A Label-Aware Autoregressive Framework for Cross-Domain NER

Jinpeng Hu$^{\heartsuit}$, **He Zhao**$^{\clubsuit}$ **Dandan Guo**$^{\spadesuit\dagger}$,
Xiang Wan$^{\heartsuit\diamondsuit\dagger}$, **Tsung-Hui Chang**$^{\heartsuit}$
$^{\heartsuit}$Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong,
Shenzhen, Guangdong, China
$^{\spadesuit}$The Chinese University of Hong Kong, Shenzhen
$^{\clubsuit}$Monash University, Australia $^{\diamondsuit}$Pazhou Lab, Guangzhou, 510330, China
{jinpenghu@link, guodandan, changtsunghui}@cuhk.edu.cn
ethan.zhao@monash.edu wanxiang@sribd.cn

Abstract

Cross-domain named entity recognition (NER) aims to borrow the entity information from the source domain to help the entity recognition in the target domain with limited labeled data. Despite the promising performance of existing approaches, most of them focus on reducing the discrepancy of token representation between source and target domains, while the transfer of the valuable label information is often not explicitly considered or even ignored. Therefore, we propose a novel autoregressive framework to advance cross-domain NER by first enhancing the relationship between labels and tokens and then further improving the transferability of label information. Specifically, we associate each label with an embedding vector, and for each token, we utilize a bidirectional LSTM (Bi-LSTM) to encode the labels of its previous tokens for modeling internal context information and label dependence. Afterward, we propose a Bi-Attention module that merges the token representation from a pre-trained model and the label features from the Bi-LSTM as the label-aware information, which is concatenated to the token representation to facilitate cross-domain NER. In doing so, label information contained in the embedding vectors can be effectively transferred to the target domain, and Bi-LSTM can further model the label relationship among different domains by pre-train and then fine-tune setting. Experimental results on several datasets confirm the effectiveness of our model, where our model achieves significant improvements over existing methods.[1]

1 Introduction

Named entity recognition (NER) is a fundamental task in natural language processing (NLP), aiming to identify salient information from raw texts, such as persons, locations, and so on. NER can be viewed as a specific sequence labeling problem, where models built upon pre-trained language models have recently achieved significant improvements. However, most conventional approaches trained on specific domains (source domains) are hard to generalize to new domains (target domains) due to the differences in text genre and limitation of labeled data. Thus, cross-domain NER has been proposed for alleviating this problem, which aims to learn information from the source domain to enhance NER in the target domain.

For example, Jia et al. (2019) utilized a parameter generation network to combine cross-domain language modeling and NER, thereby enhancing the model to extract knowledge of domain differences from raw texts. Furthermore, Liu et al. (2020b); Gururangan et al. (2020) proposed to continue pre-training the language models on the target domain-related corpus. Despite the outstanding performance, existing approaches mainly focus on handling the text discrepancy between different domains and apply Conditional Random Fields (CRF) (Lafferty et al., 2001) to capture label-label dependence in neighbor tags. Several issues cannot be appropriately solved. First, most of them rely heavily on the powerful encoder to implicitly extract token-label relationships due to the limitation of the sequence labeling framework, which is insufficient, especially for the limited data in a new domain, where the encoder is hard to be fully trained. In cross-domain NER, token-label relationships are more critical since better token-label interaction can help the model distinguish the differences and similarities between the two domains. For example, in the general domain, "Bayes" usually is a "*person*" entity, while in the artificial intelligence (AI) domain, for "supervised learning are Naive Bayes classifier", "Bayes" is an "*algorithm*" entity. Clearly, if a model is aware that the NE label of the previous phrase "supervised learning" is an "*AI field*" entity and thus pays more attention to this

[†] Corresponding author.

[1] Our code is released at https://github.com/jinpeng01/LANER.

Findings of the Association for Computational Linguistics: NAACL 2022, pages 2222 - 2232
July 10-15, 2022 ©2022 Association for Computational Linguistics

