] l is the total number of entity types in the source and target domains, such as {O, PER, LOC, ORG, MISC}. Our discussion also makes sense in the case that source domain and target domain have different entity types.

The original theorem, however, concerns only one model h for transfer learning. In our supervised settings, in contrast, their CRF layers are specific to the source and target domains, respectively. Below we use h^* to denote our overall model with shared multi-cell LSTM model and domain-specific CRF layers. Further, we use h_1 to denote the target domain subsystem that consists of the shared multi-cell LSTM model and the target-specific CRF layer, and h_2 to denote its source counterpart. Theorem 1 can be extended to our settings as follows:

Lemma 1 If $\epsilon_\alpha(h^*) = \alpha\epsilon_T(h_1) + (1-\alpha)\epsilon_S(h_2)$, then:

$$\epsilon_T(h_1) \le 2\epsilon_\alpha(h^*) + (1-\alpha)\left(\tfrac{3}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_S, \mathcal{D}_T\right) + \lambda^*\right)$$

Proof. The proof is mainly based on trangle inequalities, see **Appendix A** for details. □

Considering that the upper bounds of $\epsilon_T(h)$ ($\epsilon_T(h_1)$), $\epsilon_\alpha(h)$ ($\epsilon_\alpha(h^*)$) and λ (λ^*) are small when training converges, our goal is to reduce $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T)$. In particular, we define a model h is a composition function $h = g \circ f$, where f represents the multi-cell LSTM model and g represents the CRF layer, $\circ$ denotes function composition. We assume h' and h'' share the same multi-cell LSTM model, namely $h' = g' \circ f$ and $h'' = g'' \circ f$, we have

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) = 2 \sup_{g', g'' \in \mathcal{G}} \Big| \Pr_{x\sim\mathcal{D}_S}\left[g' \circ f(x) \neq g'' \circ f(x)\right] - \Pr_{x\sim\mathcal{D}_T}\left[g' \circ f(x) \neq g'' \circ f(x)\right] \Big|$$

To obtain the supremum of the right term, we may wish to assume that both g' and g'' can classify correctly in the source domain, then

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) \approx 2 \sup_{g', g'' \in \mathcal{G}} \Big| \Pr_{x\sim\mathcal{D}_T}\left[g' \circ f(x) \neq g'' \circ f(x)\right] \Big|$$

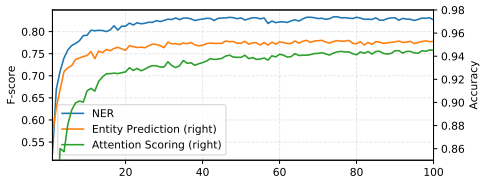
The optimization objective is as follows:

$$\min_{f\in\mathcal{F}} \sup_{g', g'' \in \mathcal{G}} \Big| \Pr_{x\sim\mathcal{D}_T}\left[g' \circ f(x) \neq g'' \circ f(x)\right] \Big|$$

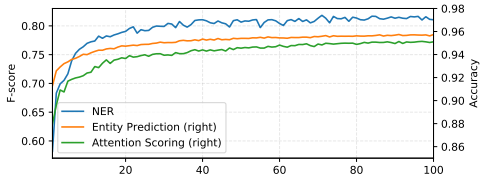
Aiming to $\min_{f\in\mathcal{F}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T)$, we decompose the unified feature space into several entity typed distributions using multi-cell LSTM, resulting in that source- and target-domain features belonging to the same entity type are clustered together. The proof is mainly based on the cluster assumption (Chapelle and Zien, 2005), which is equivalent to the low density separation assumption, states that the decision boundary should lie on a low-density region. According to the cluster assumption, both g' and g'' tend to cross the low-density regions in the shared

