Domain-Adapted Dependency Parsing for Cross-Domain Named Entity Recognition

Chenxiao Dou[1], **Xianghui Sun**[2], **Yaoshu Wang**[*3], **Yunjie Ji**[2], **Baochang Ma**[2], **Xiangang Li**[2]

[1]Nanhu Academy of Electronics and Information Technology
[2]Beike
[3]Shenzhen Institute of Computing Sciences, Shenzhen University
douchenxiao@cnaeit.com, {sunxianghui002,jiyunjie001,mabaochang001,lixiangang002}@ke.com, yaoshuw@sics.ac.cn

Abstract

In recent years, many researchers have leveraged structural information from dependency trees to improve Named Entity Recognition (NER). Most of their methods take dependency-tree labels as input features for NER model training. However, such dependency information is not inherently provided in most NER corpora, making the methods with low usability in practice. To effectively exploit the potential of word-dependency knowledge, motivated by the success of Multi-Task Learning on cross-domain NER, we investigate a novel NER learning method incorporating cross-domain Dependency Parsing (DP) as its auxiliary learning task. Then, considering the high consistency of word-dependency relations across domains, we present an unsupervised domain-adapted method to transfer word-dependency knowledge from high-resource domains to low-resource ones. With the help of cross-domain DP to bridge different domains, both useful cross-domain and cross-task knowledge can be learned by our model to considerably benefit cross-domain NER. To make better use of the cross-task knowledge between NER and DP, we unify both tasks in a shared network architecture for joint learning, using Maximum Mean Discrepancy (MMD). Finally, through extensive experiments, we show our proposed method can not only effectively take advantage of word-dependency knowledge, but also significantly outperform other Multi-Task Learning methods on cross-domain NER. Our code is open-source and available at https://github.com/xianghuisun/DADP.

Introduction

Named Entity Recognition (NER) is the foundation for many tasks of information extraction, aiming to locate and identify named entities in natural-language sentences, such as *Person* and *Location* (Li et al. 2022). The extracted named entities carry rich semantic information which plays an important role in downstream NLP tasks, such as Entity Resolution and Question Answering (Li et al. 2022). In practice, the main challenge of NER comes from the sparsity issue of data annotation, which may cost unaffordable efforts from human experts to label named entities in text corpora. Due to the high cost of manual labelling, cross-domain NER has attracted widespread attention from academy and industry. In recent years, Multi-Task Learning (MTL) (Caruana 1997) is widely used in solving cross-domain NER. Researchers claim that cross-domain NER task can benefit greatly from the contextual representations learned from related cross-domain NLP tasks, such as Entity Type Prediction (Qian et al. 2021) and Language Modeling (Liu et al. 2018). Shared knowledge across different tasks and different domains is regarded as the key factor to the success of Multi-Task Learning on cross-domain NER.

Dependency Parsing (DP) (Dozat and Manning 2017) is a classical NLP task to retrieve a syntactic structure from sentence, named as Dependency Tree which discloses long-distance and pairwise-relation information of words. Many studies show the effectiveness of adopting dependency-tree information on improving NER (Jie, Muis, and Lu 2017; Li et al. 2021; Guo, Zhang, and Lu 2019). Compared to other auxiliary NLP tasks commonly used in Multi-Task Learning, DP seems to be more promising in introducing shared cross-task knowledge to NER, because the boundary of a named entity often corresponds to the boundary of a sub dependency tree in sentence-level. For example, in Figure 1, given the named entity *The Cape of Good Hope*, we can easily observe that its start position and end position are the same to the start position and end position of the sub dependency tree rooted at *Cape*. Certainly, such boundary knowledge learned from DP has great value to be used across tasks to help finding named-entity boundary. However, there exists a practical problem to join DP and NER, that DP annotation is not an inherent part of natural-language sentences and is barely provided in NER corpora. Apparently, it is unrealistic to manually label dependency relations for target data, which is time-consuming and labor-intensive.

To address the DP labelling problem, a promising solution is to adopt cross-domain DP as the auxiliary task to jointly learn with NER. By utilizing Domain Adaptation (DA) techniques (Wang and Deng 2018), cross-domain DP is able to transfer learned DP knowledge from high-resource domains to low-resource domains. In this way, with limited DP labels, useful cross-task knowledge from cross-domain DP can be acquired for NER. Furthermore, with relatively invariant syntactic features learned through DP domain adaptation, cross-domain DP can also be used to bridge different domains for cross-domain NER. Intuitively, compared to other NLP tasks, such as NER whose data distribution

*Yaoshu Wang is the corresponding author.