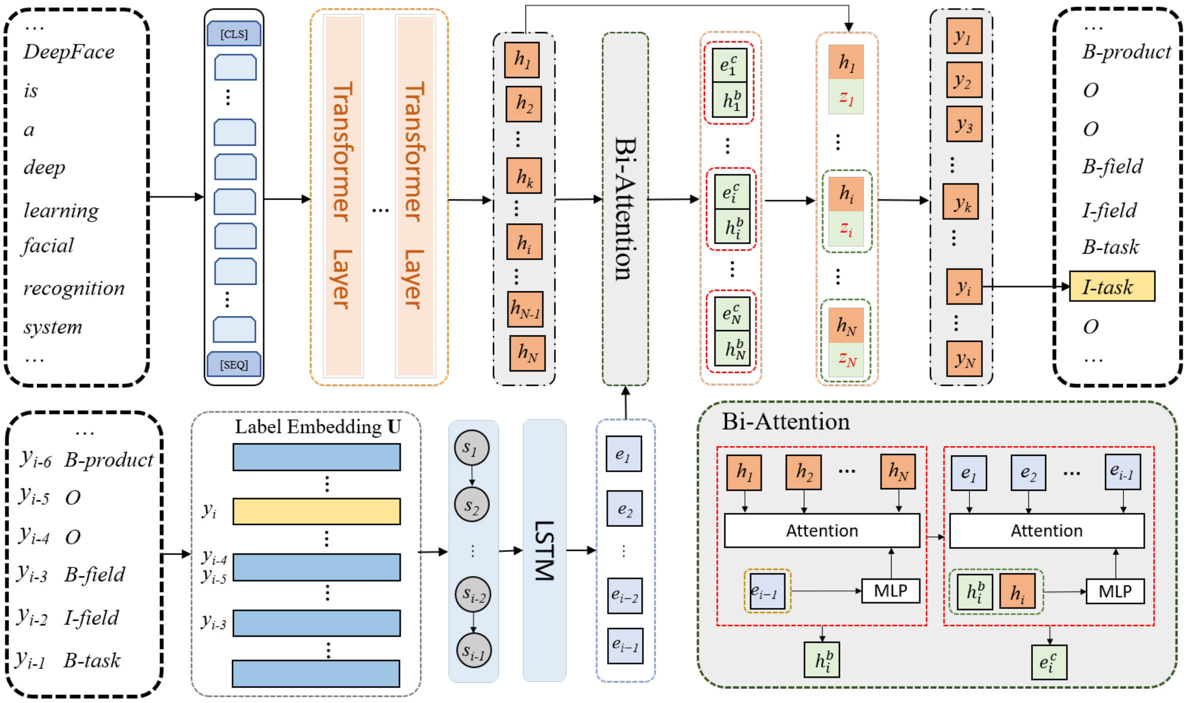

Figure 1: The overall architecture of our proposed model. The upper part is the general sequence labeling model paradigm, and the left bottom part is used to extract label-aware information (i.e., $\mathbf{z}_i$) and the right bottom part reveals the Bi-Attention structure.

phrase, predicting “Bayes” as an “*algorithm*” entity shall be an easier task. Such previous label information (e.g., labels of “supervised learning”) can be explicitly used to help the model enhance the relationship between tokens and labels instead of purely depending on the encoder itself. Second, remote label-label relationships and label semantic information are also important for cross-domain NER instead of only modeling the interdependency among adjacent labels as CRF does. For example, for “Like a Girl which is from the Scenery and Fish”, “Like a Girl” is an “*album*” entity and “Scenery and Fish” is a “*song*” entity. Although they are not adjacent, since entity type ‘*album*” is semantically close to “*song*”, they tend to exist in the same sentence. When the model has predicted an “*album*” entity, it will pay more attention to the “*song*” entity during predicting other words in the sentence, which is a type of helpful supplementary information. Third, when two domains share the same NE types, these shared labels usually represent similar meanings such that they are easily adapted to the target domain, which is paid less attention in previous studies. Therefore, fully using shared NE labels and further appropriately modeling the correlations between the shared labels and target domain-related ones are also beneficial to advance cross-domain NER.

In this paper, we propose a novel autoregressive cross-domain NER framework to help the model facilitate domain adaptation by improving the relationship between the source text and its named entity (NE) labels and enhancing label information transfer. In detail, we associate each label with an embedding vector (randomly initialized and learned later), and for each token in the original text, we input the embeddings of the label sequence generated from previous steps into a bidirectional LSTM (Bi-LSTM), whose hidden states model label sequence information. Next, we propose a Bi-Attention module to perform two attention between token representations from a pre-trained model and label features from the Bi-LSTM to calculate label background and context information and then concatenate them as the label-aware information. We then fuse label-related knowledge into current token representation for promoting cross-domain NER. In doing so, our model can learn label embeddings and the potential relationship between tokens and labels by pre-training on the source domain, especially for shared entity labels, and then adapt them to the target domain by fine-tuning. Experimental results on several datasets show that our approach outperforms existing studies.

2 Method

2.1 Problem Definition

NER can be conventionally performed as a sequence labeling problem (Lample et al., 2016; Luo et al., 2020), where named entities can be viewed as labels of tokens. Specifically, given an input sequence $\mathcal{X}=\{x_1, x_2, \cdots, x_N\}$ with N tokens, the goal of NER is to output the corresponding label sequence $\mathcal{Y}=\{y_1, y_2, \cdots, y_N\}$ with the same length, i.e., modeling $p(\mathcal{Y} \mid \mathcal{X})$. In the cross-domain NER task, we are given two datasets from the source and target domains, denoted as $\mathcal{D}_{src}$ and $\mathcal{D}_{tgt}$, respectively. The aim is to learn valuable knowledge from $\mathcal{D}_{src}$ and transfer it to the target domain $\mathcal{D}_{tgt}$.