Dataset	Entity Type	Size	Train	Dev	Test
CoNLL-2003	PER, LOC	#Sentence	15.0K	3.5K	3.7K
	ORG, MISC	#Entity	23.5K	5.9K	5.6K
Broad Twitter	PER, LOC	#Sentence	6.3K	1.0K	2.0K
	ORG	#Entity	8.8K	1.7K	4.4K
Twitter	PER, LOC	#Sentence	4.3K	1.4K	1.5K
	ORG, MISC	#Entity	7.5K	2.5K	2.5K
BioNLP13PC	CHEM, CC	#Sentence	2.5K	0.9K	1.7K
	GGP, etc.	#Entity	7.9K	2.7K	5.3K
BioNLP13CG	CHEM, CC	#Sentence	3.0K	1.0K	1.9K
	GGP, etc.	#Entity	10.8K	3.6K	6.9K
CBS News	PER, LOC	#Sentence	-	-	2.0K
	ORG, MISC	#Entity	-	-	3.4K

Table 1: Statistic of datasets.



(a) BioNLP.



(b) Twitter.

Figure 2: Performances of the main NER and auxiliary tasks against the total number of training iteratons.

feature space of both source and target domains. This results in $\Pr_{x\sim\mathcal{D}_T}[g' \circ f(x) \neq g'' \circ f(x)] \approx \Pr_{x\sim\mathcal{D}_S}[g' \circ f(x) \neq g'' \circ f(x)] \approx 0$, which well meets the above optimization objecive.

4 Experiments

4.1 Experimental Settings

Datasets. We take six publicly available datasets for experiments, including BioNLP13PC and BioNLP13CG (Nédellec et al., 2013), CoNLL-2003 English dataset (Sang and Meulder, 2003), Broad Twitter dataset (Derczynski et al., 2016), Twitter dataset (Lu et al., 2018) and CBS SciTech News dataset (Jia et al., 2019). Statistics of the datasets are shown in Table 1. In unsupervised domain adaptation experiments, 398,990 unlabeled sentences from CBS SciTech News collected by Jia et al. (2019) are used for target-domain LM training, a named entity dictionary from Web resource collected by Peng et al. (2019) is used for target-domain auxiliary tasks training.

The CoNLL-2003, Twitter and CBS News have the same four types of entities, namely PER (person), LOC (location), ORG (organization) and MISC (miscellaneous). The Broad Twitter dataset consists of three types: PER, LOC and ORG. BioNLP13CG mainly consists of five types, namely CHEM (simple chemical), CC (cellular component), GGP (gene and gene product), SPE (species) and CELL (cell), BioNLP13PC mainly consists of three types: CHEM, CC and GGP.

Hyperparameters. We choose NCRF++ (Yang and Zhang, 2018) for developing the models. The multi-task baselines are based on Jia et al. (2019). Our hyperparameter settings largely follow Yang et al. (2018); word embeddings for all models are initialized with PubMed 200-dimension vectors (Chiu et al., 2016) in BioNLP experiments and GloVe 100-dimension vectors (Pennington et al., 2014) in other experiments. All word embeddings are fine-tuned during training. Character embeddings are randomly initialized.

4.2 Development Experiments

Figure 2 shows the performances of the main target-domain NER task and the auxiliary entity prediction and attention scoring tasks on the development sets of BioNLP13CG and Twitter when the number of training iterations increases. As can be seen from the figure, all the three tasks have the same trend of improvement without potential conflicts between tasks, which shows that all the three tasks take the feature space of the same form.

4.3 Supervised Domain Adaptation

We conduct supervised domain adaptation on BioNLP dataset, Broad Twitter dataset and Twitter dataset, respectively. In particular, for the BioNLP dataset, BioNLP13CG is used as the target-domain NER dataset and BioNLP13PC as the source-domain dataset. These two datasets have some different entity types. In the Broad Twitter dataset, Broad Twitter is used as the target-domain dataset and the CoNLL-2003 as the source-domain dataset. These two datasets have a different entity type MISC. In the Twitter dataset, Twitter is used as the target-domain dataset and the CoNLL-2003 as the source-domain dataset. These two datasets have the same entity types. The overall results are listed in Table 2.

