Three Heads Are Better Than One: Improving Cross-Domain NER with Progressive Decomposed Network

Xuming Hu[1,2], Zhaochen Hong[2], Yong Jiang[3]*, Zhichao Lin[3], Xiaobin Wang[3], Pengjun Xie[3], Philip S. Yu[4]

[1]AI Thrust, Hong Kong University of Science and Technology (Guangzhou),
[2]School of Software, Tsinghua University,
[3]DAMO Academy, Alibaba Group,
[4] Department of Computer Science, University of Illinois Chicago
xuminghu@hotmail.com, jiangyong.ml@gmail.com

Abstract

Cross-domain named entity recognition (NER) tasks encourage NER models to transfer knowledge from data-rich source domains to sparsely labeled target domains. Previous works adopt the paradigms of pre-training on the source domain followed by fine-tuning on the target domain. However, these works ignore that general labeled NER source domain data can be easily retrieved in the real world, and soliciting more source domains could bring more benefits. Unfortunately, previous paradigms cannot efficiently transfer knowledge from multiple source domains. In this work, to transfer multiple source domains' knowledge, we decouple the NER task into the pipeline tasks of mention detection and entity typing, where the mention detection unifies the training object across domains, thus providing the entity typing with higher-quality entity mentions. Additionally, we request multiple general source domain models to suggest the potential named entities for sentences in the target domain explicitly, and transfer their knowledge to the target domain models through the knowledge progressive networks implicitly. Furthermore, we propose two methods to analyze in which source domain knowledge transfer occurs, thus helping us judge which source domain brings the greatest benefit. In our experiment, we develop a Chinese cross-domain NER dataset. Our model improved the F1 score by an average of 12.50% across 8 Chinese and English datasets compared to models without source domain data.

Introduction

Named Entity Recognition (NER) aims to infer a label for each token in the sentence to determine whether it is a part of an entity and classify entities into predefined types. Recent NER models show decent performance when sufficient data are available (Nasar, Jaffry, and Malik 2021; Liu et al. 2022). However, in practice, we always focus on specific domains, such as artificial intelligence, music, and culture, where labeled data are often difficult to obtain. A natural thought is whether we can transfer knowledge from data-rich domains to sparsely labeled domains to improve performance, which motivates our research on the cross-domain NER.

Cross-domain NER tasks require the model to have sufficient knowledge transfer ability and to quickly adapt to the target domain with a few training samples. Existing studies consider the CoNLL 2003 dataset (Sang and De Meulder 2003) from Reuters News as the source domain, and annotate Wikipedia datasets about Politics, Natural Science, Music, Literature, and Artificial Intelligence as target domains (Liu et al. 2021). Following this setting, previous works (Yang, Salakhutdinov, and Cohen 2017; Jia, Liang, and Zhang 2019; Jia and Zhang 2020; Zheng, Chen, and Ma 2022) adopt the paradigm of pre-training in the source domain and then fine-tuning in the target domain. Zhang et al. (2022b); Xu and Cai (2023); Xu et al. (2023) jointly trains data from the source and target domains. Although these methods have achieved good performance, the training paradigms lose the scalability to expand to the real-world scenarios: In the real world, the general labeled NER source domain data can be easily retrieved (Li et al. 2020), and more source domains lead to better results, which have been shown in other NLP tasks (Imani et al. 2022; Fujinuma, Boyd-Graber, and Kann 2022). Therefore, how to transfer knowledge from the ***multi-source*** instead of the ***single-source*** domains to the target domain is a more crucial problem. Simply pre-train models sequentially on each source domain often leads to a catastrophic problem of forgetting source domain knowledge (French 1999). Furthermore, the jointly training paradigm cannot handle multiple source domains with different entity types (Hu et al. 2021c), and at the same time, ignores the diversity of each source domain and its different contribution to the target domain (Zhang et al. 2022b).

To leverage *multi-source* domain knowledge to benefit the training of the target domain, we adopt the Progressive Decomposed Networks to unify the training paradigms and transfer knowledge across domains. As illustrated in Figure 1, the Decomposed Network decouples the NER task into the pipeline tasks of mention detection and entity typing. The mention detection task aims to infer a `B`, `I`, or `O` label for each token to determine whether it is part of an entity, thus bridging the gap between different domains through a unified training paradigm. The entity typing task shares the same semantic encoder (Devlin et al. 2019) as the mention detection task, and classifies the obtained higher-quality entity mentions into predefined entity types. To utilize knowledge from multiple different sources domain, we adopt two progressive methods: (1) ***Explicitly.*** We solicit each general source domain model to

*Corresponding author.
Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

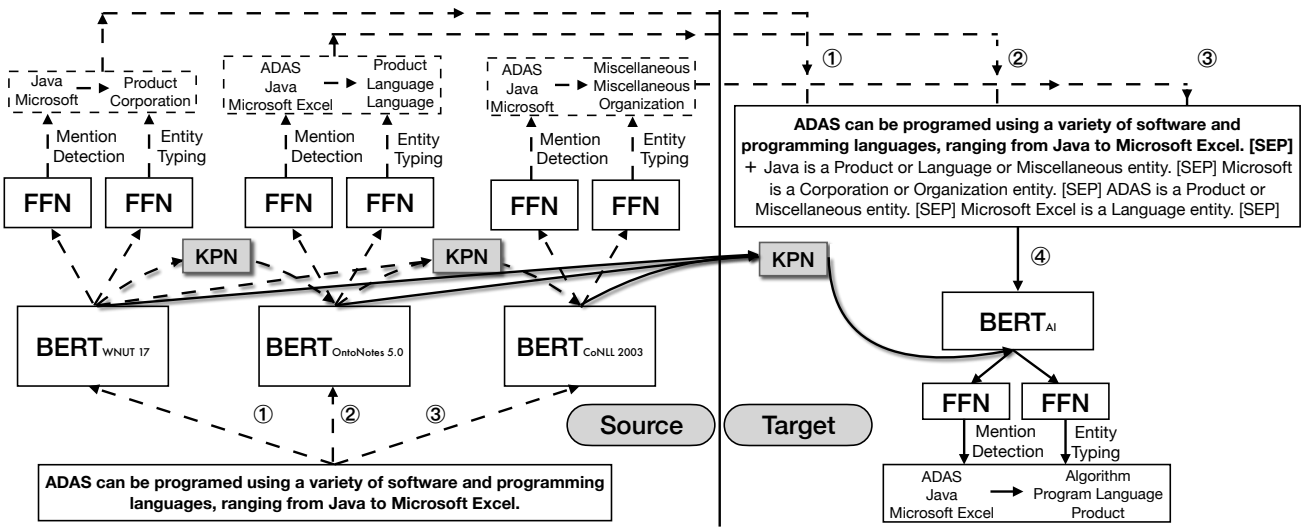


Figure 1: Overview of the proposed Progressive Decomposed Network. The network consists of two tasks, mention detection and entity typing, which obtain entity mentions and classify them into predefined types. The Progressive procedure consists of two methods. First, the potential named entities obtained by the source domain models are copied after the target domain sentence. Second, the embeddings predicted by the source domain models ①, ②, and ③ are transferred to the target domain model ④ through the Knowledge Progressive Networks.