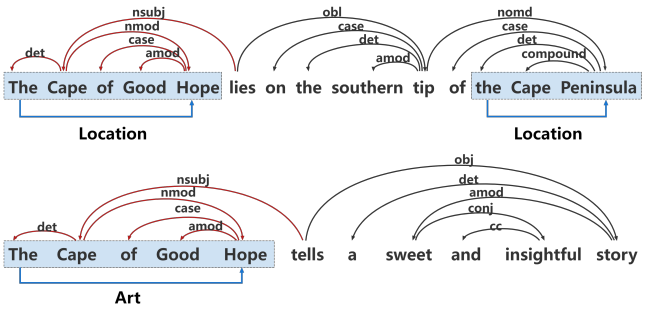


Figure 1: Example sentences are given with dependency relations on the top and named-entity spans on the bottom. The first sentence comes from Geography Domain and the second sentence comes from Entertainment Domain.

may shift frequently with domain changing, DP usually has a relatively stable distribution across domains. It is because that no matter what data domain is, sentences of the same language usually share the same grammar and syntax, leading to the consistency of DP across different domains. For illustration, in Figure 1, the phrase *The Cape of Good Hope* may refer to a *Location* entity in Geography Domain and change to an *Art* entity in Entertainment Domain, but its dependency-tree structural information remains consistent across the two domains. This example shows that cross-domain grammar knowledge sometimes is more stable and useful than cross-domain semantic knowledge on finding named entities, when the topics of two domains are significantly different. Therefore, attracted by both cross-task and cross-domain knowledge of DP, we are interested in utilizing cross-domain DP to benefit cross-domain NER in MTL framework.

Motivated by the above, in this paper, we propose a novel MTL framework for cross-domain NER, using cross-domain DP as the auxiliary task, and our contributions are summarized as follows. First, targeting the sparsity issue of dependency labels, we learn the DP knowledge in high-resource labelled domains, and take the idea of domain adaptation, to transfer the learned knowledge to low-resource unlabelled domains. Second, to make the learned DP representations adapted to both the source and target domains without supervision, Maximum Mean Discrepancy (MMD) (Gretton et al. 2012) is then used as the metric to minimize the difference between DP representation distributions of the two domains. Third, for better use of the cross-task knowledge between NER and DP, we unify both tasks in a shared network architecture for joint learning, to obtain more representative semantic features. In this way, at the top of the network structure, a biaffine classifier (Dozat and Manning 2017) is employed to identify named entities, based on the learned common and unique knowledge from NER and DP. Finally, through extensive experiments, we show our proposed method can not only effectively take advantage of dependency knowledge, but also significantly outperform other Multi-Task Learning methods on cross-domain NER.

Related Works

Recently, utilizing dependency-tree information in NER models seems to be a broadly adopted strategy. For example, (Jie, Muis, and Lu 2017) exploits the global structured information of dependency trees to guide NER learning. To capture exact interaction information of dependency trees, (Xu et al. 2021) proposes a novel LSTM-based NER method. (Li et al. 2021) and (Guo, Zhang, and Lu 2019) adopt the information of dependency-guided graph convolutional networks to improve NER. However, all of the methods inevitably require dependency information as indispensable inputs, leading to a practical problem since such annotations are costly to obtain.

Multi-Task Learning is a promising technique widely used in cross-domain NER field, aiming to jointly learn NER with other NLP tasks to improve the semantic representations. (Qian et al. 2021) takes sentence-level named type prediction as the auxiliary task for cross-domain NER. (Xiao et al. 2019) introduces a similarity-based NER method, which incorporates an auxiliary classifier to distinguish entity words from non-entity words. Character-level language modeling is used to help cross-domain NER learning in (Liu et al. 2018). (Jia, Xiao, and Zhang 2019) employs cross-domian LM as a bridge for NER adaptation across domains. (Jia and Zhang 2020) utilizes a multi-cell compositional LSTM structure to detect entity type across domains to improve cross-domain NER. Regrettably, to our knowledge, few studies have used DP as the auxiliary task to jointly learn with NER. It is mainly because that NER corpora usually provides no information about dependency relations. Therefore, to incorporate NER and DP in a Multi-Task Learning framework, an unsupervised manner is needed.

Unsupervised Domain Adaptation (Wang and Deng 2018) targets to transfer knowledge from high-resource labelled domains to low-resource unlabelled domains. MMD (Gretton et al. 2012) is often adopted as the metric to measure the difference of representation distributions in the process of Unsupervised Domain Adaptation, which has achieved encouraging results in many NLP tasks. For instance, (Zhang et al. 2021) applies MMD to solve cross-domain NER problem and (Bista et al. 2020) uses MMD to adapt cross-domain knowledge for document summarization. Motivated by the adaptation idea, in this paper, we exploit the power of MMD to solve the problem of dependency annotations, and combine NER and DP in the same joint-learning framework.

