TransAdv: A Translation-based Adversarial Learning Framework for Zero-Resource Cross-Lingual Named Entity Recognition

Yichun Zhao[1], **Jintao Du**[2], **Gongshen Liu**[1]*, **Huijia Zhu**[2]
[1]Shanghai Jiao Tong University
[2] Tiansuan Lab, Ant Group Co., Ltd.
[1]{zhaoyichun, lgshen}@sjtu.edu.cn
[2]{lingke.djt, huijia.zhj}@antgroup.com

Abstract

Zero-Resource Cross-Lingual Named Entity Recognition aims at training an NER model of the target language using only labeled source language data and unlabeled target language data. Existing methods are mainly divided into three categories: model transfer based, data transfer based and knowledge transfer based. Each method has its own disadvantages, and combining more than one of them often leads to better performance. However, the performance of data transfer based methods is often limited by inevitable noise in the translation process. To handle the problem, we propose a framework named TransAdv to mitigate lexical and syntactic errors of word-by-word translated data, better utilizing the data by multi-level adversarial learning and multi-model knowledge distillation. Extensive experiments are conducted over 6 target languages with English as the source language, and the results show that TransAdv achieves competitive performance to the state-of-the-art models.

1 Introduction

Named Entity Recognition (NER) is a fundamental task that aims to locate named entities in a given sentence and assign them to predefined types, i.e., person, location, organization, etc. In recent years, neural NER models have achieved remarkable performance on this task with a large amount of labeled data. However, many low-resource languages do not have enough data for supervised learning. Therefore, transferring labeled data or trained models from high-resource to low-resource languages is gaining increasing attention.

In this paper, we concentrate on zero-resource cross-lingual NER where no labeled data in the target language is available. Existing methods fall into three main categories: i) model transfer based methods (Wu and Dredze, 2019; Wu et al., 2020c), which train a source model on the labeled source language data to learn language-independent features and then directly apply it to the target language; ii) data transfer based methods (Mayhew et al., 2017; Xie et al., 2018), which translate the labeled source language data and map all entity labels to generate pseudo target language data; iii) knowledge transfer based methods (Wu et al., 2020a; Chen et al., 2021), which train a source model on the labeled source language data and then apply it over the unlabeled target language data to distill a student model.

Each kind of method has its drawbacks, (Wu et al., 2020b) is the first to unify the three kinds of methods with great success. However, the noise in the translation process significantly limits its performance. There are two common translation strategies for cross-lingual NER tasks: i) sentence translation followed by entity alignment, where the propagation of entity alignment errors is inevitable; ii) directly word-by-word translation (Wu et al., 2020b), where the generated sentence is noisy in terms of word order.

To better utilize the translated data, we propose a translation-based adversarial learning framework named TransAdv for zero-resource cross-lingual NER, and the overview architecture is shown in Figure 1. The contributions of our work can be summarized as follows:

- We better unify data transfer and knowledge transfer for cross-lingual NER, mitigating lexical and syntactic errors of word-by-word translated data through multi-level adversarial learning and multi-model knowledge distillation.

- We conduct extensive experiments over 6 target languages with English as the source language, and the results validate the effectiveness and reasonableness of our model.

[*]Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 742–749
December 7-11, 2022 ©2022 Association for Computational Linguistics