predict the potential named entities for sentences in the target domain. For example, as shown in Figure 1, we treat the publicly available WNUT 17, OntoNotes 5.0, and CoNLL 2003 as source domain datasets, and BERT trained on OntoNotes 5.0 can infer the types of entities ADAS, Java, and Microsoft Excel as `Product`, `Language`, and `Language`. We copy these potential named entities at the end of the sentence. Although the target domain has different entity detection and typing goals, it still provides more guidance for the limited target domain data. (2) ***Implicitly.*** We employ the Knowledge Progressive Network (KPN) to transfer knowledge from each source domain. When we train the target domain model in step ④, the trained ①, ②, and ③ models will give the embeddings of the sentence in the target domain, and the knowledge will be transferred to the target domain model through the KPNs which could be optimized.

Furthermore, we analyze in which knowledge transfer of the source domain occurs and propose two approaches: a quick analytical approach based on Fisher information (Amari 1998) and an intuitive approach based on perturbation analysis. Following previous work (Liu et al. 2021), we explore a Chinese cross-domain NER dataset and evaluate our model on cross-domain NER datasets in two languages (English and Chinese).

Our contributions are as follows: **(1)** To face the challenges in the multi-source cross-domain NER tasks, we propose the Progressive Decomposed Networks to unify the training object across domains and transfer knowledge from each source domain to the target domain explicitly and implicitly. **(2)** We propose two methods, based on a perturbation analysis and derived from the Fisher Information to analyze in which source domain knowledge transfer orrurs. **(3)**We explore a Chinese cross-domain NER dataset and show that our model outperforms the strong baselines on eight English and Chinese cross-domain NER datasets.

Problem Definition

NER could be naturally viewed as a sequence labeling problem (Lample et al. 2016; Luo, Xiao, and Zhao 2020). Specifically, given an input sequence $\mathcal{X} = \{x_1, x_2, \cdots, x_N\}$ with N tokens, the output is the corresponding label sequence $\mathcal{Y} = \{y_1, y_2, \cdots, y_N\}$, which marks where each entity starts and ends and the entity type. In the cross-domain NER task, we are given datasets from the source domain $\mathcal{D}_{src} = \{(\mathcal{X}_m^s, \mathcal{Y}_m^s)\}_{m=1}^{N_s}$ and the target domain $\mathcal{D}_{tgt} = \{(\mathcal{X}_i^t, \mathcal{Y}_i^t)\}_{i=1}^{N_t}$. Our objective is to learn a classifier $\mathcal{F} = f(\mathcal{X}; \theta)$ could transfer knowledge between domains to bridge the discrepancies. Note that the labels of the source domain ($\mathcal{Y}^s$) and the target domain ($\mathcal{Y}^t$) do not match, and the number of samples in the target domain is much smaller than the source domain ($N_t \ll N_s$).

Proposed Method

We propose the Progressive Decomposed Network to transfer knowledge from multi-source domains to the target domain and analyze the degree of transferable provided by each source domain.

The Decomposed Network

Due to a mismatch in the number of entity types between multi-source and target domains, the same classifier $\mathcal{F}$ cannot be trained across domains. A very intuitive idea is to share as many $\mathcal{F}$ sub-modules as possible. In practice, as shown in Figure 1, the Decomposed network decouples the NER task into two pipeline tasks of mention detection and entity typing. For the mention detection task, we adopt the `B`, `I`, or `O` label without specific entity type for each token and

intend to discover potential mentions in the input sequence as entities. We use the pre-trained BERT (Devlin et al. 2019) denoted as $f_{\text{BERT}}(\cdot)$ to encode the input sequence $\mathcal{X}$:

$$[\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_N] = f_{\text{BERT}}(x_1, x_2, \cdots, x_N) \tag{1}$$

Then we adopt a feed-forward network $f_{\text{MD}}(\cdot)$ for `B`, `I`, or `O` tagging of the mention detection:

$$\hat{\boldsymbol{y}}_i^{\text{MD}} = f_{\text{MD}}(\mathbf{h}_i), \tag{2}$$

where $\hat{\boldsymbol{y}}_i^{\text{MD}}$ is the predicted tag vector of the token x_i in the input sequence. The mention detection loss of an sequence $\mathcal{X}$ is:

$$\mathcal{L}^{\text{MD}} = -\sum_{i=1}^{L^{\text{MD}}} \boldsymbol{y}_i^{\text{MD}} \log \hat{\boldsymbol{y}}_i^{\text{MD}}, \tag{3}$$

where L^{MD} is the length of the sequence, $\boldsymbol{y}_i^{\text{MD}}$ is the ground-truth tag vector of the token x_i.

Due to the unified three-type training object, the mention detection task is easy to share networks f_{BERT} and f_{MD} across domains. However, the target domain and the source domain have different numbers of entity types, so we adopt different feed-forward networks f_{ET} for entity typing:

$$\boldsymbol{e}_i = \text{Concat}(\mathbf{h}_l, \mathbf{h}_m), \tag{4}$$

$$\hat{\boldsymbol{y}}_i^{\text{ET}} = f_{\text{ET}}(\boldsymbol{e}_i), \tag{5}$$

where $\boldsymbol{e}_i \in \mathbb{R}^{2 \cdot d}$ is the representation of the i-th entity mention which begins at l-th token and ends at m-th token in the input sequence $\mathcal{X}$, $\hat{\boldsymbol{y}}_i^{\text{ET}}$ is the predicted type vector of the i-th entity mention. The entity typing loss of a sequence $\mathcal{X}$ is:

$$\mathcal{L}^{\text{ET}} = -\sum_{i=1}^{L^{\text{ET}}} \boldsymbol{y}_i^{\text{ET}} \log \hat{\boldsymbol{y}}_i^{\text{ET}} \tag{6}$$

where L^{ET} is the number of entity in the sequence, $\boldsymbol{y}_i^{\text{ET}}$ is the ground-truth type vector of the i-th entity mention in the input sequence. Overall, the total loss of two pipeline tasks on training set is:

$$\mathcal{L} = -\frac{1}{Z} \sum_{i=1}^{Z} (\lambda \mathcal{L}_i^{\text{MD}} + (1 - \lambda) \mathcal{L}_i^{\text{ET}}) \tag{7}$$

where Z is the size of the training set, λ balances between the mention detection loss and entity typing loss and we set $\lambda = 0.4$. In practice, the classifier $\mathcal{F}$ consists of f_{BERT}, f_{MD}, and f_{ET}. Despite the mismatch in the number of entity types between multi-source and target domains, we share as many trained networks f_{BERT} and f_{MD} as possible across domains by decomposing the NER task.