Preliminary of MMD

Maximum Mean Discrepancy is a non-parametric statistical metric to measure the difference between two data distribution p and q, in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ with a characteristic kernel k. MMD has an important property that $p = q$ if and only if the MMD measured on p and q is equal to zero. Given two datasets X and Y independently and identically sampled from p and q with sizes of M and N, the corresponding empirical MMD denoted as

$\text{MMD}(X, Y)$ can be written as the flowing:

$$\text{MMD}(X, Y) = ||\frac{1}{M}\sum_{x\in X}\phi(x) - \frac{1}{N}\sum_{y\in Y}\phi(y)||_{\mathcal{H}} \quad (1)$$

where $\phi(\cdot)$ is the nonlinear feature mapping that induces $\mathcal{H}$ and takes the canonical form $\phi(x) = k(x, \cdot)$ (Hearst et al. 1998). Gaussian Kernel is used as our characteristic kernel k. Accordingly, in terms of kernel functions, the squared empirical MMD can be written as:

$$\begin{aligned}\text{MMD}^2(X, Y) = &\frac{1}{M^2}\sum_{\substack{x\in X\\ x'\in X}} k(x, x') \\ &- \frac{2}{MN}\sum_{\substack{x\in X\\ y\in Y}} k(x, y) + \frac{1}{N^2}\sum_{\substack{y\in Y\\ y'\in Y}} k(y, y')\end{aligned} \quad (2)$$

In this paper, we use the above MMD definition to measure the distribution difference between source and target domains in the learned representation spaces.

Method

The overall structure of the proposed method is given in Figure 2, which is designed for a joint-learning NER framework using DP as the auxiliary task. The framework unifies NER and DP with the same neural network to take the advantage of cross-task and cross-domain NLP knowledge. In the process of learning, MMD loss is considered to transfer DP knowledge from source-domain dataset to target-domain dataset.

Shared Layer

In the proposed model, to make use of NLP cross-task knowledge, we use the same bottom-representation layers for both the main NER task and the auxiliary DP task. On the bottom, we use BERT (Devlin et al. 2019) as the fundamental encoder to extract inputs' semantic features. As both tasks share the same bottom-representation layers, the input sentences come from either source DP dataset $\mathcal{D}_s$ or target NER dataset $\mathcal{D}_t$. Given an input sentence $x = [x_1, x_2, ..., x_l]$ with length l, we first feed each word x_i into the BERT module to obtain its word embedding w_i. After word tokens converts into embeddings $w = [w_1, w_2, ..., w_l]$, we apply a Bidirectional LSTM (BiLSTM) (Graves and Schmidhuber 2005) to the word vectors. Then, for w_i, its hidden outputs produced by the forward and backward LSTMs can be formally written as:

$$\begin{aligned}\overrightarrow{h}_i &= \text{LSTM}(\overrightarrow{h}_{i-1}, w_i, \theta_f) \\ \overleftarrow{h}_i &= \text{LSTM}(\overleftarrow{h}_{i+1}, w_i, \theta_b),\end{aligned} \quad (3)$$

where θ_f and θ_b are corresponding parameters for the forward and backward directions. Finally, the concatenation of the forward and backwards hidden outputs is taken as the output vector of BiLSTM:

$$h_i = \overrightarrow{h_i} \oplus \overleftarrow{h_i}, \quad (4)$$

where $\oplus$ represents vector concatenation.

Task Layer

DP Module Our DP module is incorporated to learn and extract dependency information of words on assisting NER. Inspired by the work of (Dozat and Manning 2017), we also exploit the biaffine classifier to identify dependency relations in sentences. After the BiLSTM layer, to strip away redundant information that is not task-related, two smaller MLP layers are connected to distill informative features for DP-start (head) representation and DP-end (dependent) representation. With the two layers, given the i-th input word, we denote its representation vector for DP-start as s_i^{dp} and for DP-end as e_i^{dp}. Then, the dependency relation, starting at the i-th word and ending at the j-th, is represented as a pair (s_i^{dp}, e_j^{dp}). At last, for all candidate pairs, we introduce a biaffine classifier to score their relation types in one $l \times l \times m$ tensor r^{dp}, where m is the number of dependency relations including *non-dependency*. Formally, we have:

$$\begin{aligned}s_i^{dp} &= \text{MLP}^{\text{dp-start}}(h_i) \\ e_j^{dp} &= \text{MLP}^{\text{dp-end}}(h_j) \\ r_{i,j}^{dp} &= s_i^{dp\top}\mathbf{U}^{dp}e_j^{dp} \\ &\quad + \mathbf{W}^{dp}(s_i^{dp} \oplus e_j^{dp}) + b^{dp},\end{aligned} \quad (5)$$

where s_i^{dp} and e_j^{dp} have the same dimension d, $\mathbf{U}^{dp}$ is a $d \times m \times d$ tensor, $\mathbf{W}^{dp}$ is a tensor of $2d \times m$ shape and b^{dp} is the bias. The relation type with the maximum score in $r_{i,j}^{dp}$ is taken as the predicted dependency type.

It is remarkable that in our method, the DP module is not used to predict dependency annotations as input features for NER, but performs to learn the cross-task knowledge and improve the semantic representations of our model. Inspired by the fact that a named entity often share the same boundary with its corresponding sub dependency tree, our DP module is responsible for producing DP boundary representation features for NER boundary detection.