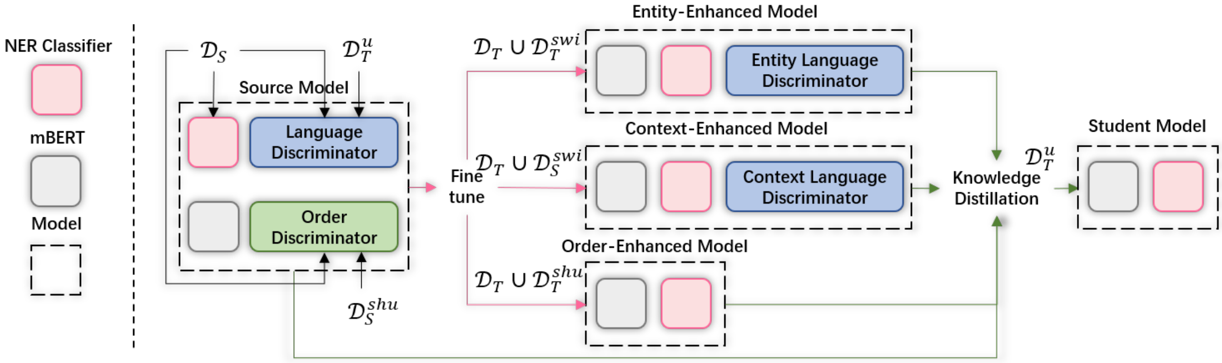


Figure 1: The overall architecture of our proposed TransAdv.

2 Methodology

2.1 Task Definition

Following previous works, we formulate cross-lingual NER as a standard sequence labeling problem. When given an input sentence $\boldsymbol{x} = \{x_1, x_2, \cdots, x_n\}$ of length n, a model aims to extract all named entities appearing in the sentence by generating a sequence of labels $\boldsymbol{y} = \{y_1, y_2, \cdots, y_n\}$ over the set of entity labels $\mathcal{Y}$. We denote the labeled source language data as $\mathcal{D}_S = \{(\boldsymbol{x}, \boldsymbol{y})\}$ and the unlabeled target language data as $\mathcal{D}_T^u = \{\tilde{\boldsymbol{x}}\}$. In the case of zero-resource cross-lingual NER, a model is trained with $\mathcal{D}_S$ and $\mathcal{D}_T^u$, then evaluated on the labeled test data of the target language.

2.2 Data Creation

In this section, we construct multiple datasets based on the labeled source language data as shown in Figure 2.

Following (Wu et al., 2020b), we apply MUSE (Lample et al., 2018) to translate a source language sentence x_S into a target language sentence x_T word-by-word. The entity label of each source language word is then directly copied to its corresponding translated word. Since MUSE has inevitable translation errors that it may not strictly translate each word into the target language, we also try Google translate API for more accurate word-by-word translation. After word-by-word translating and copying entity labels, we construct a pseudo target language training dataset $\mathcal{D}_T$ from $\mathcal{D}_S$.

(Zhang et al., 2021a) propose an aspect code-switching mechanism to augment the training data for cross-lingual aspect-based sentiment analysis. In this section, we apply a similar mechanism to switch named entities between the source and translated sentences to construct two bilingual sentences: x_S^{swi} is derived from x_S with named entities in x_T, and x_T^{swi} is derived from x_T with named entities in x_S. The entity label of each word in x_S^{swi} and x_T^{swi} is also the same as its corresponding word in x_S and x_T benefiting from word-by-word translation. Therefore, we can construct two bilingual datasets $\mathcal{D}_S^{swi}$ and $\mathcal{D}_T^{swi}$.

Due to the difference between the word orders of the source language and the target language, we also design a word shuffling method for NER data. Since NER is a coarse-grained sequence labeling task, completely shuffling all words in a sentence will affect the internal relations of the words within entities. Therefore, we separately shuffle the words in each entity or each context between entities, with all entity labels retained. For sentences x_S and x_T, two shuffled sentences are denoted as x_S^{shu} and x_T^{shu}. Based on $\mathcal{D}_S$ and $\mathcal{D}_T$, we can build two shuffled datasets $\mathcal{D}_S^{shu}$ and $\mathcal{D}_T^{shu}$.

2.3 Multi-Level Adversarial Learning for Cross-Lingual NER

In cross-lingual tasks, the source and the target language usually have differences in lexical and syntactic features. To avoid the model overfitting on the source language data and make the model better fine-tuned on the word-by-word translated target language data, we follow (Chen et al., 2021) and propose a multi-level adversarial network. It is formulated as a multi-task problem with NER, word-level language classification and sentence-level order classification. Different modules in the network and their loss functions are defined as below:

Generator We choose multilingual BERT (Devlin et al., 2019) as the generator and feed a given sen-