The Progressive Procedure

In addition to unifying the training paradigm as much as possible, we employ two methods to handle the process of knowledge transfer from multi-source domains explicitly and implicitly.

First, we explicitly utilize the Decomposed Network fine-tuned in the source domain to predict potential entities and types for sentences in the target domain. For example, as illustrated in Figure 1, for the sentence in the target domain: "*ADAS can be programed using a variety of software and programming languages, ranging from Java to Microsoft Excel*", we leverage the $\mathcal{F}$ fine-tuned on WNUT 17 and obtain the potential entities: (*Java*, `Product`) and (*Microsoft*, `Corporation`). Then we explicitly copy these information at the end of the corresponding sentence to indicate potential entities, such as: "*Java is a Product entity. [SEP] Microsoft is a Corporation entity.*". Note that all source domains will copy potential entities for the target domain. Although the tagging methods of entity mentions in the target domain are different from those in the source domain, we still actively believe that by showing entity examples and types, the model can have a better sense about the NER task in the target domain. These auxiliary entity information is only used as input to f_{BERT}, but will not be used by f_{MD} and f_{ET} for inference.

Second, we implicitly progressive knowledge across domains with the Knowledge Progressive Networks. As shown in Figure 1, we assume that we are now training the k-th domain (could be any source domain or the target domain). For the input token x_i, the knowledge transferred from the previous j-th domain can be represented as:

$$\mathbf{h}_i^{(j:k)} = \text{LN}\left(\text{Dropout}\left(\boldsymbol{W}^{(j:k)} \mathbf{h}_i^{(j)}\right)\right), \tag{8}$$

where $\mathbf{h}_i^{(j:k)} \in \mathbb{R}^d$ is the hidden embedding of x_i transferred from the j -th domain to the k-th domain, $\boldsymbol{W}^{(j:k)} \in \mathbb{R}^{d \times d}$ is the corresponding weight matrix, and LN is the Layer Normalization Layer.

For the target model, inspired by Asghar et al. (2020), we aggregate the knowledge and feed it into mention detection and entity typing tasks:

$$\tilde{\mathbf{h}}_i^{(k)} = \text{LN}\left(\mathbf{h}_i^{(k)} + \sum_{j<k} \alpha_j \mathbf{h}_i^{(j:k)}\right), \tag{9}$$

where α_j is a parameter to adjust the weight of the knowledge from j-th domain. After obtaining the updated $\tilde{\mathbf{h}}_i^{(k)}$, we treat it as a feature into Eq. 2 and 5 for the mention detection and entity typing tasks.

Knowledge Transfer Analysis

Unlike previous fine-tuning of the source domain models on the target domain, our method does not destroy the features learned by the source domain models. This allows us to study in which source domain knowledge transfer occurs. We propose two approaches: a quick analytical approach based on the Fisher Information (Amari 1998) and an intuitive approach based on the perturbation analysis.

Source Domain Perturbation Sensitivity

The source domain perturbation analysis aims to estimate which embeddings of the source domain contribute significantly to the performance of the target domain. To this end, we inject Gaussian noise into the hidden representation $\mathbf{h}^{(i)}$ output by each domain. A new sample is used in each forward pass, and the average effect of these perturbations over 10 epochs is calculated. We scale the noise variance to be proportional to the variance of each feature embedding, which is

invariant to arbitrary scaling factors in the network weights. In practice, we define $\Lambda^{(j)} = 1/\sigma^2(j)$ as the noise-injected precision of the embedding in the j-th source domain, which causes a λ% F1 drop in the target domain, where λ is a fixed hyperparameter. We set $\lambda = 30$ in our analysis. The source domain perturbation sensitivity (SDPS) can be calculated as:

$$\mathrm{SDPS}(j) = \frac{\Lambda^{(j)}}{\sum_j \Lambda^{(j)}}. \tag{10}$$

Source Domain Fisher Sensitivity

Although calculating the sensitivity of the perturbation is intuitive for analyzing the impact of the source domain, it requires a long calculation time and is affected by random factors, therefore, we introduce a method based on the Fisher Information (Amari 1998) for fast and theorical analysis.

We denote $\mathbf{h}^{(j)}$ as the hidden embedding of sentence $\mathcal{X}$ given by the j-th source domain,and $\mathcal{Y}$ is its entity labels, $\tilde{p}(\hat{\mathcal{Y}}|\mathcal{X})$ as the model-induced softmax probability distribution of the entity typing task. Note that our NER task is a pipeline task, so we only focus on the output distribution of the entity typing task. We can use the Fisher Information matrix $\mathbf{F}^{(j)}$ to get a local approximation to the perturbation sensitivity of $\mathbf{h}^{(j)}$:

$$\mathbf{F}^{(j)} = \mathbb{E}_{p(\mathcal{Y},\mathcal{X})}\left[\frac{\partial \log(\tilde{p})}{\partial \mathbf{h}^{(j)}} \frac{\partial \log(\tilde{p})}{\partial \mathbf{h}^{(j)}}^T\right], \tag{11}$$

where $p(\hat{\mathcal{Y}}, \mathcal{X})$ is the joint probability distribution of $\hat{\mathcal{Y}}$ and $\mathcal{X}$. Specifically, for the m-th dimension of the hidden embedding $\mathbf{h}^{(j)}$, its j-th source domain fisher sensitivity can be calculated as:

$$\mathrm{SDFS}(j, m) = \frac{\mathbf{F}^{(j)}(m, m)}{\sum_j \mathbf{F}^{(j)}(m, m)}. \tag{12}$$

In practice, we can further consider the SDFS score for each domain as a overview of its sensitivity and decide the influence of the j-th source domain to the final prediction:

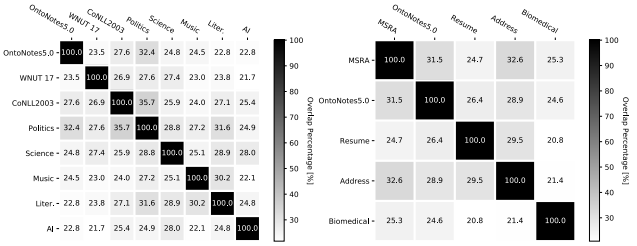
$$\mathrm{SDFS}(j) = \sum_m \mathrm{SDFS}(j, m). \tag{13}$$

Experimental Evaluation

We first explore a Chinese cross-domain NER dataset and show that our model outperforms the strong baselines on eight English and Chinese target domain datasets. Then we analyze each module in our model to verify why knowledge can be transferred across domains. We also adopt the perturbation analysis and the Fisher Information to study in which source domain knowledge transfer occurs.