Target-domain only settings. In comparison with target-domain only models BILSTM and MULTI-

Methods	Datasets					
	BioNLP		Broad Twitter		Twitter	
	F_1	#Params	F_1	#Params	F_1	#Params
Crichton et al. (2017)	78.90	-	-	-	-	-
Lu et al. (2018)	-	-	-	-	80.75	-
Wang et al. (2019)	82.48	-	-	-	-	-
Jia et al. (2019)	79.86	-	-	-	-	-
BiLstm+Elmo (Peters et al., 2018)	-	-	76.48	94,590K	82.83	94,631K
BiLstm+BioElmo (Peters et al., 2018)	85.61	94,605K	-	-	-	-
Bert-base (Devlin et al., 2019)	-	-	77.28	108M	83.77	108M
BioBert-base (Lee et al., 2020)	85.72	108M	-	-	-	-
BiLstm	79.24	304K	72.98	210K	77.18	211K
Multi-cell Lstm	78.76	2,704K	72.54	641K	77.05	743K
Multi-Task (Lstm)	81.06	309K	73.84	214K	79.55	215K
Multi-Task (Lstm)[Repro]*	81.45	312K	73.82	214K	79.90	215K
Multi-Task+Pgn	81.17	4,533K	73.70	3,238K	80.07	3,239K
Multi-Task+Grad	81.63	447K	74.12	342K	79.72	344K
Ours	83.12†	2,929K	74.82†	827K	81.37†	828K
Ours+Elmo/BioElmo	86.65	105M	76.36	97,090K	84.31	97,091K
Ours+Bert-base/BioBert-base	**86.96**†‡	117M	**78.43**†‡	111M	**85.80**†‡	111M

Table 2: Results on three few-shot datasets. * indicates that we reproduce the baseline bi-directional Lstm in a similar way to our model for fair comparisons. † indicates statistical significance compared to target-domain settings and cross-domain settings with $p < 0.01$ by t-test. ‡ indicates statistical significance compared to LM pre-training based methods with $p < 0.01$ by t-test.

cell Lstm, all of the multi-task models obtain significantly better results on all of the three datasets. This shows the effectiveness of multi-task learning in few-shot transfer.

Cross-domain settings. We make comparisons with the traditional parameter sharing mechanism Multi-Task(Lstm) (Yang et al., 2017) together with two improved methods, Multi-Task+Pgn (Jia et al., 2019), which adds an parameter generation networks (Pgn) to generate parameters for source- and target-domain LSTMs and Multi-Task+Grad (Zhou et al., 2019), which adds a generalized resource-adversarial discriminator (Grad) and leverages adversarial training. The results show that our method can significantly outperform these multi-task methods on the same datasets, which shows the effectiveness of our multi-cell structure in cross-domain settings.

Comparison with the state-of-the-art models. Results show that our model outperforms cross-domain method of Jia et al. (2019), cross-type method of Wang et al. (2019) and methods using addition features (Crichton et al., 2017; Lu et al., 2018). Recently, LM pre-training based methods such as Elmo/BioElmo (Peters et al., 2018), Bert (Devlin et al., 2019) and BioBert (Lee et al., 2020) achieve state-of-the-art results on NER. However, these methods use additional large-scale language resources, thus it is unfair to make direct comparisons with our method. Thus we leverage the outputs of LM pre-training methods as contextualized word embeddings. In particular, we use the same batch size as our method and the Adam optimizer with an initial learning rate 3e-5 in Bert fine-tuning baselines. Results show that our method benifits from these LM pre-training output features and outperforms these LM pre-training based methods.

Methods	F_1	#Params	#Raw
Jia et al. (2019)	73.59	12,916K	18,474K
Bert-base (Devlin et al., 2019)	74.23	108M	3,700M
BiLSTM	70.73	211K	-
Multi-cell Lstm	70.03	743K	-
BiLstm+Lm	71.30	211K	1,931K
BiLstm+Lm+Dict	72.49	212K	1,931K
Multi-cell Lstm+Lm	72.81	743K	1,931K
Multi-cell Lstm+Lm(All)	73.56	743K	8,664K
Multi-cell Lstm+Lm+Dict	**75.19**†	743K	1,931K

Table 3: Results on CBS News datasets. #Raw indicates number of words in raw data used in the experiment. † indicates statistical significance compared with all of the baselines with $p < 0.01$ by t-test.