NER Module Our NER module is the core component of the proposed model, adopting the information of both tasks for learning. Enlightened by the idea of (Yu, Bohnet, and Poesio 2020), we reuse the biaffine classifier to find named-entity spans in sentences. Similar to the DP module, two additional smaller MLP layers are applied to extract useful features for NER-start representation and NER-end representation. Given the i-th input word, s_i^{ner} and e_i^{ner} are used to denote its representation vector for NER-start and NER-end. But unlike DP, when processing the span classification, we do not only consider their NER representations but also their DP representations for learning. It is because that a word, which is the head or dependent of one dependency relation, usually locates at the start or end of one named-entity span in a sentence, as Figure 1 shows. Therefore, in our model, both the DP-start and DP-end vectors are concatenated to either the NER-start vector or the NER-end vector, as to provide supplementary information for NER. Accordingly, the concatenated NER-start and NER-end vectors of the i-th word are denoted as $\hat{s}_i^{ner}$ and $\hat{e}_i^{ner}$ separately. And, the span from the i-th word to the j-th one is represented as $(\hat{s}_i^{ner}, \hat{e}_j^{ner})$.

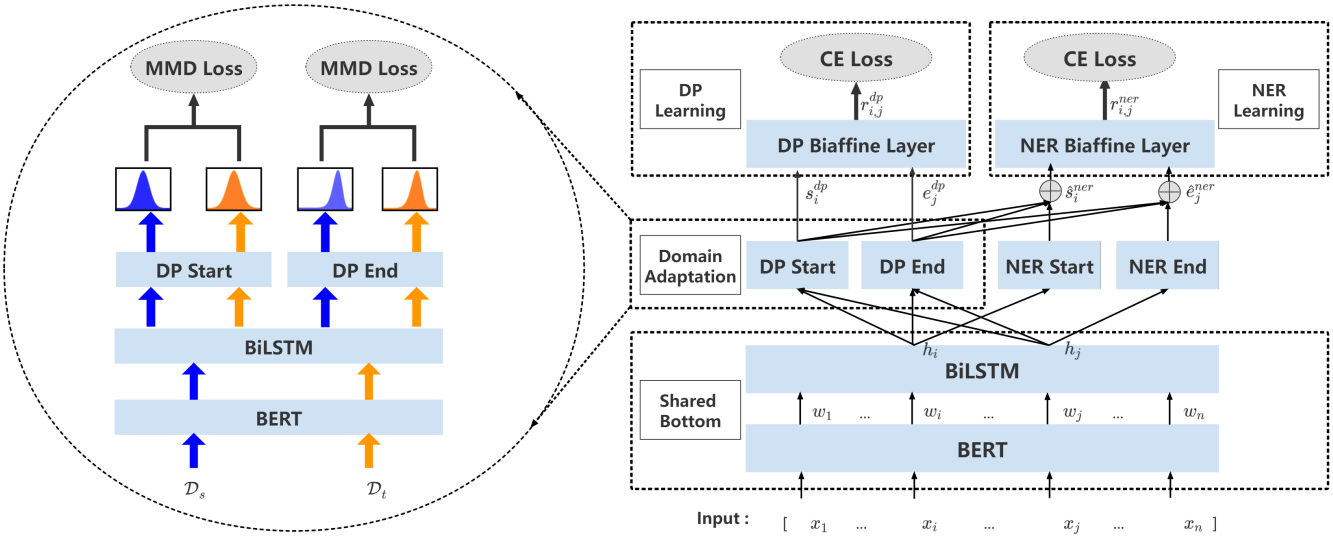


Figure 2: The left part introduces the process of DP Domain Adaptation. The right part describes the overall structure of our proposed network and gives the workflow about how a span starting at word x_i and ending at x_j is scored.

Finally, for all candidate spans, another biaffine classifier is applied to score their named-entity types in one $l \times l \times n$ tensor r^{dp}, where n is the number of named-entity types including *non-entity*. Formally, we have:

$$\begin{aligned} s_i^{ner} &= \text{MLP}^{\text{ner-start}}(h_i) \\ e_j^{ner} &= \text{MLP}^{\text{ner-end}}(h_j) \\ \hat{s}_i^{ner} &= s_i^{ner} \oplus s_i^{dp} \oplus e_i^{dp} \\ \hat{e}_j^{ner} &= e_j^{ner} \oplus e_j^{dp} \oplus s_j^{dp} \\ r_{i,j}^{ner} &= \hat{s}_i^{ner\top} \mathbf{U}^{ner} \hat{e}_j^{ner} \\ &\quad + \mathbf{W}^{ner}(\hat{s}_i^{ner} \oplus \hat{e}_j^{ner}) + b^{ner}, \end{aligned} \tag{6}$$

where s_i^{ner} and e_j^{ner} have the same dimension d, $\mathbf{U}^{ner}$ is a $3d \times n \times 3d$ tensor, $\mathbf{W}^{ner}$ is with a $6d \times n$ shape and b^{ner} is the bias.

Training

Auxiliary Objective Our method takes DP as the auxiliary task for joint learning with NER. To solve the lack of dependency annotations, we perform DP learning on labelled source domain and adapt the learned model to unlabelled target domain. Correspondingly, the auxiliary loss can be decomposed into two parts, DP learning loss and DP domain-adaptation loss.