Datasets

For the cross-domain NER task in English, we adopt CoNLL 2003 (Sang and De Meulder 2003) (Newswire domain), WNUT 17 (Derczynski et al. 2017) (Social Media domain), and OntoNotes 5.0 (English) (Pradhan et al. 2013) (General domain) as source domain datasets, while Politics, Natural Science, Music, Literature and Artificial Intelligence proposed by Liu et al. (2021) as target domain datasets. To better



(a) The vocabulary overlaps of the English cross-domain NER datas.

(b) The vocabulary overlaps of the Chinese cross-domain NER datas.

Figure 2: The vocabulary overlaps of the NER datasets.

demonstrate the effectiveness of our model, we explore a Chinese cross-domain NER dataset. The two source domains are OntoNotes 5.0 (Chinese) (Pradhan et al. 2013) (General domain), and MSRA dataset (Levow 2006) (Newswire domain). The three target domains are Resume (Zhang and Yang 2018), Address(Alibaba 2021), and Biomedical (Zhang et al. 2022a). The detailed statistics of datasets are shown in Appendix .

In the English cross-domain NER datasets, the samples in the source domain are far more than the target domain, but the entity types are less. Following previous datasets (Liu et al. 2021), we randomly select 0.2k training samples from the Chinese NER dataset in the target domain as training data. To show the diversity of the target domain datasets, we also calculate the domain overlaps by counting the vocabulary overlaps of the NER datasets (Liu et al. 2021). We consider the top 5k most common words while creating vocabularies for each domain (excluding stopwords). From Figure 2, we can observe that the vocabulary overlaps between domains are generally small, further demonstrating that the domains in our collected datasets are quite different.

Baselines and Evaluation Metric

We compare our method with competitive baselines that focus on cross-domain NER tasks: (1) **BiLSTM-CRF** (Lample et al. 2016). (2) **Coach** (Liu et al. 2020). (3) **LM-NER** (Jia, Liang, and Zhang 2019). (4) **MultiCell-LM** (Jia and Zhang 2020). (5) **BERT-JF** and **BERT-PF** (Liu et al. 2021). (6) **Style-NER** (Chen et al. 2021). (7) **LST-NER** (Zheng, Chen, and Ma 2022). (8) **LANER** (Hu et al. 2022). (9) **DoSEA** (Tang et al. 2022). (10) **MTD** (Zhang et al. 2022b). (11) **MTD-MoCL** (Xu et al. 2023). (12) **DH-GAT** (Xu and Cai 2023). We give a detailed introduction to the baselines in Appendix . We utilize the F1 score as our evaluation metric, focusing on precise entity mention and type matching. Furthermore, we report the F1 score for entity mention matching, grounded in the `B,I,O` labels.

Implementation Details

For fair comparison with baseline models, we adopt BERT-Base and BERT-Base-Chinese (Devlin et al. 2019) as our language models. The training sequence for the English source domain datasets is WNUT 17, OntoNotes 5.0, CoNLL 2003, and the Chinese datasets are OntoNotes 5.0, MSRA. For the Decomposed Network, we let the target domain model shares the parameters of the last source domain model. We tune

Methods	English Target Domain Datasets					Chinese Target Domain Datasets		
	Politics	Science	Music	Litera.	AI	Resume	Address	Biomedical
BiLSTM-CRF†	53.44 (73.45)	46.65 (67.18)	42.79 (61.95)	41.23 (60.22)	41.68 (60.67)	77.87 (85.32)	58.68 (81.38)	36.34 (62.47)
BiLSTM-CRF	56.60 (74.52)	49.97 (69.45)	44.79 (64.17)	43.03 (62.27)	43.56 (62.93)	81.45 (88.79)	61.23 (82.45)	39.42 (64.02)
Coach	61.50 (79.44)	52.09 (72.36)	51.66 (71.73)	48.35 (68.53)	45.15 (65.63)	85.37 (90.35)	64.30 (83.94)	42.35 (65.96)
BERT-CRF†	65.79 (85.24)	63.42 (84.55)	65.53 (85.21)	60.24 (79.43)	50.46 (71.62)	91.33 (96.54)	75.44 (91.69)	45.68 (68.49)
LM-NER	68.44 (87.25)	64.31 (85.41)	63.56 (84.93)	59.59 (77.31)	53.70 (74.58)	92.25 (96.90)	77.15 (92.40)	47.55 (70.05)
BERT-JF	68.85 (87.25)	65.03 (85.49)	67.59 (86.16)	62.57 (83.77)	58.57 (76.08)	92.46 (96.99)	77.31 (92.49)	47.27 (69.87)
BERT-PF	68.71 (87.82)	64.94 (85.35)	68.30 (86.81)	63.63 (85.13)	58.88 (76.76)	92.35 (96.95)	77.19 (92.45)	47.39 (70.12)
MultiCell-LM	70.56 (90.64)	66.42 (84.58)	70.52 (90.35)	66.96 (84.64)	58.28 (75.93)	93.14 (97.32)	77.97 (93.14)	48.14 (71.15)
Style-NER	68.78 (87.72)	63.95 (85.95)	65.43 (85.83)	60.94 (78.37)	58.73 (75.41)	92.38 (97.40)	77.45 (92.78)	48.25 (71.39)
LST-NER	70.44 (90.18)	66.83 (85.55)	72.08 (92.28)	67.12 (86.24)	60.32 (77.86)	92.97 (97.79)	77.79 (93.05)	48.83 (71.87)
LANER	71.65 (91.22)	69.29 (88.45)	73.07 (92.44)	67.98 (86.58)	61.72 (80.42)	92.66 (97.67)	77.63 (92.97)	48.47 (71.60)
DoSEA	75.52 (94.34)	71.60 (91.08)	73.10 (92.69)	68.59 (87.78)	66.03 (84.14)	93.57 (98.23)	78.02 (93.33)	49.11 (72.16)
MTD	76.70 (94.96)	72.35 (92.28)	76.10 (94.70)	69.22 (88.40)	68.93 (88.22)	93.80 (98.47)	78.25 (93.41)	49.62 (72.57)
MTD-MoCL‡	76.74 (94.82)	72.59 (92.27)	76.26 (94.82)	69.30 (88.55)	69.05 (88.32)	93.82 (98.52)	78.15 (93.23)	49.45 (72.42)
DH-GAT‡	76.88 (95.12)	72.86 (92.42)	77.10 (94.87)	70.03 (89.05)	69.30 (89.03)	93.95 (98.79)	78.69 (93.78)	49.88 (72.97)
Ours	**77.82 (95.45)**	**73.63 (93.01)**	**77.57 (95.18)**	**70.39 (90.58)**	**70.05 (90.35)**	**95.31 (99.14)**	**79.29 (93.96)**	**52.07 (74.30)**
w/o PE	77.25 (95.24)	73.06 (92.46)	76.94 (95.11)	69.55 (88.82)	69.25 (88.48)	94.14 (97.92)	77.66 (92.38)	49.48 (72.33)
w/o KPNs	75.86 (94.51)	71.47 (91.73)	75.67 (94.28)	68.34 (88.12)	68.08 (87.69)	93.16 (97.47)	76.39 (92.01)	48.22 (71.32)
re. JT	76.34 (94.73)	71.89 (91.92)	76.13 (94.50)	68.82 (88.32)	68.53 (87.87)	93.67 (97.64)	76.86 (92.20)	48.71 (71.51)
w/o DN	75.88 (94.54)	71.75 (91.42)	74.98 (93.86)	68.82 (87.95)	68.12 (87.75)	93.52 (97.63)	76.56 (92.08)	48.45 (71.51)
re. SSN	76.13 (94.66)	71.89 (91.57)	75.07 (93.95)	68.73 (87.79)	68.11 (87.77)	93.70 (97.68)	76.85 (92.19)	48.74 (71.69)