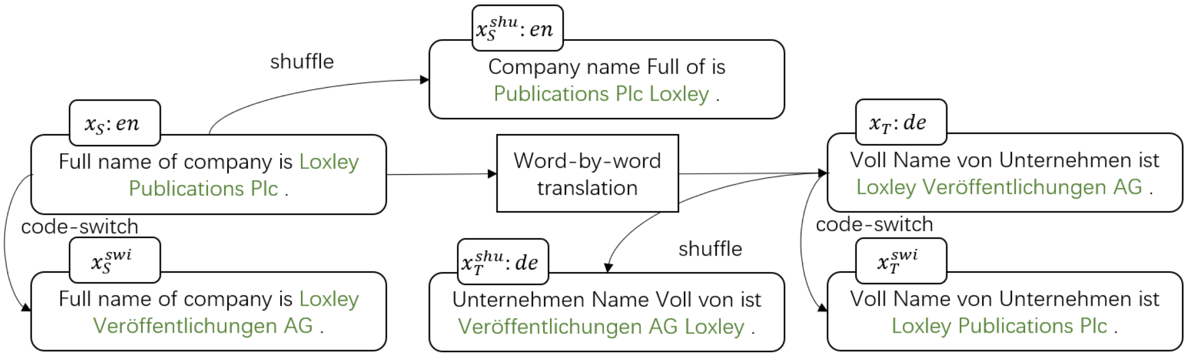

x_S^{shu}: en
shuffle
Company name Full of is Publications Plc Loxley .
x_S: en
Full name of company is Loxley Publications Plc .
Word-by-word translation
x_T: de
Voll Name von Unternehmen ist Loxley Veröffentlichungen AG .
code-switch
x_S^{swi}
Full name of company is Loxley Veröffentlichungen AG .
x_T^{shu}: de
shuffle
Unternehmen Name Voll von ist Veröffentlichungen AG Loxley .
code-switch
x_T^{swi}
Voll Name von Unternehmen ist Loxley Publications Plc .

Figure 2: The process of data creation.

tence $\boldsymbol{x} = \{x_1, x_2, \cdots, x_n\}$ into mBERT to generate the feature vectors $\boldsymbol{h} = \{h_1, h_2, \cdots, h_n\}$:

$$\boldsymbol{h} = \text{mBERT}(\boldsymbol{x}) \tag{1}$$

NER Classifier We feed $\boldsymbol{h}$ into a fully-connected layer followed by a softmax layer to yield a probability distribution over the entity label set $\mathcal{Y}$:

$$p_i^{ner} = \text{softmax}(\boldsymbol{W}^{ner} h_i + \boldsymbol{b}^{ner}) \tag{2}$$

where $h_i \in \mathbb{R}^{d_g}$ denotes the feature vector of the ith word with d_g being the dimension of $\boldsymbol{h}$, $p_i^{ner} \in \mathbb{R}^{|\mathcal{Y}|}$, $W^{ner} \in \mathbb{R}^{|\mathcal{Y}| \times d_g}$ and $b^{ner} \in \mathbb{R}^{|\mathcal{Y}|}$.

Language Discriminator We feed $\boldsymbol{h}$ into two fully-connected layers followed by a sigmoid layer to classify the language of each word:

$$p_i^l = \text{sigmoid}(\boldsymbol{W}_1^l ReLU(\boldsymbol{W}_2^l h_i)) \tag{3}$$

where $p_i^l \in \mathbb{R}^1$, $W_1^l \in \mathbb{R}^{1 \times d_l}$ and $W_2^l \in \mathbb{R}^{d_l \times d_g}$ with d_l being the hidden dimension of the language discriminator.

Order Discriminator We first feed $\boldsymbol{h}$ into a one-layer LSTM to encode sequence features of the sentence, then the hidden state of the last word is fed into a fully-connected layer followed by a sigmoid layer to classify the order of the sentence:

$$\begin{aligned} h_i' &= \overrightarrow{LSTM}(h_i, h_{i-1}') \\ p^o &= \text{sigmoid}(\boldsymbol{W}^o h_n') \end{aligned} \tag{4}$$

where $p^o \in \mathbb{R}^1$, $W^o \in \mathbb{R}^{1 \times d_o}$ with d_o being the hidden dimension of the order discriminator.