Given a source dataset $\mathcal{D}_s$ with dependency labels, we use Cross Entropy (CE) on dependency relation classification as our DP learning loss:

$$\mathcal{L}^{dp} = -\sum_{x \in \mathcal{D}_s} \sum_{\substack{1 \le i \le l \\ 1 \le j \le l}} \log\left(\frac{exp(r_{i,j}^{dp}(z))}{\sum_{1 \le k \le m} exp(r_{i,j}^{dp}(k))}\right), \tag{7}$$

where z is the label index of the true dependency relation from x_i to x_j, l is the length of x and m is the number of dependency classes.

To make the trained DP model effective as well for a target dataset $\mathcal{D}_t$, we adapt the model by minimizing the MMD between the source and target domains in the DP representation spaces. Formally, the squared MMDs, computed in both DP-head and DP-end representation spaces, are used as DP domain-adaptation loss:

$$\mathcal{L}^{da} = \text{MMD}^2(\mathcal{S}_{\mathcal{D}_s}, \mathcal{S}_{\mathcal{D}_t}) + \text{MMD}^2(\mathcal{E}_{\mathcal{D}_s}, \mathcal{E}_{\mathcal{D}_t}), \tag{8}$$

where $\mathcal{S}_{\mathcal{D}_s}$ and $\mathcal{S}_{\mathcal{D}_t}$ are the sets of DP-start vectors on the source and target domains respectively, $\mathcal{E}_{\mathcal{D}_s}$ and $\mathcal{E}_{\mathcal{D}_t}$ are the sets of DP-end vectors on the two domains separately.

Main Objective For our main task, NER learning loss is taken as the objective. Given a target dataset $\mathcal{D}_t$ with entity labels, we use the Cross Entropy on named-entity span classification as our NER learning loss:

$$\mathcal{L}^{ner} = -\sum_{x \in \mathcal{D}_t} \sum_{\substack{1 \le i \le l \\ i \le j \le l}} \log\left(\frac{exp(r_{i,j}^{ner}(y))}{\sum_{1 \le k \le n} exp(r_{i,j}^{ner}(k))}\right), \tag{9}$$

where y is the label index of the true span type from x_i to x_j, l is the length of x and n is the number of named-entity classes.

In most of cross-domain NER scenarios, the size of target NER dataset is often small. Pre-training NER on source NER data can help learning informative cross-domain knowledge to improve the model.

Training Process Using the framework of Multi-Task Learning, the overall objective of our method can be generalized to minimize the joint loss as the following:

$$\mathcal{L} = \lambda(\mathcal{L}^{dp} + \mathcal{L}^{da}) + \beta \mathcal{L}^{ner}, \tag{10}$$

where λ and β are the parameters to balance the weights of the auxiliary and main tasks.

Dataset	Type	Train	Test	Domain
OntoNotes	#sentence	59924	8262	General
	#entity	81828	11057	
CoNLL03	#sentence	14987	3684	News
	#entity	23499	5648	
WNUT17	#sentence	3394	1286	Social Media
	#entity	3160	1589	
NCBI	#sentence	5424	940	Biomedical
	#entity	11249	1877	
MitRest	#sentence	7660	1521	Restaurant
	#entity	15363	3151	

Table 1: Statistic of datasets

To efficiently train our model, two different settings are used in the process of joint learning. At the beginning of learning, to make the model quickly derive the ability of dependency parsing, a large λ and small β are set for the auxiliary objective. After the auxiliary loss becomes small, to improve the model's performance on named-entity recognition, a small λ and large β are set for the main objective. In this way, the cross-domain and cross-task knowledge can be both effectively learned. The details will be described in the experiment section.

Experiments

Setup

Datasets. To evaluate the effectiveness of the proposed method, we conduct experiments on four English NER datasets, including CoNLL03 [1], WNUT17 [2], MitRest [3] and NCBI [4]. The four datasets come from four different domains, which are listed in Table 1. In addition, as DP task is taken as the auxiliary task in the proposed method, we adopt OntoNotes 5.0 [5] as our DP source dataset, converted to the Stanford dependency-tree format by using Stanford CoreNLP (Manning et al. 2014). Detailed statistics of the datasets are listed in Table 1.

Hyperparameters. We set the threshold of the maximum epoch as 100 for every model training. To our proposed model, the adopted BiLSTM module is incorporated with two 768-dimension LSTM layers. Each representation layer after BiLSTM is introduced with 128 dimensions. For the two biaffine classifiers, the parameters are configured as described in the previous section.With all the datasets, we use the batch size as 16 and the input maximum length as 256. In the training process, AdamW is taken as our optimizer with the learning rate 2e-5.

Baselines. The proposed NER method, which is based on Domain-Adapted Dependency Parsing (DADP), is compared with five other methods in our experiments. In all experimental settings, Large-BERT is taken as the pretrained model.