Table 1: Entity mention and type matching F1 (%) comparisons in the 8 target domain NER datasets. We report the entity mention matching F1 (%) in brackets. † means directly fine-tuning the corresponding model on the target domain datasets. ‡ means we product the code with the given parameters. Results of our model are averaged over three runs with different seeds. *PE*: Potential Entities, *KPNs*: Knowledge Propagation Networks, *JT*: Joint Training, *DN*: Decomposed Network, *SSN*: Structured Semantic Network.

hyperparameters on dev sets. For the experiments in both English and Chinese datasets, we provide an overview of the hyperparameters in all trained models. Specifically, we trained these models for 5 epochs in source domains and 15 epoches in the target domain with learning rate 5×10^{-5} and batch size 32. For the first 10% epochs, we used a linear warmup learning rate strategy. For the KPN module, we initialize α_j with a small value of 0.05 and set its dropout rate to 0.15. Furthermore, to achieve better convergence, we used a $10\times$ higher learning rate for it compared to other modules.

Results and Analysis

Overall Performance. Table 1 shows the F1 results for entity mention and type matching in the eight NER datasets for the target domain. Almost all methods could gain performance improvements from the source domain datasets when compared with the models that only fine-tune on the target domain datasets. Especially in the AI domain, our method can achieve an incredible 19.59% improvement in F1 compared to directly fine-tuning BERT-CRF (50.46 vs. 70.05). Furthermore, we could observe that our method consistently outperforms all baseline models (with the Student's T test $p < 0.05$). More specifically, compared to the previous SOTA model: DH-GAT, our method on average achieves 0.66% higher F1 in English target domain datasets and 1.39% higher F1 in Chinese target domain datasets. When considering entity mention matching F1, our method also obtains higher performance than all the baselines. Furthermore, we find that the entity mention matching F1 and the entity mention and type matching F1 are positively correlated. Therefore, we can attribute part of the good effect of our method to: In the Decomposed Network, we provide higher-quality entity mentions for entity typing task by unifying the training paradigms of entity mention task.

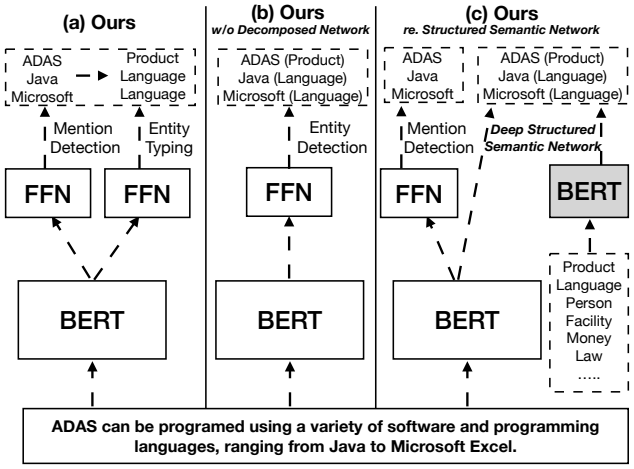


Figure 3: Ablation study of the Decomposed Network.

Ablation Study. We explore ablation studies to highlight the Decomposed Network and progressive procedure's efficacy. For the progressive procedure, Ours *w/o PE* omits entities copied after target domain sentences. Ours *w/o KPNs* excludes Knowledge Progressive Networks, signifying the removal of knowledge from multiple source domains, leaving only the last source domain's shared parameters. Ours *re.*

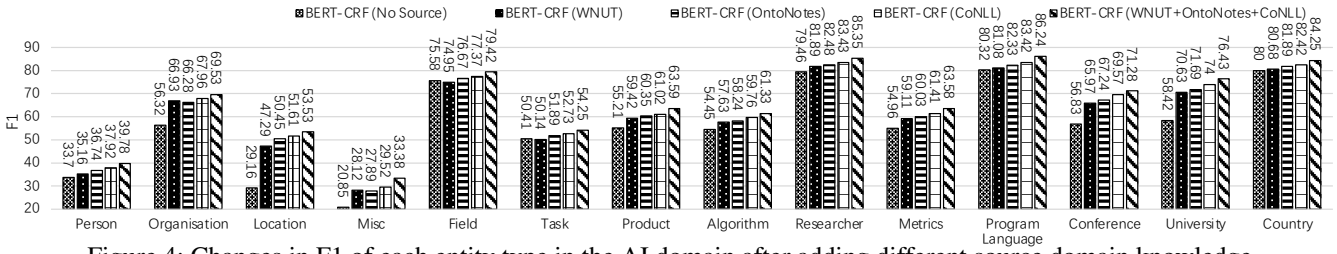


Figure 4: Changes in F1 of each entity type in the AI domain after adding different source domain knowledge.