4.4 Unsupervised Domain Adaptation

We conduct unsupervised domain adaptation on the CBS SciTech News test set, using CoNLL-2003 as the source-domain dataset. The overall results are listed in Table 3.

Adding target-domain LM training. Only using the source-domain NER data, BiLstm and Multi-cell Lstm give comparable results, 70.73% F_1 and 70.03% F_1, respectively. In comparison with the source-domain only models, all of the models

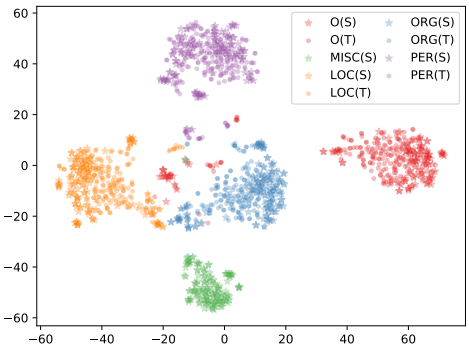


Figure 3: t-SNE visualization of ET cell states $\{\mathbf{c}_k\}_{k=1}^{l}$ on the CoNLL-2003 test set and Broad Twitter test set, differentiated by signal star and dot, respectively. Different entity types are represented by different colours.

using LM obtain significantly better results, which shows the effectiveness of using target-domain LM in zero-shot transfer. When using the same amount of target-domain raw data as Jia et al. (2019), The result of MULTI-CELL LSTM+LM(ALL) is comparable to the state-of-the-art (Jia et al., 2019) (73.56% F_1 v.s. 73.59% F_1), which uses both source-domain LM and target-domain LM. This shows the effectiveness of multi-cell structure for zero-shot transfer.

Adding a named entity dictionary. With the named entity dictionary collected by Peng et al. (2019), the results show a significant improvement (75.19% F_1 v.s. 72.81% F_1). To make fair comparison, we add the entity dictionary information to BILSTM+LM by doing an entity type prediction task together with the target-domain LM. BILSTM+LM+DICT achieves better result than BILSTM+LM (72.49% F_1 v.s. 71.30% F_1), but it still cannot be comparable to our results. This shows that the auxiliary tasks can help learn entity knowledge from raw data, even if the named entity dictionary can not label all entities in a sentence.

4.5 Analysis

Visualization. In the proposed multi-cell LSTM, both ET cells and C cell play important roles in constructing a shared feature spaces across domains. We visualize feature spaces of ET cells and C cell in the Broad Twitter experiments.

Figure 3 uses t-SNE (Maaten and Hinton, 2008) to visualize the ET cell states $\{\mathbf{c}_k\}_{k=1}^{l}$. From the figure we can see that different ET cells can generate different feature distributions (gathering in different clusters of different colours), and states

Entity group		CHEM	CC	GGP	CELL	SPE	**All**
Is in Source?		✓	✓	✓	×	×	-
LSTM	F_1	69.13	78.29	82.79	85.00	79.08	79.23
	Δ	-	-	-	-	-	-
MULTI	F_1	73.57	79.67	85.83	85.14	79.47	81.05
	Δ	+4.44	+1.38	+3.04	+0.14	+0.39	+1.82
Ours	F_1	74.95	80.00	86.67	87.10	81.92	82.70
	Δ	+5.82	+1.71	+3.88	+2.10	+2.84	+3.47

Table 4: Fine-grained comparisons on BioNLP.