During training, different datasets will first be fed into mBERT separately as shown in Figure 1, then the generated $\boldsymbol{h}$ will be sent to the corresponding module. We have a total of 4 loss functions: the NER task loss $\mathcal{L}^{ner}$, the language discriminator loss $\mathcal{L}^l$, the order discriminator loss $\mathcal{L}^o$, and the generator loss $\mathcal{L}^g$:

$$\begin{aligned} \mathcal{L}^{ner} &= -\sum_{i=1}^{n} \sum_{k \in \mathcal{Y}} I(y_i^{ner} = k) log(p_i^{ner}) \\ \mathcal{L}^l &= -\sum_{i=1}^{n} [y_i^l log(p_i^l) + (1 - y_i^l) log(1 - p_i^l)] \\ \mathcal{L}^o &= -[y^o log(p^o) + (1 - y^o) log(1 - p^o)] \\ \mathcal{L}^g &= -\sum_{i=1}^{n} [y_i^l log(1 - p_i^l) + (1 - y_i^l) log(p_i^l)] \\ &- [y^o log(1 - p^o) + (1 - y^o) log(p^o)] \end{aligned} \tag{5}$$

where y_i^{ner} and y_i^l denote the ground truth entity tag and language tag of the word x_i, y^o denotes the ground truth order tag of the sentence $\boldsymbol{x}$.

Similarly to (Chen et al., 2021), for the NER task, the parameters of the generator and the NER classifier are updated based on $\mathcal{L}^{ner}$; for the adversarial task, the parameters of two discriminators are updated based on $\mathcal{L}^l$ and $\mathcal{L}^o$ respectively, while the parameters of the generator are updated based on $\mathcal{L}^g$. Finally, we denote the trained source model as Θ_{src}.

2.4 Multi-Model Knowledge Distillation on Unlabeled Data

Based on Θ_{src}, we further fine-tune it on different datasets to derive teacher models with different emphases. Actually, three combinations of the datasets constructed in section 2.2 are considered in our network, including $\mathcal{D}_{entity} = \mathcal{D}_T \cup \mathcal{D}_T^{swi}$, $\mathcal{D}_{context} = \mathcal{D}_T \cup \mathcal{D}_S^{swi}$ and $\mathcal{D}_{order} = \mathcal{D}_T \cup \mathcal{D}_T^{shu}$.

$\mathcal{D}_{entity}$ contains the word-by-word translated dataset $\mathcal{D}_T$ to involve knowledge of the target language and the code-switch dataset $\mathcal{D}_T^{swi}$ to share the same contexts but entities with different languages. $\mathcal{D}_{context}$ contains $\mathcal{D}_T$ and the code-switch dataset $\mathcal{D}_S^{swi}$ to share the same entities but contexts with different languages. $\mathcal{D}_{order}$ contains $\mathcal{D}_T$ and the shuffled dataset $\mathcal{D}_T^{shu}$ to share the same sentences but different orders of words. An entity-enhanced teacher model Θ_{entity}, a context-enhanced teacher model $\Theta_{context}$ and an order-enhanced teacher model Θ_{order} are derived by fine-tuning Θ_{src} on $\mathcal{D}_{entity}$, $\mathcal{D}_{context}$ and $\mathcal{D}_{order}$ with the same loss function $\mathcal{L}^{ner}$ as in Eq. 5.

During fine-tuning, the language discriminator trained in section 2.3 is also loaded to continue adversarial fine-tuning with Θ_{entity} and $\Theta_{context}$ while the adversarial strategy is more fine-grained. For Θ_{entity} we only discriminate languages of entity words and for $\Theta_{context}$ we only discriminate languages of context words. The new language discriminator losses are shown as in Eq. 6. These two discriminators are adapted to characteristics of $\mathcal{D}_{entity}$ and $\mathcal{D}_{context}$ with the aim of enabling Θ_{entity} and $\Theta_{context}$ to better fuse representations of entity or context words respectively of the source and the target languages.

$$\mathcal{L}^{el} = -\sum_{x_i \in entity} [y_i^l log(p_i^l) + (1-y_i^l) log(1-p_i^l)]$$
$$\mathcal{L}^{cl} = -\sum_{x_i \in context} [y_i^l log(p_i^l) + (1-y_i^l) log(1-p_i^l)] \tag{6}$$