JT deletes the KPNs but combines all source data, utilizing the Decomposed Network for knowledge transfer. Regarding the Decomposed Network, we suggest two variants. As depicted in Figure 3, Ours *w/o DN* merges mention detection and entity typing tasks for simultaneous predictions. In Ours *re. SSN*, we aim to standardize the cross-domain training for entity typing by adding the structured semantic network. The entitie types get semantic representations using BERT with fixed parameters, and we compute the cosine similarity between mention features and type representations:

$$\hat{\boldsymbol{y}}_{i,j}^{\mathrm{ET}} = \underset{\boldsymbol{m}_1,\boldsymbol{m}_2,\cdots,\boldsymbol{m}_Z}{\operatorname{argmax}} \frac{\exp\left(\cos\left(\boldsymbol{e}_i,\boldsymbol{m}_j\right)\right)}{\sum_{z=1}^{Z}\exp\left(\cos\left(\boldsymbol{e}_i,\boldsymbol{m}_z\right)\right)}, \quad (14)$$

where Z is the number of entity types, $\boldsymbol{m}_z \in \mathbb{R}^d$ is the semantic representation of the z-th type.

Table 1 shows all modules boost performance. Specifically, lacking potential entity signals and KPNs, Ours *w/o PE* and Ours *w/o KPNs* perform 1.10% and 2.37% worse on average. This aligns with using more relevant knowledge from the source domain aiding the target domain model. Also, training that retains source domain features outperforms joint training (Ours *re. JT*) by 1.88% on average.

The Decomposed Network gives 2.26% performance boost on average over all datasets compared to the independent training paradigm alternatives (Ours *w/o DN*). Although we unify the training paradigm for the entity typing task across domains (Ours *re. SSN*), the performance of the model drops by 2.14% on average. One reason is that the semantic features of the queried entity types cannot be fully conveyed by the fixed parameter BERT and insufficiently informative entity types.

Source Domain Analysis. We study the effect of the number and order of source domains on the target domain model. The number of different source domains determines the amount of knowledge we can transfer from the source domain, and the different order determines which source domain model shares parameters with the target domain. From Table 2, we could observe that **(1)** All source domains are helpful to the performance improvement of the target domain. The improvements brought about by WNUT 17, OntoNotes 5.0, and CoNLL 2003 are 8.98%, 10.27%, and 11.65%, respectively. **(2)** More source domains can bring more improvements. **(3)** Different sequences of the source domain bring different improvements to the target domain, and sharing parameters between the model of the CoNLL 2003 and the target domain model can bring the greatest improvement. We give the results of the Chinese cross-domain NER datasets in Table 3. We are able to draw similar conclusions to the cross-domain dataset for English: **(1)** All source domains are helpful to the performance improvement of the target domain. The improvements brought about by OntoNotes 5.0 and MSRA are 1.36% and 3.40%, respectively. **(2)** More source domains can bring more improvements. **(3)** Different source domain sequences bring different improvements to the target domain, and sharing parameters between the model of the MSRA and the target domain model (OntoNotes 5.0 $\rightarrow$ MSRA $\rightarrow$ Target Domains) can bring the greatest improvement. A very natural question arises: **How and to what extent the model in the source domain affects the performance of the target domain model?**

We adopt the source domain perturbation sensitivity (SDPS) and source domain fisher sensitivity (SDFS) introduced in Section to measure in which source domain knowledge transfer occurs. From Figure 5, we observe that SDPS

Source Domains			Target Domains				
W	O	C	Politics	Science	Music	Litera.	AI
✔	✔	✔	65.79	63.42	65.53	60.24	50.46
✗	✔	✔	72.94	68.76	72.55	65.81	65.20
✔	✗	✔	75.24	71.03	74.78	68.11	67.59
✔	✔	✗	76.69	72.37	76.04	69.69	68.99
✗	✗	✔	75.49	71.34	75.09	68.67	67.92
✗	✔	✗	77.02	72.89	76.42	69.87	69.45
✔	✗	✗	77.34	73.29	76.82	70.09	69.70
①	②	③	**77.82**	73.63	77.57	70.39	**70.05**
①	③	②	77.39	73.34	77.18	69.99	69.85
②	①	③	77.69	**73.89**	**77.60**	**70.43**	69.94
②	③	①	77.19	73.08	77.24	69.89	69.85
③	①	②	77.50	73.23	77.44	69.67	69.74
③	②	①	77.07	73.19	77.35	69.71	69.72

Table 2: Sequential analysis of source domains. ①, ②, and ③ indicate the usage order of the source domain data: W for WNUT, O for OntoNotes, and C for CoNLL.

Source Domains		Target Domains		
OntoNotes	MSRA	Resume	Address	Biomedical
✗	✗	91.33	75.44	45.68
✔	✗	92.76	76.32	47.44
✗	✔	94.52	78.25	49.89
①	②	**95.31**	**79.29**	**52.07**
②	①	94.78	78.73	50.63

Table 3: Sequential analysis of source domains. ① and ② indicate the usage order of the source domain data.

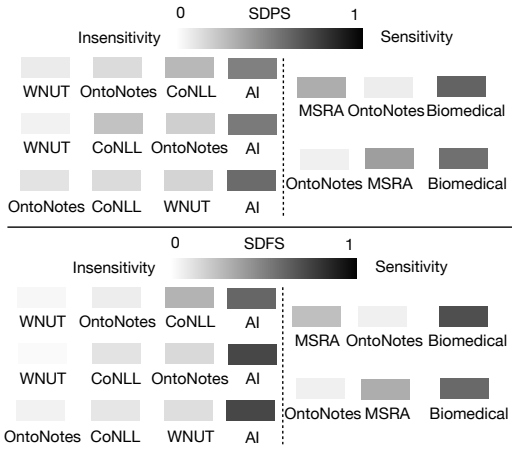


Figure 5: Comparison of per-domain sensitivities obtained with the SDPS and SDFD methods.

and SDFS can almost draw similar conclusions, all target domain models are sensitive to the source domains. In summary, for the English AI domain and the Chinese Biomedical domain, CoNLL 2003 and MSRA have the highest sensitivity, while WNUT and OntoNotes have the lowest, which means that among all the source domains, CoNLL 2003 and MSRA have transferred the most knowledge to the target domain. This finding is highly correlated with our experimental results: the model from CoNLL 2003 and MSRA can give the greatest boost to the target domain.