Many existing (cross-domain) NER models (Liu et al., 2020b; Jia and Zhang, 2020; Lin and Lu, 2018) predict a token's entity purely based on the context of the sequence, and they formulate $p(\mathcal{Y} \mid \mathcal{X}) = \prod_{i=1}^{N} p\left(y_i \mid \mathcal{X}\right)$. However, these approaches pay less attention to the label information and the relationship between labels and tokens. To explicitly enhance such relationship and capture label information, we propose a novel framework to predict NE labels by utilizing both previous labels and token representation, which can be formulated as an autoregressive model:

$$p(\mathcal{Y} \mid \mathcal{X})=\prod_{i=1}^{N} p\left(y_i \mid y_1, \ldots, y_{i-1}, \mathcal{X}\right). \tag{1}$$

In the cross-domain setting, such information can be extended between the labels in the source and target domains, which is an effective way of transferring knowledge to the target domain.

2.2 Proposed Model

As mentioned above, our proposed model consists of three main parts: the input sequence encoder that encodes the input sequence $\mathcal{X}$, the label encoder that encodes the previous tokens' labels $y_1, \cdots, y_{i-1}$, and the label predictor that predicts NER labels of tokens. An overview of our proposed model is shown in Figure 1, whose details are introduced as follows.

2.2.1 Input Sequence Encoder

Following many other cross-domain NER methods (e.g., Liu et al. (2020b)), we use a pre-trained BERT (Devlin et al., 2019) model denoted as $f_{\text{dte}}(\cdot)$ to encode the input sequence:

$$[\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_N] = f_{\text{dte}}(x_1, x_2, \cdots, x_N), \tag{2}$$

where $\mathbf{h}_i$ is a d_1-dimensional vector for each token x_i, which is expected to capture the contextual information of the corresponding token.

2.2.2 Label Encoder

To model the relations between the token sequence and label sequence, we propose a novel label encoder to extract the contextual information from the label sequence. An important distinction of our work from most previous approaches is that we predict the NE labels based on both commonly-used current token representation (i.e., $\mathbf{h}_i$) and *label-aware information* extracted from the previous labels (i.e., $y_{1:i-1}$[2]). Intuitively, the process of generating labels has a flavor of the sequence-to-sequence decoders. In detail, we first construct a randomly initialized label lookup table $\mathbf{U} \in \mathrm{R}^{K*d_2}$, where K denotes the number of unique labels in source or target domains, and d_2 is the size of label embedding. For a label y_k with $k \in \{1 : K\}$, we can embed it to $\mathbf{s}_k \in \mathrm{R}^{d_2}$ by using $\mathbf{U}$. To fully utilize the label-related knowledge for the current token x_i, we employ a Bi-LSTM (Hochreiter et al., 1997) to encode the label sequence (i.e., $y_{1:i-1}$) , expressed as:

$$[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{i-1}] = f_{\text{re}}(\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_{i-1}), \tag{3}$$

where $f_{\text{re}}(\cdot)$ is the label sequence encoder (i.e., Bi-LSTM) and $\mathbf{e}_k \in \mathrm{R}^{2d_2}$ is the output of the Bi-LSTM for $k \in \{1 : i-1\}$, which is expected to capture the contextual information of previous labels.

2.2.3 Label Predictor

The label predictor is to leverage the contextual information of both the input sequence and previous label sequence to predict the NER labels. To merge the two kinds of information, we introduce a simple yet effective Bi-Attention module. Specifically, following (Wang et al., 2016), we regard the last hidden state of Bi-LSTM in the label encoder (i.e., $\mathbf{e}_{i-1}$ in Equation (3)) as the representation of label sequence, which severs as the query vector, while all token representations from the input sequence encoder (i.e., $\mathbf{h}_{1:N}$ in Equation (2)) are viewed as the key and value matrices. Before performing the matrix product, we apply a fully connected layer to project the $\mathbf{e}_{i-1}$ into the same dimension as the $\mathbf{h}_i$:

$$\mathbf{e}'_{i-1} = \mathbf{W}_2 \cdot \mathbf{e}_{i-1} + \mathbf{b}_2, \tag{4}$$

[2] In the training stage, the previous labels are from the ground truth while they are predicted by our model in the test stage.

where $\mathbf{e}'_{i-1}$ is a d_1-dimensional vector. We then compute the attention weight with the softmax function:

$$\mathbf{a}_i^b = \mathrm{Softmax}(\mathbf{e}'_{i-1}\mathbf{h}^{\mathrm{T}}). \tag{5}$$

Herein, $\mathbf{a}_i^b$ can be viewed as a probability distribution and used to produce a weighted sum over the input token representations (i.e., $[\mathbf{h}_{1:N})$:

$$\mathbf{h}_i^b = \sum_k^N a_{i,k}^b \mathbf{h}_k. \tag{6}$$

Since label background information $\mathbf{h}_i^b$ is guided by $\mathbf{e}_{i-1}$, it is naturally to represent the relationship between the label of current token (i.e., y_i) and the whole input sequence. In addition, it is also necessary to capture the relationship between the current token x_i and previously predicted labels (i.e., $y_{1:i-1}$), which can improve the sensitivity of x_i to previous NE tags. We first concatenate token representation $\mathbf{h}_i$ and label background information $\mathbf{h}_i^b$ as a comprehensive intermediate state, which is further mapped to a $2d_2$-dimensional vector:

$$\mathbf{h}'_i = \mathbf{W}_3 \cdot \mathbf{h}_i \oplus \mathbf{h}_i^b + \mathbf{b}_3. \tag{7}$$