- **BiLSTM-CRF:** we take (Huang, Xu, and Yu 2015) as the naive baseline, which applies the classical model of BiLSTM and CRF for only NER task.
- **Dep-Flat:** (Jie and Lu 2019) is a dependency-guided NER method, which uses dependency-tree input features to improve BiLSTM-CRF.
- **Dep-GCN:** (Xu et al. 2021) is an improved version of Dep-Flat, introducing a novel LSTM structure to utilize the dependency features extracted by graph convolutional networks.
- **DA-LM:** (Jia, Xiao, and Zhang 2019) follows MTL framework for NER, employing cross-domain LM as the auxiliary task to improve cross-domain NER.
- **DA-ET:** (Jia and Zhang 2020) also follows MTL framework for NER, utilizing a multi-cell compositional LSTM structure to detect Entity Type across domains to improve cross-domain NER.

As mentioned previously, dependency annotations are the pre-requisite for the listed dependency-guided methods, but except OntoNotes 5.0, the other datasets do not provide dependency labels. To make the methods able to run on CoNLL03, WNUT17, MitRest and NCBI, we employ a third-party tool, spaCy (Honnibal and Montani 2017), to predict the labels for usage. For the listed MTL-based methods targeting cross-domain NER, we use OntoNotes 5.0 as their NER source-domain dataset. The standard *Recall*, *Precision* and *F1* are used as the evaluation measures in the following experiments.

Results and Analysis

Table 2 shows the overall results of comparisons. From the results, it can be observed that the proposed DADP outperforms other NER methods over three datasets.

To show the effectiveness of DADP, we first compare it with the MTL-based methods. Based on the results, it can be seen that DA-LM and DA-ET have similar performances over the three datasets. We think the main reason is that both LM and Entity Type Detection tasks focus on learning common semantic knowledge of words across domains, which provides the two models similar representative features for NER. Therefore, the difference between their performance is small. Compared with them, DADP achieves obvious lifts of F1 over the three datasets. We guess the reason is that not only word-level semantic knowledge is captured by our NER module but also syntax-level grammar knowledge is captured by our DP module. In this way, with the help of richer representative features, DADP can have better performances than DA-LM and DA-ET .

Next, our DADP is compared with the dependency-guided methods, Dep-Flat and Dep-GCN. The two dependency-guided methods achieve similar high-performance scores on MitRest, which is as expected. Because most of restaurant-domain sentences are written in common styles, their dependency relations can be easily and accurately predicted by the parsing tool spaCy. With

[1]https://www.clips.uantwerpen.be/conll2003/ner/

[2]http://noisy-text.github.io/2017/

[3]https://groups.csail.mit.edu/sls/downloads/restaurant/

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

[5]https://catalog.ldc.upenn.edu/LDC2013T19

Category	Method	MitRest			WNUT17			NCBI		
		P	R	F1	P	R	F1	P	R	F1
Cross-domain MTL Model	DA-ET	77.01	75.22	76.10	51.01	36.32	42.45	**85.62**	86.78	86.19
	DA-LM	76.07	74.55	75.30	50.48	34.95	41.30	84.27	85.69	84.97
Dependency-guided Model	Dep-Flat	76.26	78.19	77.21	27.46	**65.65**	38.72	84.49	83.96	84.22
	Dep-GCN	77.02	**79.27**	78.12	29.51	63.86	40.37	83.17	85.94	84.53
	BiLSTM-CRF	76.40	77.14	76.76	49.64	32.37	39.18	84.91	86.39	85.64
	DADP	**78.64**	79.20	**78.92**	**60.58**	36.99	**45.93**	83.56	**89.48**	**86.42**

Table 2: Performance Comparison by Precision, Recall and F1 on Three Datasets.

# Symbol	Dep-GCN			DA-ET			DADP		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
1	90.21	89.79	90.00	89.87	89.50	89.68	91.99	92.08	92.03
2	87.27	85.58	86.42	89.63	87.24	88.42	91.37	90.82	91.10
3	84.32	81.39	82.83	88.32	85.80	87.04	90.10	91.35	90.72
≥ 4	79.82	78.15	78.98	87.67	84.02	85.81	89.31	90.32	89.81

Table 3: Robustness Comparison between Dep-GCN, DA-ET and DADP on CoNLL03

the predicted dependency annotations of high quality, it is unsurprising that the methods perform well on MitRest. However, on WNUT17 and NCBI, Dep-Flat and Dep-GCN lose their effects. As WNUT17 sourced from social media data that contains many uncommon grammars, and NCBI sourced from biomedical domains that includes obscure sentences, such written styles make both of them hard to be correctly parsed. With the low-quality predicted labels, the poor performances of Dep-Flat and Dep-GCN are inevitable. Compared with them, DADP takes no dependency labels as input features to predict NER. Our proposed method mainly exploits cross-domain DP, to learn representative features of grammar knowledge across domains for NER, rather than directly using dependency labels as data features. In this way, without the affection of noise labels, DADP outperforms Dep-Flat and Dep-GCN significantly on WNUT17 and NCBI.