Analyze Target Domain Entity Type. We try to answer how source domain knowledge helps the target domain model by showing the change in F1 performance of each entity type in the AI domain after adding different source domain knowledge. As shown in Figure 4, we find that all entity types in the AI domain have an improvement in F1 after introducing the knowledge of the source domain. Among them, the types that are closer to the existing entity types in the source domain: for example, from `Location` (CoNLL 2003) to `University` (AI), from `Organization` (CoNLL 2003) to `Conference` (AI) can get a greater improvement.

Case study. We give three cases in Table 4. When we remove the source domain datasets, the ability of the model to recognize entities is reduced and even cannot recognize *Troponymy* as an entity. When we delete the copied potential entities, WNUT and OntoNotes predict that *Siri* is a `Product` prompt that cannot be delivered to the target domain. Due to the scarce training labeled data and lack of knowledge, the model cannot avoid mispredicting *Siri* as a `Program-Language` label. After adding source domain data, the model often confuses similar entities. For example, in the third example, when predicting *unsupervised classification*, it is affected by entity *unsupervised learning*, so the wrong prediction entity type becomes `Field`, but not `Task`.

Related Work

Cross-domain NER is a pivotal task in low-resource information extraction which aims at transferring knowledge from data-rich source domains to sparsely labeled target domains.

Unrecognized Entity	***Troponymy*** is one of the possible relations between verbs in the semantic network of the WordNet database. Label: **`Miscellaneous`** Prediction w/o Sources: **`O` (Not an entity)** Prediction w. Sources: **`Miscellaneous`**
Lack of Knowledge	A special case of keyword spotting is wake word detection used by personal digital assistants such as Alexa or ***Siri***... Label: **`Product`** Prediction w/o Potential Entities: **`Program-Language`** Prediction w. Potential Entities: **`Product`**
Similar Entity Confusion	Categorization tasks in which no labels are supplied are referred to as Cluster analysis, ***unsupervised classification***, unsupervised learning ... Label: **`Task`** Prediction w/o Sources: **`Field`** Prediction w. Sources: **`Field`**

Table 4: Predictions with/without source domain datasets or the Potential Entities in the AI domain. We mark the *entity*.

Cross-domain methods can be used for data mining (Chen et al. 2022b, 2023a), recommendation systems (Chen et al. 2023c,b, 2022a), information extraction (Hu et al. 2020, 2021a,b, 2023a,b), etc. Previous efforts attempted to find cross-domain invariant features in label semantics (Wang et al. 2018) and model parameters (Liu et al. 2021).

The methods of label semantic transfer attempt to align label features across domains and transfer label representation across domains (Kim et al. 2015). Wang et al. (2018) adopted a variant of the maximum mean discrepancy (MMD) for the label-aware double transfer learning framework. Liu et al. (2020) studied the coarse-to-fine representation in entity type label representations and proposed a two-stage pipeline model. Zhang et al. (2022b) explored the semantic transfer of labels simultaneously in the entity span and the type space, thus achieving smaller discrepancies in the cross-domain transfer. The methods of model parameter transfer aim to share model parameters between different domains through knowledge distillation (Yang et al. 2019; Nguyen, Gelli, and Poria 2021; Zhang et al. 2021), domain prediction tasks (Lin and Lu 2018; Zhou et al. 2019; Jia and Zhang 2020; Xu and Cai 2023) or generative methods (Jia, Liang, and Zhang 2019; Chen et al. 2021; Xu et al. 2023). However, these methods neglect to request more source domain data to explicitly and implicitly transfer knowledge to the target domain.

Conclusions

In this paper, we propose a progressive decomposed network to transfer knowledge of multiple source domains to the target domain. To analyze in which source domain knowledge transfer occurs, we propose two methods named SDPS and SDFS. Experiments on eight public datasets across two languages show the effectiveness of our model.

Acknowledgements

This work is supported in part by NSF under grant III-2106758.

References

Alibaba, D. A. 2021. CCKS2021 Chinese Address Element Analysis Dataset.

Amari, S.-I. 1998. Natural gradient works efficiently in learning. *Neural computation*, 10(2): 251–276.

Asghar, N.; Mou, L.; Selby, K. A.; Pantasdo, K. D.; Poupart, P.; and Jiang, X. 2020. Progressive Memory Banks for Incremental Domain Adaptation. In *International Conference on Learning Representations*.

Chen, S.; Aguilar, G.; Neves, L.; and Solorio, T. 2021. Data Augmentation for Cross-Domain Named Entity Recognition. In *Proc. of EMNLP*, 5346–5356.

Chen, Y.; Fang, Y.; Zhang, Y.; and King, I. 2023a. Bipartite Graph Convolutional Hashing for Effective and Efficient Top-N Search in Hamming Space. In *Proc. of WWW*, 3164–3172.

Chen, Y.; Guo, H.; Zhang, Y.; Ma, C.; Tang, R.; Li, J.; and King, I. 2022a. Learning binarized graph representations with multi-faceted quantization reinforcement for top-k recommendation. In *Proc. of SIGKDD*, 168–178.

Chen, Y.; Truong, Q.-T.; Shen, X.; Wang, M.; Li, J.; Chan, J.; and King, I. 2023b. Topological Representation Learning for E-commerce Shopping Behaviors. In *Proceedings of the 19th International Workshop on Mining and Learning with Graphs (MLG)*.

Chen, Y.; Zhang, Y.; Guo, H.; Tang, R.; and King, I. 2022b. An Effective Post-training Embedding Binarization Approach for Fast Online Top-K Passage Matching. In *AACL*, 102–108.

Chen, Y.; Zhang, Y.; Yang, M.; Song, Z.; Ma, C.; and King, I. 2023c. WSFE: Wasserstein Sub-graph Feature Encoder for Effective User Segmentation in Collaborative Filtering. 2521–2525.

Derczynski, L.; Nichols, E.; van Erp, M.; and Limsopatham, N. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 140–147.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL-HLT*, 4171–4186.

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4): 128–135.

Fujinuma, Y.; Boyd-Graber, J.; and Kann, K. 2022. Match the Script, Adapt if Multilingual: Analyzing the Effect of Multilingual Pretraining on Cross-lingual Transferability. In *Proc. of ACL*, 1500–1512.

Hu, J.; Zhao, H.; Guo, D.; Wan, X.; and Chang, T.-H. 2022. A Label-Aware Autoregressive Framework for Cross-Domain NER. In *Proc. of NAACL: Findings*, 2222–2232.