Below, we still adopt a simple attention module to compute the label context information:

$$\mathbf{e}_i^c = \sum_k^{i-1} a_{i,k}^c \mathbf{e}_k, \mathbf{a}_i^c = \mathrm{Softmax}(\mathbf{h}'_i\mathbf{e}^{\mathrm{T}}). \tag{8}$$

where $\mathbf{a}_i^c$ indicates the weight vector of token i over $\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{i-1}$. Finally, we concatenate label background information over input sequence and context information over predicted NE labels as the final label-aware information $\mathbf{z}_i$:

$$\mathbf{z}_i = \mathbf{h}_i^b \oplus \mathbf{e}_i^c. \tag{9}$$

Note that, $\mathbf{W}_2$, $\mathbf{W}_3$, $\mathbf{b}_2$ and $\mathbf{b}_3$ are learnable parameters.

To further fuse label-related knowledge into token x_i, we concatenate h_i and corresponding *label-aware information* $\mathbf{z}_i$, formulated as:

$$\mathbf{u} = [\mathbf{h}_1 \oplus \mathbf{z}_1, \mathbf{h}_2 \oplus \mathbf{z}_2, \cdots, \mathbf{h}_n \oplus \mathbf{z}_N], \tag{10}$$

where $\mathbf{u}$ is the final sequence representation. Finally, we apply a trainable matrix $\mathbf{W}_o$ and bias $\mathbf{b}_o$ to map the $\mathbf{u}_i$ to the output space by $\mathbf{o}_i = \mathbf{W}_o \cdot \mathbf{u}_i + \mathbf{b}_o$ and utilize a softmax function to obtain the distribution with respect to all NE labels.

2.3 Pre-training and Fine-tuning in the Cross-domain Setting

To enhance text feature extraction in the target domain, we follow Liu et al. (2020b) to continue pre-training the input sequence encoder on the domain-related corpus to narrow the difference between source and target domain in terms of domain background and text distribution (Gururangan et al., 2020) and further capture more productive features from the target domain, which refers as domain-adaptive pre-training (DAPT). Moreover, to effectively transfer information to the target domain, we train our model in two stages: pre-training and fine-tuning. In detail, in the first stage, we train our model on $\mathcal{D}_{src}$ to learn text knowledge and enhance feature extractor. More importantly, this process can learn valuable label embeddings before accessing the target domain, especially for shared NE labels. At the second stage, we fine-tune our model on the target domain to adopt it to $\mathcal{D}_{tgt}$. Since we utilize a Bi-LSTM to encode label sequences with the help of the pre-trained shared label embeddings from the first stage, our model can further learn relations between the shared NE labels and target domain-specific NE labels (i.e., the labels that only exist in the target domain) as well as the intrinsic label dependency information. This can further help the model to leverage the knowledge of the source domain to better understand these unseen labels in the target domain.

DATA	NUM	TYPE	TRAIN	DEV	TEST
CONLL2003	4	#SENT. #ENT.	15.0k 23.4k	3.5k 5.9k	3.7k 5.6k
POLITICS	9	#SENT. #ENT.	0.2k 1.3k	0.5k 3.5k	0.7k 4.2k
SCIENCE	17	#SENT. #ENT.	0.2k 1.1k	0.5k 2.5k	0.5k 3.1k
MUSIC	13	#SENT. #ENT.	0.1k 0.6k	0.4k 2.7k	0.5k 3.3k
LITERATURE	12	#SENT. #ENT.	0.1k 0.5k	0.4k 2.1k	0.4k 2.3k
AI	14	#SENT. #ENT.	0.1k 0.5k	0.4k 1.5k	0.4k 1.8k
MOVIE	14	#SENT. #ENT.	7.8k 23.0k	- -	2.0k 5.7k
RESTAURANT	8	#SENT. #ENT.	7.7k 15.4k	- -	1.5k 3.2k

Table 1: The statistics of datasets, including the number of entity types (**NUM**), the number of sentences (#SENT.), and the number of entities (#ENT.).

3 Experimental Setting

3.1 Datasets

We conduct our experiments on the following datasets: Conll2003 (Sang and De Meulder, 2003), CrossNER (Liu et al., 2020b), MIT Movie (Movie)

MODEL	CONLL2003							
	POLITICS	SCIENCE	MUSIC	LITERATURE	AI	AVERAGE	MOVIE	RESTAURANT
w/o DAPT								
LSTM-CRF†	56.60	49.97	44.79	43.03	43.56	47.59	68.31*	78.13 *
CROSS-DOMAIN LM†	68.44	64.31	63.56	59.59	53.70	61.92	-	-
FLAIR	69.54	64.71	65.60	61.35	52.48	62.73	-	-
COACH†	61.50	52.09	51.66	48.35	45.15	51.75	67.62*	77.82*
BARTNER-BASE	69.90	65.14	65.35	58.93	53.00	62.46	71.55	79.53
MULTI-CELL LSTM†	70.56	66.42	70.52	66.96	58.28	66.55	69.41*	78.67*
OURS	**71.65**	**69.29**	**73.07**	**67.98**	**61.72**	**68.74**	**72.41**	**80.55**
Introducing DAPT								
MULTI-CELL LSTM + DAPT†	71.45	67.68	74.19	68.63	61.64	68.71	-	-
OURS+DAPT	**74.06**	**71.83**	**78.78**	**71.11**	**65.79**	**72.31**	**-**	**-**

Table 2: Comparisons of existing studies and our proposed models with respect to F1 scores. AVERAGE is the average F1 score of five domains in the CrossNER dataset. † indicates that the results are directly cited from Liu et al. (2020b) (except values with *). Results of our model are averaged over three runs with different seeds.