We then implement a multi-model distillation on the unlabeled target language dataset $\mathcal{D}_T^u$. Let $\tilde{x}_i$ denotes the ith word in an unlabeled sentence $\tilde{\boldsymbol{x}} \in \mathcal{D}_T^u$ and $p^{ner}(\tilde{x}_i, \Theta)$ denotes the probability distribution predicted by model Θ. We combine soft labels generated by Θ_{src} and three enhanced teacher models to obtain the united soft label:

$$p^{uni}(\tilde{x}_i) = w_1 p^{ner}(\tilde{x}_i, \Theta_{src}) + w_2 p^{ner}(\tilde{x}_i, \Theta_{entity}) + w_3 p^{ner}(\tilde{x}_i, \Theta_{context}) + w_4 p^{ner}(\tilde{x}_i, \Theta_{order}) \tag{7}$$

where w_k is the weight for each model.

Finally, we distill a student model Θ_{stu} by minimizing the mean squared error (MSE) between $p^{uni}(\tilde{x}_i)$ and the probability distribution predicted by Θ_{stu}:

$$\mathcal{L}^{kd} = \frac{1}{n}\sum_{i=1}^{n} MSE(p^{uni}(\tilde{x}_i), p^{ner}(\tilde{x}_i, \Theta_{stu})) \tag{8}$$

For inference on the labeled test data of the target language, we only employ the distilled student model Θ_{stu}.

3 Experiments

3.1 Baselines

We compare our model with the following zero-resource cross-lingual NER models to evaluate the performance of TransAdv: **mBERT-FT** (Wu and Dredze, 2019) fine-tune the multilingual BERT. **AdvCE** (Keung et al., 2019) improve upon mBERT's performance via adversarial learning. **TSL** (Wu et al., 2020a) propose a teacher-student learning method. **Unitrans** (Wu et al., 2020b) propose an approach to unify both model and data transfer. **RIKD** (Liang et al., 2021) propose a reinforced knowledge distillation framework. **AdvPicker** (Chen et al., 2021) attempt to select language-independent data by adversarial learning. **TOF** (Zhang et al., 2021b) design a target-oriented fine-tuning framework to exploit various data.

3.2 Datasets and Metrics

(a) Statistics of CoNLL.

Language	Type	Train	Dev	Test
English-en (CoNLL-2003)	Sentence Entity	14987 23499	3466 5942	3684 5648
German-de (CoNLL-2003)	Sentence Entity	12705 11851	3068 4833	3160 3673
Spanish-es (CoNLL-2002)	Sentence Entity	8323 18798	1915 4351	1517 3558
Dutch-nl (CoNLL-2002)	Sentence Entity	15806 13344	2895 2616	5195 3941

(b) Statistics of WikiAnn.

Language	Type	Train	Dev	Test
English-en	Sentence Entity	20000 27931	10000 14146	10000 13958
Arabic-ar	Sentence Entity	20000 22500	10000 11266	10000 11259
Hindi-hi	Sentence Entity	5000 6124	1000 1226	1000 1228
Chinese-zh	Sentence Entity	20000 25031	10000 12493	10000 12532

Table 1: Statistics of the datasets.

We conducted experiments on the following NER benchmark datasets: CoNLL-2002 (Sang and Erik, 2002) for Spanish[es] and Dutch[nl], CoNLL-2003 (Sang and De Meulder, 2003) for English[en] and German[de], and WikiAnn (Pan et al., 2017) for English[en], Arabic[ar], Hindi[hi] and Chinese[zh]. Each dataset of a certain language is split into train, dev and test sets and statistics of all datasets are shown in Table 1. All datasets are annotated with 4 entity types: LOC, MISC, ORG and PER, using the BIO entity labeling scheme.

Following previous work (Sang and Erik, 2002), we employ entity level F1 score as the metric of evaluation. We run each experiment 5 times with different random seeds and report the average F1 score on the test set for reproducibility. More details of implementation details and model analysis are in appendix A and B.

3.3 Main Results

(a) Results on CoNLL.