Performance on Robustness Test

To further study the robustness of DADP, we compare it with Dep-GCN and DA-ET on CoNLL03, which is a rather clean dataset of NER. In the experiment, we randomly insert specific symbols into the sentences of CoNLL03, to mess up the corpus. Tests on sentences with the different numbers of inserted symbols are given in Table 3. From the table, we can find that with noise symbols increasing, the performance of Dep-GCN decreases obviously. The reason may be that the inserted symbols break the coherence of sentences, leading to poor-quality parsed labels. As the dependency information is the key to the success of Dep-GCN, its robustness can be easily challenged. DA-ET takes no dependency information as input but it still suffers from the noise. The reason may be that the named entities are split by inserted symbols, making the auxiliary Entity Type Unit hard to tell entity words from non-entity words. In this way, the performance of DA-ET is also affected by the number of noise symbols. For our DADP, its performance is always the highest in different settings, because DADP uses the dependency annotation for representation learning to NER instead of direct input features. With rich semantic representations in the word and syntax level, DADP has good tolerance to noise symbols as expected, leading to its strong robustness. This experimental result also reflects the flexibility of DADP on exploiting DP knowledge.

Setting	WNUT17		
	P(%)	R(%)	F1(%)
NER	64.19	30.02	40.90
NER + DP	61.41	35.27	44.80
NER + DP + DA	60.58	36.99	45.93

Setting	NCBI		
	P(%)	R(%)	F1(%)
NER	80.22	87.91	83.89
NER + DP	83.01	88.74	85.78
NER + DP + DA	83.56	89.48	86.42

Table 4: Ablation Study of DADP on WNUT17 and NCBI

Performance on Few-Shot NER

To further investigate the performance of DADP on few-shot NER using cross-domain knowledge, we compare it with two other cross-domain MTL-based methods DA-LM and DA-ET on the three datasets. In the experiment, we randomly select 50%, 25% and 10% samples from the original datasets as target-domain training datasets, to add dif-

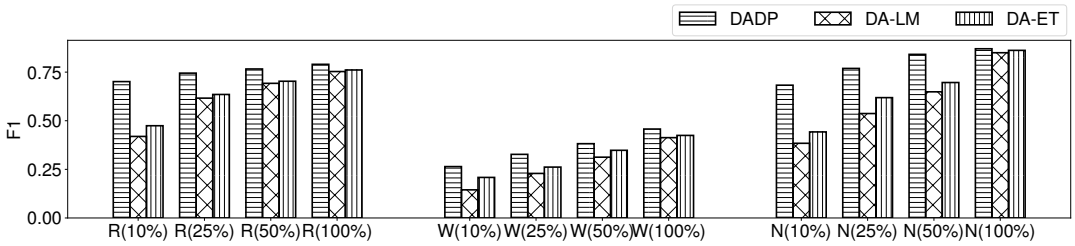
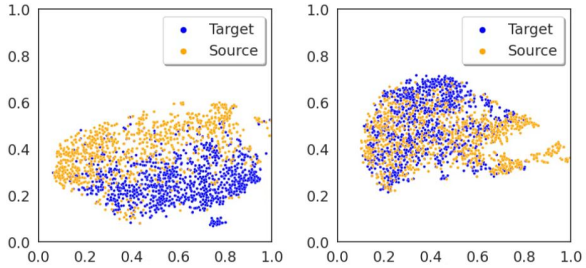


Figure 3: F1 varying training size (R: MitRest; W: WNUT17; N: NCBI)

ficulty of model training on cross-domain NER. The F1 scores achieved by the mentioned methods with different sizes of target-domain datasets are given in Figure 3. From the figure, we can find that with the size of target-domain dataset decreasing, the performance of DA-LM and DA-ET decreases rapidly. The reason is that as our setting takes OntoNotes 5.0 as the source-domain dataset, there exist semantic gaps between the source domain and the target domains. In other words, no sufficient word-level common knowledge exist between the source and target domains. Limited by the gaps, DA-LM and DA-ET, which focus on transferring word-level semantic knowledge across domains, are harder to extract useful common knowledge from the fewer data samples. In comparison, with the size of target-domain dataset decreasing, the performance of DADP decreases considerably slowly. The reason is that dependency structural knowledge is more stable and general across different domains, which can be learned even from small datasets. Therefore, it is not strange that DADP, which focuses on transferring syntax-level semantic knowledge across domains, can significantly outperform other two MTL-based NER methods. This overall result shows the effectiveness and potential of DADP on improving cross-domain NER.