(Liu et al., 2013b) and MIT Restaurant (Restaurant) (Liu et al., 2013a), where the first one is regarded as the source domain dataset and the others are performed as the target domain datasets. Specifically, Conll2003 is a popular NER dataset collected from the Reuters Corpus and is tagged with four NE types, including PER, LOC, ORG and MISC. CrossNER is drawn from Wikipedia and contains five different domain datasets: politics, natural science, music, literature, and AI. Movie and Restaurant corpus consist of user utterances for movie and restaurant domains with 12 and 8 classes. For all datasets, we follow their official splits of training, validation, and test sets, and their statistics are summarized in Table 1. Note that in this paper, we employ the standard BIO scheme to represent each NE label.

3.2 Baselines and Evaluation Metrics

To explore the performance of our proposed model, we compare it to following main baselines:

- **BERT-TAGGER** (Devlin et al., 2019): This fine-tunes the BERT model with a label classifier.
- **DAPT-TAGGER** (Liu et al., 2020b): This first applies DAPT and then is directly fine-tuned on the cross-domain NER task.
- **BERT-CRF**, **DAPT-CRF** (Liu et al., 2020b): These have the same main architecture as **BERT-TAGGER** and **DAPT-TAGGER**, and the difference is that they incorporate a CRF layer.

We also compare our model to existing studies:

- **LSTM-CRF** (Lample et al., 2016): This proposes to combine character- and word-level features and utilize a bidirectional LSTM with a sequential CRF layer to perform NER.
- **FLAIR** (Akbik et al., 2018): This leverages the internal states of a character language model to produce contextual string embedding and then integrate them into the NER model.
- **COACH** (Liu et al., 2020a): This learns the slot entity pattern and combines the features for each slot entity to enhance entity types prediction.
- **CROSS-DOMAIN LM** (Jia et al., 2019): This employs a parameter generation network to combine cross-domain language modeling and NER, thereby enhancing The model performance.
- **MULTI-CELL LSTM** (Jia and Zhang, 2020): This utilizes a multi-cell compositional LSTM structure for enhancing NER domain adaptation.
- **BARTNER** (Yan et al., 2021): This formulates NER tasks as an entity span sequence generation problem and incorporates BART as their backbone (Lewis et al., 2020).

To make a fair comparison, we exploit F1 scores as the evaluation metric.

3.3 Implementation Details

In our experiments, our model is implemented based on transformers[3] and Liu et al. (2020b)[4]. We choose BERT-base-cased[5] as our input sequence encoder to extract the features from the source sequence and follow its default model setting where we use 12 layers of self-attention with 768-dimensional embeddings. The dimension of the label embedding (i.e., d_2) is set to 100, and the hidden size of LSTM is the same as the d_2, which is also set to 100. Other hyperparameters, including the learning rate, batch size, and the number of epochs, are reported in Appendix A.1. During the training process, we utilize Adam (Kingma and Ba, 2015) to optimize all the trainable parameters, including the ones in the pre-trained model. The

[3] https://github.com/huggingface/transformers

[4] https://github.com/zliucr/CrossNER

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

MODEL	SETTINGS	CONLL2003						MOVIE	RESTAURANT
		POLITICS	SCIENCE	MUSIC	LITERATURE	AI	AVERAGE		
BERT	LB+LC	**71.65**	69.29	**73.07**	**67.98**	**61.72**	**68.74**	**72.41**	**80.55**
BERT	w/o LC	70.94	**71.11**	71.51	67.24	59.23	67.93	72.26	80.35
BERT	w/o LB	70.61	68.43	70.07	67.53	59.41	67.21	70.79	79.08
BERT+CRF	w/o LB+LC	70.47	66.77	70.34	67.15	58.03	66.55	69.92	78.74
BERT-TAGGER†	w/o LB+LC	68.71	64.94	68.30	63.63	58.88	64.89	69.80*	78.63*
DAPT	LB+LC	**74.06**	**71.83**	**78.78**	**71.11**	**65.79**	**72.31**	-	-
DAPT	w/o LC	73.99	71.55	78.71	70.38	64.78	71.89	-	-
DAPT	w/o LB	73.94	70.81	77.41	69.45	62.82	70.88	-	-
DAPT+CRF	w/o LB+LC	73.07	68.99	77.53	68.82	62.63	70.21	-	-
DAPT-TAGGER†	w/o LB+LC	72.05	68.78	75.71	69.04	62.56	69.63	-	-

Table 3: The performance of baselines and our full model. LC and LB represent label context information, and label background information, respectively. † denotes the results from Liu et al. (2020b) (except values with *).

model that achieves the highest performance on the validation sets is evaluated on the test set.