Model	es	nl	de	Average
mBERT-FT	74.96	77.57	69.56	73.57
AdvCE	74.3	77.6	71.9	74.60
TSL	76.94	80.89	73.22	77.02
RIKD	77.84	82.46	75.48	78.59
AdvPicker	79.00	82.90	75.01	78.97
Unitrans	79.31	82.90	74.82	79.01
TOF	80.35	82.79	**76.57**	79.90
TransAdv	**80.93**	**83.78**	75.52	**80.08**

(b) Results on WikiAnn.

Model	ar	hi	zh	Average
mBERT-FT	42.30	67.60	52.90	54.27
TSL	43.12	69.54	48.12	53.59
RIKD	**45.96**	70.28	50.40	55.55
TransAdv	42.53	**74.24**	**54.25**	**57.01**

Table 2: Results of TransAdv and baselines. (F1%). All results are from original papers or the paper of RIKD.

The main results of baselines and TransAdv on CoNLL and WikiAnn are shown in Table 2. According to the results, TransAdv outperforms all baselines, proving our model's effectiveness.

In general, due to the strong effect of knowledge distillation, knowledge transfer based methods such as TSL, RIKD, AdvPicker and our TransAdv significantly surpass model transfer based methods like mBERT-FT and AdvCE which directly apply the model to the target language.

For western languages in CoNLL, TransAdv achieves 1.62%, 0.88% and 0.7% absolute gain of F1 scores over Unitrans which also employs word-by-word translation. Despite using the same translation resources, our model still has a significant improvement over it, which may due to the adversarial network mitigating the lexical and syntactic errors of the translated data. Compared with TOF, which is the state-of-the-art model, TransAdv achieves an absolute F1 scores increase of 0.58%, 0.99% on es, nl and decrease of 1.05% on de. TOF requires extra labeled data of Machine Reading Comprehension (MRC) for both the source language and target language, which is costly and not in a strictly zero-resource case. For many low-resource languages, word-by-word translation is much more available than labeled MRC data.

As for non-western languages in WikiAnn, TransAdv also shows significant improvements over the baselines on hi, zh. We even achieve 0.68% absolute F1 scores gain on zh over mBERT-FT, which re-tokenizes the Chinese dataset and obtains relatively high results.

4 Conclusion

In this paper, we propose a framework named TranAdv for zero-resource cross-lingual NER, which mitigates lexical and syntactic errors of word-by-word translated data and better utilizes it through multi-level adversarial learning and multi-model knowledge distillation. We evaluate TransAdv over 6 target languages with English as the source language. Experimental results show that TransAdv achieves competitive performance to the state-of-the-art models.

5 Limitations

Although word-by-word translation data is easy to obtain in most cases, high-quality translation models are not available for some low-resource languages that are extremely short of parallel corpora. Moreover, when the difference in word order between the source and target languages is slight, adversarial training of word order may result in the loss of valid order information.

Acknowledge

This research work has been sponsored by Ant Group Security and Risk Management Fund, the Joint Funds of the National Natural Science Foundation of China (Grant No. U21B2020), and Shanghai Science and Technology Plan (Grant No. 22511104400).

References

Weile Chen, Huiqiang Jiang, Qianhui Wu, Börje Karlsson, and Yi Guan. 2021. Advpicker: Effectively leveraging unlabeled data via adversarial discriminator for cross-lingual ner. In *Proc. of ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Phillip Keung, Yichao Lu, and Vikas Bhardwaj. 2019. Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and ner. In *Proc. of EMNLP*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proc. of ICLR*.

Shining Liang, Ming Gong, Jian Pei, Linjun Shou, Wanli Zuo, Xianglin Zuo, and Daxin Jiang. 2021. Reinforced iterative knowledge distillation for cross-lingual named entity recognition. In *Proc. of KDD*.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*.

Stephen Mayhew, Chen-Tse Tsai, and Dan Roth. 2017. Cheap translation for cross-lingual named entity recognition. In *Proc. of EMNLP*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proc. of ACL*.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proc. of AACL*.

Tjong Kim Sang and F Erik. 2002. Introduction to the conll-2002 shared task: language-independent named entity recognition. In *Proceedings of CoNLL-2002/Roth, Dan edit*.