Hu, X.; Hong, Z.; Zhang, C.; Liu, A.; Meng, S.; Wen, L.; King, I.; and Yu, P. S. 2023a. Reading broadly to open your mind improving open relation extraction with search documents under self-supervisions. *IEEE Transactions on Knowledge and Data Engineering*.

Hu, X.; Liu, A.; Tan, Z.; Zhang, X.; Zhang, C.; King, I.; and Yu, P. S. 2023b. GDA: Generative Data Augmentation Techniques for Relation Extraction Tasks. *arXiv preprint arXiv:2305.16663*.

Hu, X.; Wen, L.; Xu, Y.; Zhang, C.; and Yu, P. S. 2020. SelfORE: Self-supervised Relational Feature Learning for Open Relation Extraction. In *Proc. of EMNLP*, 3673–3682.

Hu, X.; Zhang, C.; Ma, F.; Liu, C.; Wen, L.; and Yu, P. S. 2021a. Semi-supervised Relation Extraction via Incremental Meta Self-Training. In *Findings of EMNLP*, 487–496.

Hu, X.; Zhang, C.; Yang, Y.; Li, X.; Lin, L.; Wen, L.; and Yu, P. S. 2021b. Gradient Imitation Reinforcement Learning for Low Resource Relation Extraction. In *Proc. of EMNLP*, 2737–2746.

Hu, Z.; Jiang, Y.; Bach, N.; Wang, T.; Huang, Z.; Huang, F.; and Tu, K. 2021c. Multi-View Cross-Lingual Structured Prediction with Minimum Supervision. In *Proc. of ACL-IJCNLP*, 2661–2674.

Imani, A.; Severini, S.; Sabet, M. J.; Yvon, F.; and Schütze, H. 2022. Graph-Based Multilingual Label Propagation for Low-Resource Part-of-Speech Tagging. In *Proc. of EMNLP*, 1577–1589.

Jia, C.; Liang, X.; and Zhang, Y. 2019. Cross-domain NER using cross-domain language modeling. In *Proc. of ACL*, 2464–2474.

Jia, C.; and Zhang, Y. 2020. Multi-cell compositional LSTM for NER domain adaptation. In *Proc. of ACL*, 5906–5917.

Kim, Y.-B.; Stratos, K.; Sarikaya, R.; and Jeong, M. 2015. New transfer learning techniques for disparate label sets. In *Proc. of ACL*, 473–482.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural Architectures for Named Entity Recognition. In *Proc. of NAACL-HLT*, 260–270.

Levow, G.-A. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 108–117.

Li, J.; Sun, A.; Han, J.; and Li, C. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1): 50–70.

Lin, B. Y.; and Lu, W. 2018. Neural Adaptation Layers for Cross-domain Named Entity Recognition. In *Proc. of EMNLP*, 2012–2022.

Liu, P.; Guo, Y.; Wang, F.; and Li, G. 2022. Chinese named entity recognition: The state of the art. *Neurocomputing*, 473: 37–53.

Liu, Z.; Winata, G. I.; Xu, P.; and Fung, P. 2020. Coach: A Coarse-to-Fine Approach for Cross-domain Slot Filling. In *Proc. of ACL*, 19–25.

Liu, Z.; Xu, Y.; Yu, T.; Dai, W.; Ji, Z.; Cahyawijaya, S.; Madotto, A.; and Fung, P. 2021. Crossner: Evaluating cross-domain named entity recognition. In *Proc. of AAAI*, volume 35, 13452–13460.

Luo, Y.; Xiao, F.; and Zhao, H. 2020. Hierarchical contextualized representation for named entity recognition. In *Proc. of AAAI*, volume 34, 8441–8448.

Nasar, Z.; Jaffry, S. W.; and Malik, M. K. 2021. Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys (CSUR)*, 54(1): 1–39.

Nguyen, H.-V.; Gelli, F.; and Poria, S. 2021. DOZEN: cross-domain zero shot named entity recognition with knowledge graph. In *Proc. of SIGIR*, 1642–1646.

Pradhan, S.; Moschitti, A.; Xue, N.; Ng, H. T.; Björkelund, A.; Uryupina, O.; Zhang, Y.; and Zhong, Z. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, 143–152.

Sang, E. T. K.; and De Meulder, F. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proc. of HLT-NAACL*, 142–147.

Tang, M.; Zhang, P.; He, Y.; Xu, Y.; Chao, C.; and Xu, H. 2022. DoSEA: A Domain-specific Entity-aware Framework for Cross-Domain Named Entity Recogition. In *Proc. of COLING*, 2147–2156.

Wang, Z.; Qu, Y.; Chen, L.; Shen, J.; Zhang, W.; Zhang, S.; Gao, Y.; Gu, G.; Chen, K.; and Yu, Y. 2018. Label-Aware Double Transfer Learning for Cross-Specialty Medical Named Entity Recognition. In *Proc. of NAACL-HLT*, 1–15.

Xu, J.; and Cai, Y. 2023. Decoupled Hyperbolic Graph Attention Network for Cross-domain Named Entity Recognition. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 591–600.

Xu, J.; Zheng, C.; Cai, Y.; and Chua, T.-S. 2023. Improving Named Entity Recognition via Bridge-based Domain Adaptation. In *Findings of the Association for Computational Linguistics: ACL 2023*, 3869–3882.

Yang, H.; Huang, S.; Dai, X.; and Chen, J. 2019. Fine-grained Knowledge Fusion for Sequence Labeling Domain Adaptation. In *Proc. of EMNLP-IJCNLP*, 4197–4206.

Yang, Z.; Salakhutdinov, R.; and Cohen, W. W. 2017. Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks. In *ICLR (Poster)*.

Zhang, N.; Chen, M.; Bi, Z.; Liang, X.; Li, L.; Shang, X.; Yin, K.; Tan, C.; Xu, J.; Huang, F.; et al. 2022a. CBLUE: A Chinese Biomedical Language Understanding Evaluation Benchmark. In *Proc. of ACL*, 7888–7915.

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 *Proc. of EMNLP*, 5441–5451.

Zhang, X.; Yu, B.; Wang, Y.; Liu, T.; Su, T.; and Xu, H. 2022b. Exploring Modular Task Decomposition in Cross-domain Named Entity Recognition. In *Proc. of SIGIR*, 301–311.

Zhang, Y.; and Yang, J. 2018. Chinese NER Using Lattice LSTM. In *Proc. of ACL*, 1554–1564.

Zheng, J.; Chen, H.; and Ma, Q. 2022. Cross-domain Named Entity Recognition via Graph Matching. In *Proc. of ACL: Findings*, 2670–2680.

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