4 Results

4.1 Comparison with Previous Studies

To illustrate the effectiveness of our proposed model, we compare it to previous studies and report the results in Table 2. There are several observations. First, we can observe that our model significantly outperforms all previous works, which illustrates the effectiveness of the proposed framework. Second, the comparison between our model and BARTNER-BASE confirms the validity of incorporating label information in cross-domain NER. Although both methods utilize generative approaches to perform NER, our model can grasp label-related knowledge by directly encoding NE tags with the help of a label embedding table. However, BARTNER-BASE needs to covert NE labels to original tokens, which may hurt the label information extraction and transfer. Third, our model demonstrates its superiority of simplicity when compared with those works that either incorporate external resources or introduce complicated training designs. For example, MULTI-CELL LSTM combines two auxiliary tasks, entity type prediction and attention score guidance, with the NER task, and applies multi-task learning. In contrast, our model can achieve better results with a simpler method, where we only need to train our model on the source domain and then fine-tune it to the target domain. This indicates that an appropriate design can alleviate the need for additional resources.

When DAPT is introduced to OURS and MULTI-CELL LSTM, OURS+DAPT and MULTI-CELL LSTM+DAPT further improve the performance (with 3.71% and 2.16% improvements on averaged F1-score on CrossNER dataset), which illustrates that DAPT can narrow the gap between the source and target domains. Since domain-related corpus contains abundant domain-specialized background information, it can help the model better understand the text in the target domain. Besides, both OURS and OURS+DAPT outperform MULTI-CELL LSTM+DAPT, regardless of DAPT, further demonstrating the potential of our proposed model in cross-domain NER.

4.2 Effect of Label-Aware Information

The main results are shown in Table 3. First, models incorporating label information outperform those ignoring such information (i.e., BERT and DAPT w/o LB+LC), which further confirms the validity of label information in this task. We can attribute that such information can provide valuable label-related knowledge to enhance the entity prediction. Second, on these datasets, the performance gains from our full model (i.e., BERT with LB+LC) over BERT-TAGGER on the CrossNER are larger than that of Movie and Restaurant. This observation owes to the fact that Movie and Restaurant do not share the same entity types with Conll2003, leading to a larger gap between label information from the source domain and target domain, which makes it more difficult for label features transfer. Third, our proposed framework shows its effectiveness when compared with those models that introduce the CRF layer. The reason behind this might be that our model can learn better label-related information from the source domain (including token-label and label-label relationships) and transfer it to the target domain, especially for two domains that share the same NE labels, while CRF can only recognize correlations between tags in the neighborhoods.

Moreover, we also conduct ablation studies: (1) without label context information (i.e., w/o LC), (2) without label background information (i.e., w/o LB), (3) without label context and background in-

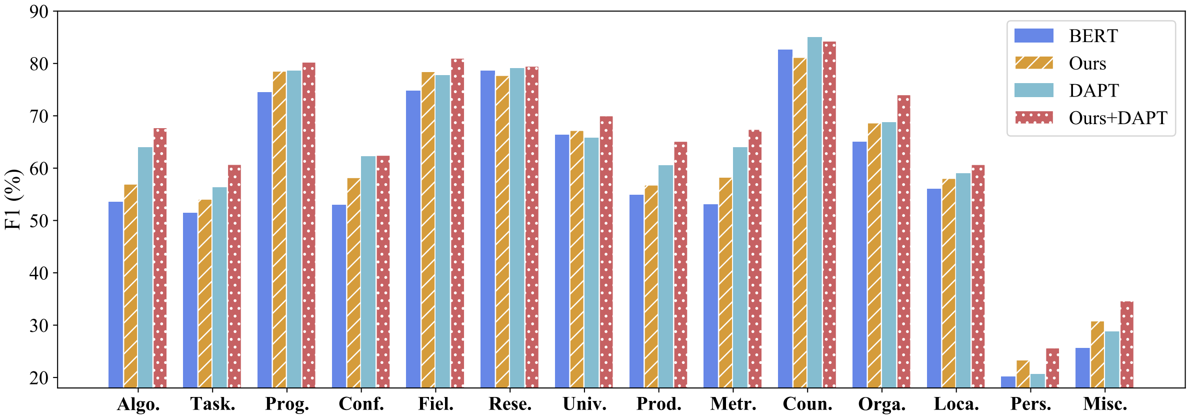

Figure 2: F1 scores of fine-grained comparisons on AI datasets. Results are averaged over three runs. The last four entity types are shared between the source and target domains.

formation (i.e., w/o LB+LC). The comparison between the base model (i.e., w/o LB+LC) and ones with LB or LC shows the effectiveness of each component in promoting cross-domain entity identification. Besides, it is observed that our full models (i.e., LB+LC) outperform w/o LB and w/o LC, which illustrates that combining label background and context information can further enhance information transfer and bring more improvements. The main reason might be that these two vectors are weighted sum over token features and label representation, respectively, with different focuses. Therefore, their combination can generate a better understanding of the label knowledge.