Qianhui Wu, Zijia Lin, Börje Karlsson, Jian-Guang Lou, and Biqing Huang. 2020a. Single-/multi-source cross-lingual ner via teacher-student learning on unlabeled data in target language. In *Proc. of ACL*.

Qianhui Wu, Zijia Lin, Börje F Karlsson, Biqing Huang, and Jianguang Lou. 2020b. Unitrans: Unifying model transfer and data transfer for cross-lingual named entity recognition with unlabeled data. In *Proc. of IJCAI*.

Qianhui Wu, Zijia Lin, Guoxin Wang, Hui Chen, Börje F Karlsson, Biqing Huang, and Chin-Yew Lin. 2020c. Enhanced meta-learning for cross-lingual named entity recognition with minimal resources. In *Proc. of AAAI*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. In *Proc. of EMNLP*.

Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A Smith, and Jaime G Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. In *Proc. of EMNLP*.

Wenxuan Zhang, Ruidan He, Haiyun Peng, Lidong Bing, and Wai Lam. 2021a. Cross-lingual aspect-based sentiment analysis with aspect term code-switching. In *Proc. of EMNLP*.

Ying Zhang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021b. Target-oriented fine-tuning for zero-resource named entity recognition. In *Proc. of ACL Findings*.

A Implementation Details

We implement TransAdv with Pytorch 1.10.2[1]. For word-by-word translation, we employ MUSE[2] (Lample et al., 2018) based on fastText[3] monolingual word embeddings for western languages of es, nl, de and Google Translate API[4] for non-western languages of ar, hi, zh. For pretrained model, the cased multilingual BERT (Devlin et al., 2019) with 12 stacked Transformer blocks, 12 attention heads and 768 hidden dimensions is implemented based on HuggingFace's Transformers[5].

We empirically set the hyper-parameters of TransAdv and employ them in all experiments. Following previous work (Wu et al., 2020c), we freeze the parameters of the embedding layer and the bottom three layers of the multilingual BERT used in our model. We train all models for 10 epochs with the batch size of 32, the maximum sequence length of 128 and the dropout rate of 0.1, saving the model with the best performance on the dev set of the target language. We use AdamW (Loshchilov and Hutter, 2018) as the optimizer with the weight decay of 0.01 and the warmup rate of 0.05. For sequence prediction, we apply Viterbi decoding to generate the predicted results. In multi-level adversarial learning, the learning rate is set to 6e-5 for $\mathcal{L}^{ner}$, 6e-7 for $\mathcal{L}^{g}$ and 5e-3 for $\mathcal{L}^{l}$, $\mathcal{L}^{o}$. The hidden dimensions of the language discriminator and order discriminator are set to 500 and 256 respectively. In multi-model knowledge distillation, hyper-parameters of corresponding modules

[1] https://pytorch.org/

[2] https://github.com/facebookresearch/MUSE

[3] https://fasttext.cc/docs/en/pretrained-vectors.html

[4] https://translate.google.com/

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

in three enhanced teacher models are the same as in the source model, and the student model is trained with the learning rate of 6e-5 for $\mathcal{L}^{kd}$. The weights of four models in Eq. 7 are all set to 1/4.

All experiments are conducted on a Nvidia RTX 3090 GPU (24GB). Θ_{src} trains in $\approx$ 60min with 179.17M parameters, three enhanced teacher models train in $\approx$ 30min, 30min, 16min with 178.25M, 178.25M, 177.86M parameters respectively, and Θ_{stu} trains in $\approx$ 23min with 177.86M parameters.

B Model Analysis

B.1 Ablation Study

To verify the validity of different modules in the proposed model, we introduce the following variants of TransAdv to further carry out an ablation study: 1) TransAdv w/o LDIS and TransAdv w/o ODIS, which remove the language discriminator or the order discriminator respectively during multi-level adversarial learning. Moreover, when the language discriminator is removed, the entity language discriminator and the context language discriminator during the adversarial learning of Θ_{entity} and $\Theta_{context}$ are also removed. 2) TransAdv w/o Θ_{entity}, TransAdv w/o $\Theta_{context}$ and TransAdv w/o Θ_{order}, which remove the corresponding teacher model during multi-model knowledge distillation. 3) TransAdv w/o MLADV, which removes the multi-level adversarial learning module with the Θ_{src} directly trained on $\mathcal{D}_S$ 4) TransAdv w/o MMKD, which removes the multi-model knowledge distillation module and then the student model is directly distilled with Θ_{src}.