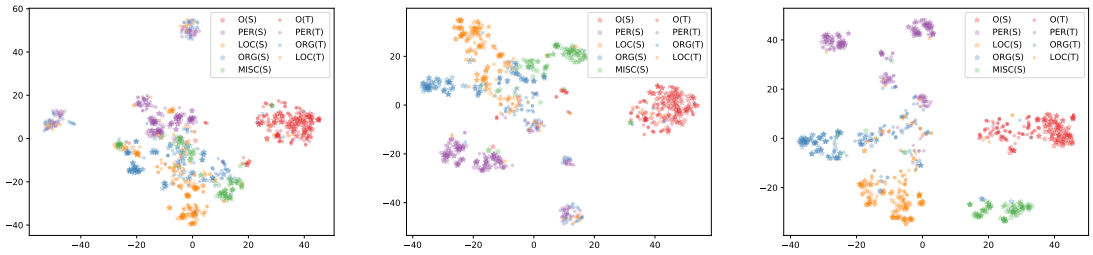
of the same ET cell gather together across domains. This indicates that our model can learn cross-domain entity typed knowledge with the help of ET cells, which are more robust across domains.

Figure 4 visualize the hidden vectors of the target-domain only baseline, the multi-task baseline and the proposed model. From the figure, we can see that both the multi-task baseline and ours can obtain similar feature distributions across domains compared with the target-domain only baseline. In comparison with the multi-task baseline, our model also shows strong matches across domains in an entity type level, which can better narrow the gap between source and target domains as discussed in §3.4.

Fine-grained comparison. We make fine-grained comparisons between our model and the multi-task baseline on the BioNLP dataset, aiming to show how our model achieves better results on the entity type level. Following Crichton et al. (2017) and Jia et al. (2019), we study five well studied entity groups (not including all entity types) in BioNLP13CG. As shown in Table 4, both MULTI (Multi-Task baseline) and Ours achieve significant F_1 improvement over the target-domain only baseline LSTM on the biochemistry entity groups that appear in both the target and the source datasets, such as CHEM, CC and GGP, which is consistent with intuition.

But for biology entity groups not appearing in the source dataset, such as CELL and SPE, MULTI using traditional parameter sharing hardly improves the performances (+0.14% F_1 for CELL and +0.39% F_1 for SPE v.s. +1.82% F_1 for All). In contrast, Ours achieves relatively strong improvements (+2.10% F_1 for CELL and +2.84% F_1 for SPE). This benefits from the distinct feature distributions across entity types by the multi-cell LSTM structure, which can effectively prevent the confusions drawn in a unified feature space.

Ablation study. We conduct ablation studies on auxiliary tasks and model parameters. The results



(a) Target-domain only. (b) Multi-Task baseline. (c) Ours.

Figure 4: t-SNE visualization of hidden vectors on the CoNLL-2003 test set and Broad Twitter test set, represented by signal star and dot, respectively. Different entity types are represented by different colours.

Methods	Datasets					
	BioNLP		Broad Twitter		CBS News	
	F_1	Δ	F_1	Δ	F_1	Δ
OURS	**83.15**	-	**74.82**	-	**75.19**	-
$-\mathcal{L}_{ent}$	82.71	-0.44	73.97	-0.85	74.95	-0.24
$-\mathcal{L}_{atten}$	81.65	-1.50	73.25	-1.57	73.04	-2.15
$-\mathcal{L}_{ent}-\mathcal{L}_{atten}$	81.74	-1.41	73.64	-1.18	72.59	-2.60
BILSTM-BASED	81.06	-2.09	73.84	-0.98	72.49	-2.70
STACKED BILSTMS	80.61	-2.54	73.86	-0.96	69.62	-5.57
HIDDEN EXPANSION	80.32	-2.83	72.34	-2.48	73.17	-2.02

Table 5: Ablation studies on BioNLP, Broad Twitter and CBS SciTech News datasets.

Sentence	The World Health Organization (WHO) describes Nipah infection as a "newly emerging zoonosis that causes severe disease in both animals and humans."
BILSTM	The World Health Organization (WHO (O) describes Nipah ORG
BILSTM+LM	The World Health Organization (WHO (O) describes Nipah MISC
Ours	The World Health Organization (WHO (ORG) describes Nipah MISC

Table 6: Example from CBS News test. Red and green represent incorrect and correct entities, respectively.

are listed in Table 5.