5 Analyses

5.1 In-domain Performance

To test the in-domain performance of our model, we utilize the single domain dataset to train our model and evaluate it on the corresponding test set, with results reported in Table 4. We can observe that our innovation in terms of incorporating label information is also productive for the in-domain NER task, where our model achieves better performance than the corresponding baselines. It can be attributed that our model can grasp a more comprehensive understanding between text and their labels and thus boost in-domain NER. However, the improvement gains from our model over baselines on the in-domain NER task are not as significant as that on cross-domain NER. An explanation for this observation may be that, in cross-domain NER, our model can better comprehend shared NE types. Therefore, it could help the model recognize differences and find more reasonable similarities between different domains, while this advantage may not be helpful for in-domain NER.

Model	Politics	Science	Music	Literature	AI
BERT†	66.56	63.73	66.59	59.95	50.37
Ours	**68.13**	**66.21**	**68.75**	**61.37**	**53.09**
DAPT†	70.45	67.59	73.39	64.96	56.36
Ours+DAPT	**71.83**	**69.23**	**74.79**	**66.35**	**58.12**

Table 4: F1 scores with respect to in-domain NER.

5.2 Fine-grained Comparison

We further investigate the fine-grained comparison on the AI dataset and visualize the results in Figure 2. We can see that our model obtains better performance in most entity types, regardless of whether DAPT is used, which indicates that the improvements gained from label-aware information are consistent across various entity classes. All shared entity types (i.e., the last four entity types) achieve increased performance by our models, indicating that our model can grasp more useful label information from the source domain and effectively transfer them to the target domain. However, for all non-shared entity types, our model leads to a slight decrease on a few entity types (e.g., **Country**). We find that **Country** is similar with **Location** and Conll2003 annotates some countries as the **Location** while CrossNER tends to label them as the **Country**. For example, *"Netherlands"* in Conll2003 is the **Location**, whereas, in the AI dataset, it is marked as the **Country**. Hence, the label information learned from the source domain may contribute to mis-classification about **Country** in the target domain. It can also explain performance drop in entity category **Researcher** since it is easily confused with **Person**.

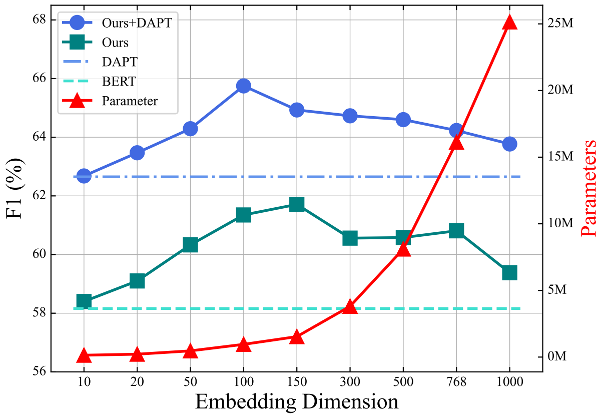

Figure 3: The F1 scores from Ours and Ours+DAPT against the label embedding dimension, where the model is tested on the AI dataset.

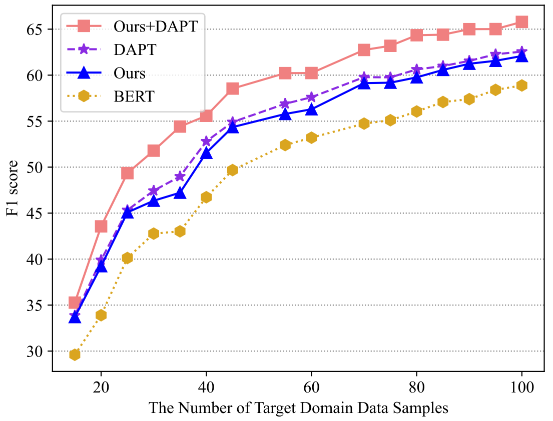

Figure 4: F1 scores of BERT, Ours, DAPT, and Ours+DAPT with different amounts of AI data in the target domain.

5.3 Effect of Label Embedding Dimension

To demonstrate the impacts of label embedding size, we train our model by varying the embedding dimension d_2 from 10 to 1000, as shown in Figure 3. It is observed that increasing the embedding size performs a better performance when the dimension is relatively small (i.e., $d_2 \leq 100$ for Ours+DAPT and $d_2 \leq 150$ for Ours). It indicates that, within this range, larger embedding can bring more valuable label information. However, when the dimension becomes too large, the performance gradually drops. It can be explained that a too large embedding matrix is difficult to be trained, resulting in redundant noise and degraded model performance. In addition, our model only introduces relatively small parameters when incorporating label-aware information. Especially when our models obtain the best results at $d_2 = 100$ and 150, their introduced extra parameters are 0.88% and 1.41% compared to the base model.