Ablation Study

To evaluate the single effectiveness of each component in our model, we conduct a series of ablation experiments on WNUT17 and NCBI as shown in Table 4. For the two datasets, we first use the only NER module to train the model and achieve F1 scores with 40.90% and 83.89%, as the baseline scores. To further improve DADP, the training strategy is used as described in the training section. At the beginning of learning, to make DP get fully trained, we process 30 rounds of DP learning with $\lambda = 1$ and $\beta = 0$. After the pretraining process, to investigate the influence of only DP learning, we perform an experiment without the adaptation loss and vary the value of λ from 0 to 1 with $\beta = 1$. We find the setting $\lambda = 0.05$ increases the baseline F1 of WNUT17 with 3.9% and that of NCBI with 1.9%, both of which give the best results among different settings. The significant improvement shows the effectiveness of the Multi-Task Learning framework using DP as the auxiliary task. However, this setting may not fully explore the effect of DP because of the gap between distributions of the source and target data. To assess the role of Domain Adaptation, we run another experiment with both DP loss and DA loss. With the help of DA,



(a) Before Adaptation (b) After Adaptation

Figure 4: Domain Adaptation for DP Representations

both of the F1s have improved at least 0.6%. The outcome demonstrates that DA has a positive effect on improving our NER model. Even though the improvement is not significant as the previous one, we find the convergence rate of DADP is accelerated with DA incorporated. The reason may be that useless semantic features to the target domain are discarded in the process of Domain Adaptation, making the weight learning process more efficient. To show the effect of Domain Adaptation, in Figure 3, we use t-SNE algorithm to visualize the DP vectors before and after the domain adaptation on WNUT17. We can observe that the source and target distributions are obviously matched after the adaptation.

Conclusion

In this paper, we propose a novel dependency-guided NER framework, which leverages dependency relations between words to improve named entity identification. Compared with other dependency-guided methods which have difficulties to apply on dependency-unlabelled NER corpora, our method demonstrates its advantage on the usability and practicality of transferring dependency-tree information across domains. With the exploration of MMD, we adapt DP knowledge across domains in an unsupervised manner. Through unifying NER and DP within the same network structure, our model can efficiently learn and utilize shared cross-domain and cross-task knowledge.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (Grant No. 2021ZD0200300).

References

Bista, U.; Mathews, A.; Menon, A.; and Xie, L. 2020. SupMMD: A Sentence Importance Model for Extractive Summarisation using Maximum Mean Discrepancy. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 4108–4122.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28: 41–75.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1*, 4171–4186. Association for Computational Linguistics.

Dozat, T.; and Manning, C. D. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Graves, A.; and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, 2047–2052 vol. 4.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. J. 2012. A kernel two-sample test. *Journal of Machine Learning Research*, 13: 723–773.

Guo, Z.; Zhang, Y.; and Lu, W. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 241–251.

Hearst, M. A.; Dumais, S. T.; Osuna, E.; Platt, J.; and Scholkopf, B. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4): 18–28.

Honnibal, M.; and Montani, I. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Jia, C.; Xiao, L.; and Zhang, Y. 2019. Cross-Domain NER using Cross-Domain Language Modeling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2464–2474. Association for Computational Linguistics.

Jia, C.; and Zhang, Y. 2020. Multi-Cell Compositional LSTM for NER Domain Adaptation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 5906–5917. Association for Computational Linguistics.

Jie, Z.; and Lu, W. 2019. Dependency-Guided LSTM-CRF for Named Entity Recognition. 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 2019, Hong Kong, China, November 3-7, 2019*, 3860–3870. Association for Computational Linguistics.

Jie, Z.; Muis, A. O.; and Lu, W. 2017. Efficient Dependency-Guided Named Entity Recognition. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 3457–3465. AAAI Press.

Li, F.; Lin, Z.; Zhang, M.; and Ji, D. 2021. A Span-Based Model for Joint Overlapped and Discontinuous Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4814–4828.

Li, J.; Sun, A.; Han, J.; and Li, C. 2022. A Survey on Deep Learning for Named Entity Recognition. *IEEE Trans. Knowl. Data Eng.*, 34(1): 50–70.

Liu, L.; Shang, J.; Ren, X.; Xu, F. F.; Gui, H.; Peng, J.; and Han, J. 2018. Empower Sequence Labeling with Task-Aware Neural Language Model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, Feb 2-7, 2018*, 5253–5260. AAAI Press.

Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; and McClosky, D. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60.

Qian, T.; Zhang, M.; Lou, Y.; and Hua, D. 2021. A Joint Model for Named Entity Recognition With Sentence-Level Entity Type Attentions. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 1438–1448.

Wang, M.; and Deng, W. 2018. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153.

Xiao, S.; Ouyang, Y.; Rong, W.; Yang, J.; and Xiong, Z. 2019. Similarity Based Auxiliary Classifier for Named Entity Recognition. 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)*, 1140–1149.

Xu, L.; Jie, Z.; Lu, W.; and Bing, L. 2021. Better Feature Integration for Named Entity Recognition. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, 3457–3469.

Yu, J.; Bohnet, B.; and Poesio, M. 2020. Named Entity Recognition as Dependency Parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6470–6476. Online: Association for Computational Linguistics.

Zhang, T.; Xia, C.; Yu, P. S.; Liu, Z.; and Zhao, S. 2021. PDALN: Progressive Domain Adaptation over a Pre-trained Model for Low-Resource Cross-Domain Named Entity Recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 5441–5451. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.