Auxiliary tasks. When we only ablate $\mathcal{L}_{ent}$, the results on all of the three datasets suffer significant decline (-0.44% F_1 on BioNLP dataset, -0.85% F_1 on Broad Twitter dataset and -0.24% F_1 on CBS News dataset, respectively). When we only ablate $\mathcal{L}_{atten}$, the results on all of the three datasets suffer significant decline (over -1.5% F_1 on all of the three datasets). When we both ablate $\mathcal{L}_{ent}$ and $\mathcal{L}_{atten}$, our model achieves similar results as the BILSTM-BASED baseline. This indicates that domain transfer of our model depends heavily on both auxiliary tasks.

Number of parameters. We use two strategies to make the number of parameters of BILSTM-BASED baseline comparable to that of our model: (i) STACKED BILSTMS, stacking multi-layer BiLSTMs and enlarging the hidden size. (ii) HIDDEN EXPANSION, with similar model structure, just enlarging the hidden size. Our model still significantly outperforms these baselines, which shows that the effects of our model do not arise from a larger number of parameters.

Case study. Table 6 shows a case study, "WHO" is an organization and "Nipah" is a virus. Without using target-domain raw data, BI-LSTM baseline miclassifies "Nipah" as ORG. Both Ours and BILSTM+LM give the correct results because this entity is mentioned in raw data. Using the multi-cell structure, our method learns the pattern "ORG, O, ORG, O" from source data without confusions by target-domain specific entities, thus Ours recognizes "WHO" correctly.

5 Conclusion

We have investigated a multi-cell compositional LSTM structure for cross-domain NER under the multi-task learning strategy. Theoretically, our method benefits from the distinct feature distributions for each entity type across domains. Results on a range of cross-domain datasets show that multi-cell compositional LSTM outperforms BiLSTM under the multi-task learning strategy.

Acknowledgments

We thank the anonymous reviewers for their helpful comments and suggestions. We gratefully acknowledge funding from the National Natural Science Foundation of China (NSFC No.61976180) and the Westlake University and Bright Dream Joint Institute for Intelligent Robotics. Yue Zhang is the corresponding author.

References

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman

Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2018. Adversarial transfer learning for chinese named entity recognition with self-attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 182–192. Association for Computational Linguistics.

Olivier Chapelle and Alexander Zien. 2005. Semi-supervised classification by low density separation. In *AISTATS*, volume 2005, pages 57–64. Citeseer.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th workshop on biomedical natural language processing*, pages 166–174. Association for Computational Linguistics.

Koby Crammer, Michael Kearns, and Jennifer Wortman. 2008. Learning from multiple sources. *Journal of Machine Learning Research*, 9(Aug):1757–1774.

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.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: 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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Bahdanau Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*, pages 1–15.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-domain ner using cross-domain language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2464–2474. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

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.

Di Lu, Leonardo Neves, Vitor Carvalho, Ning Zhang, and Heng Ji. 2018. Visual attention model for name tagging in multimodal social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1990–1999. 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.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9:2579–2605.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1654. Association for Computational Linguistics.

Raymond J Mooney and Razvan C Bunescu. 2006. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7. Association for Computational Linguistics.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2409–2419, Florence, Italy. Association for Computational Linguistics.

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 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, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Ana-Maria Popescu, Marco Pennacchiotti, and Deepa Paranjpe. 2011. Extracting events and event descriptions from twitter. In *Proceedings of the 20th international conference companion on World wide web.*, pages 105–106.

Juan Diego Rodriguez, Adam Caldwell, and Alex Liu. 2018. Transfer learning for entity recognition of novel classes. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1974–1985. Association for Computational Linguistics.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. 2019. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752.

Zhenghui Wang, Yanru Qu, Liheng Chen, Jian Shen, Weinan Zhang, Shaodian Zhang, Yimei Gao, Gen Gu, Ken Chen, and Yong Yu. 2018. Label-aware double transfer learning for cross-specialty medical named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1–15. Association for Computational Linguistics.

Huiyun Yang, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2019. Fine-grained knowledge fusion for sequence labeling domain adaptation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4188–4197. 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. Association for Computational Linguistics.

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*.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. 2019. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3461–3471. Association for Computational Linguistics.