5.4 Effect of Data Size

To explore the impact of the target domain data size, we conduct experiments on different amounts of target training data (i.e., increasing from 10 to 100 samples) based on best-performing settings. The results are shown in Figure 4. With the data size increasing, all models gradually obtain better F1 scores, which illustrates that data scale plays an important role in the NER task. Besides, it is observed that both Ours and Ours+DAPT outperform corresponding baselines (i.e., BERT and DAPT) no matter how many samples we select, which further confirms the effectiveness of incorporating label information into cross-domain NER.

6 Related Work

In NLP, NER aims to identify entities from unstructured text, which has been studied widely over the past decades. Recently, neural networks have significantly improved the performance of NER, owing to their strong ability in feature extraction (Huang et al., 2015; Lample et al., 2016; Ma and Hovy, 2016; Yan et al., 2019; Devlin et al., 2019; Luo et al., 2020; Wang et al., 2021; Yamada et al., 2020; Yan et al., 2021). For example, Huang et al. (2015); Lample et al. (2016) combined Bi-LSTM with a CRF layer to enhance NER. Devlin et al. (2019); Yamada et al. (2020); Yan et al. (2019) further introduced the Transformer-based (Vaswani et al., 2017) encoders to extract more effective information from the sequence, which is then used to facilitate NER. However, although these models achieved great performance, they required large-scale labeled training data to adapt to different domains. Therefore, cross-domain NER has drawn substantial attention in recent years and gradually become one of the hot research topics in NLP. Many approaches have been proposed to enhance cross-domain NER (Pan et al., 2013; Jia et al., 2019; Jia and Zhang, 2020; Liu et al., 2020b; Chen and Moschitti, 2019). For example, Jia et al. (2019) utilized a parameter generation network to perform cross-domain and cross-task knowledge transfer and employed multi-task learning to combine NER and language modeling tasks. Furthermore, Jia and Zhang (2020) presented a multi-cell compositional LSTM structure that incorporated the entity type by a separate cell state to enhance the cross-domain NER. Compared with these studies, our model provides a simple but effective solution for addressing cross-domain NER by improving label information transfer and predicting current labels through corresponding tokens and previous labels together.

7 Conclusion

In this paper, we have proposed a novel framework for cross-domain NER to enhance the relationship between the source text and labels and improve label information transfer, where each NE label is jointly predicted by corresponding token and previous NE labels. We not only adopt a commonly-used pre-trained model to extract token representation, but also introduce a random initialized embedding matrix and Bi-LSTM-based label encoder to model the label sequence generated from previous steps. After that, we construct two different attention between hidden states of Bi-LSTM and token representations to produce label background and context information, which are then concatenated as label-aware information and applied to predict labels. Thanks to this design, the label information can be effectively transferred from the source to the target domain. Comprehensive experimental results on several benchmark datasets illustrate the effectiveness of our model, which achieves significant improvements over existing methods.

Acknowledgements

This work is supported by Chinese Key-Area Research and Development Program of Guangdong Province (2020B0101350001), NSFC under the project "The Essential Algorithms and Technologies for Standardized Analytics of Clinical Texts" (12026610) and the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649.

Lingzhen Chen and Alessandro Moschitti. 2019. Transfer Learning for Sequence Labeling Using Source Model and Target Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6260–6267.

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.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Sepp Hochreiter, J urgen Schmidhuber, and Corso Elvezia. 1997. Long Short-term Memory. *Neural Computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv preprint arXiv:1508.01991*.

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.

Chen Jia and Yue Zhang. 2020. Multi-Cell Compositional LSTM for NER Domain Adaptation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5906–5917.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

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.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. 2013a. Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390. IEEE.

Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and Jim Glass. 2013b. Query understanding enhanced by hierarchical parsing structures. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 72–77. IEEE.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020a. Coach: A Coarse-to-Fine Approach for Cross-Domain Slot Filling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 19–25.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2020b. CrossNER: Evaluating Cross-Domain Named Entity Recognition.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical Contextualized Representation for Named Entity Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8441–8448.

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 (Volume 1: Long Papers)*, pages 1064–1074.

Sinno Jialin Pan, Zhiqiang Toh, and Jian Su. 2013. Transfer Joint Embedding for Cross-domain Named Entity Recognition. *ACM Transactions on Information Systems (TOIS)*, 31(2):1–27.

Erik 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.

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.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning. *arXiv preprint arXiv:2105.03654*.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: Adapting Transformer Encoder for Named Entity Recognition. *arXiv preprint arXiv:1911.04474*.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A Unified Generative Framework for Various NER Subtasks. *arXiv preprint arXiv:2106.01223*.

MODEL	HY.	DATA		
		CROSSNER	MOVIE	RESTAURANT
OURS	BS	16	16	16
	LR	5e-5	5e-5	1e-5
	ME	100	100	100

Table 5: The best hyper-parameters that we used in our experiments. BS, LR, and ME represent the batch size, learning rate, and max epochs, respectively.

A Appendix

A.1 Hyper-parameter Settings

We have tested several combinations of hyper-parameters in tuning our models on CrossNER, Movie and Restaurant. Table 5 reports the combinations that achieve the highest F-1 score for each dataset.