A Proof of Lemma 1

Proof. Given the precondition $\epsilon_\alpha(h^*) = \alpha\epsilon_T(h_1) + (1-\alpha)\epsilon_S(h_2)$, we use the trangle inequality as follows:

$$\begin{aligned}
&\epsilon_T(h_1) - \epsilon_\alpha(h^*) \le \big|\epsilon_\alpha(h^*) - \epsilon_T(h_1)\big| \\
=&(1-\alpha)\big|\epsilon_S(h_2) - \epsilon_T(h_1)\big| \\
\le&(1-\alpha)\Big[\big|\epsilon_S(h_2) - \epsilon_S(h_1,h_2)\big| \\
&\qquad + \big|\epsilon_S(h_1,h_2) - \epsilon_T(h_1,h_2)\big| \\
&\qquad + \big|\epsilon_T(h_1,h_2) - \epsilon_T(h_1)\big|\Big]
\end{aligned}$$

The trangle inequality in Crammer et al. (2008) states that for a class of models $\mathcal{F}$ and expected error function ϵ if for all $g_1, g_2, g_3 \in \mathcal{F}$, we have $\epsilon(g_1,g_2) \le \epsilon(g_1,g_3) + \epsilon(g_2,g_3)$. Following the above formular and the definition of $d_{\mathcal{H}\Delta\mathcal{H}}(\cdot,\cdot)$, we can further obtain:

$$\begin{aligned}
&\epsilon_T(h_1) - \epsilon_\alpha(h^*) \\
\le&(1-\alpha)\Big[\epsilon_S(h_1) + \epsilon_T(h_2) \\
&\qquad + \big|\epsilon_S(h_1,h_2) - \epsilon_T(h_1,h_2)\big|\Big] \\
\le&(1-\alpha)\Big[\epsilon_S(h_1) + \epsilon_T(h_2) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T)\Big]
\end{aligned}$$

Given the precondition $\epsilon_\alpha(h^*) = \alpha\epsilon_T(h_1) + (1-\alpha)\epsilon_S(h_2)$, we consider two UDA settings: (i) domain T with hypothesis h_1 as the source; (ii) domain S with hypothesis h_2 as the source. Using Theorem 1 under $\alpha = 0$, we can obtain:

$$\epsilon_S(h_1) \le \epsilon_T(h_1) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T) + \lambda_1$$

$$\epsilon_T(h_2) \le \epsilon_S(h_2) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T) + \lambda_2$$

As the common setting of transfer learning, we set $1>\alpha \geq \frac{1}{2}$ and then $\frac{\alpha}{1-\alpha} \geq 1$, further obtaining:

$$\epsilon_S(h_1) \leq \frac{\alpha}{1-\alpha}\epsilon_T(h_1) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \lambda_1$$

Using these conclusions to the previous inequalities, we have:

$$\begin{aligned}
&(1-\alpha)\left[\epsilon_S(h_1) + \epsilon_T(h_2) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T)\right] \\
\leq &\alpha\epsilon_T(h_1) + (1-\alpha)\epsilon_S(h_2) \\
&+ (1-\alpha)\left[\frac{3}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \lambda_1 + \lambda_2\right]
\end{aligned}$$

Setting $\lambda^* = \lambda_1 + \lambda_2$, which is the shared error of ideal joint hypothesis and use the precondition, $\epsilon_\alpha(h^*) = \alpha\epsilon_T(h_1) + (1-\alpha)\epsilon_S(h_2)$, we have

$$\begin{aligned}
&\epsilon_T(h_1) - \epsilon_\alpha(h^*) \\
\leq &(1-\alpha)\left[\epsilon_S(h_1) + \epsilon_T(h_2) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T)\right] \\
\leq &\epsilon_\alpha(h^*) + (1-\alpha)\left[\frac{3}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \lambda^*\right]
\end{aligned}$$

Finally, we obtain the **Lemma 1**:

$$\begin{aligned}
\epsilon_T(h_1) \leq &2\epsilon_\alpha(h^*) \\
&+ (1-\alpha)\left[\frac{3}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \lambda^*\right]
\end{aligned}$$

□