Methods	es	nl	de	Average
TransAdv	**80.93**	**83.78**	**75.52**	**80.08**
w/o LDIS	80.70	83.43	75.19	79.77 (0.31↓)
w/o ODIS	80.73	83.38	75.31	79.81 (0.27↓)
w/o Θ_{entity}	80.31	83.59	74.91	79.60 (0.48↓)
w/o $\Theta_{context}$	80.16	83.26	75.20	79.54 (0.54↓)
w/o Θ_{order}	80.38	83.10	75.26	79.58 (0.50↓)
w/o MLADV	80.18	83.31	75.16	79.55 (0.53↓)
w/o MMKD	77.34	81.60	74.14	77.69 (2.39↓)

Table 3: Results of ablation study for TransAdv (F1%).

The performance of each variant compared to TransAdv is shown in Table 3. From the results, we can draw out the following inferences:

1) Comparing TransAdv with TransAdv w/o LDIS and TransAdv w/o ODIS, we can see the performance drops. It confirms the effectiveness of the two discriminators that they may avoid the model overfitting on the source language and make the model better fine-tuned on the word-by-word translated target language data.

2) We observe that the performance of TransAdv outperforms the performance of TransAdv w/o Θ_{entity}, TransAdv w/o $\Theta_{context}$ and TransAdv w/o Θ_{order}, showing that teacher models derived from different combinations of datasets may have different emphases on improving the robustness of the entire model.

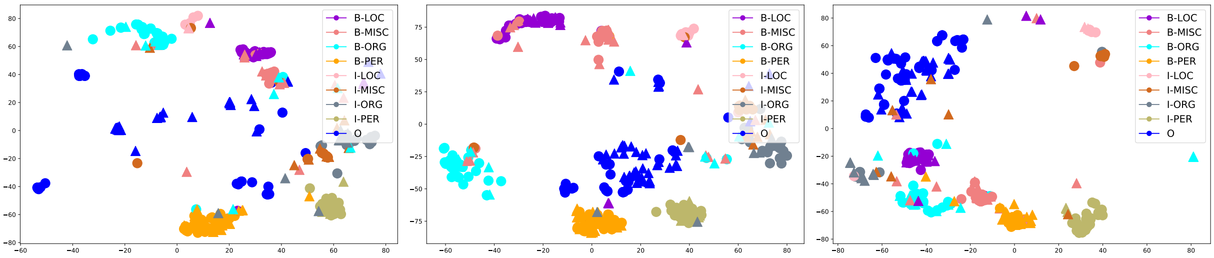
3) TransAdv w/o MLADV and TransAdv w/o MMKD both significantly decline in performance compared with TransAdv, which illustrates that the two main modules both play essential roles in TransAdv.

B.2 Analysis of Translation Strategies

To evaluate the impact of different translation strategies on TransAdv, we introduce the following translation methods: 1) MUSE: use the same word-by-word translation as (Wu et al., 2020b) based on fastText monolingual word embeddings. 2) Google Word: use Google translate API to translate the sentence word-by-word. 3) Google Phrase: split a sentence into phrases based on entity labels and then use Google translate API to translate the sentence phrase-by-phrase. 4) Google Word&Phrase: split a sentence into phrases based on entity labels and then use Google translate API to translate the sentence word-by-word for context phrases and phrase-by-phrase for entity phrases.

The comparison of different translation strategies for each language is shown in Figure 4. We observe that for western languages in CoNLL, models with MUSE obtain the best F1 score on es, nl and the second best F1 score on de; for non-western languages in WikiAnn, models with Google Word obtain the best F1 score on ar, hi and the second best F1 score on zh. It may be because when English is the source language, for western languages there are many word anchors that can be shared, and using noisier MUSE can obtain more diverse translation data without affecting the performance; whereas for non-western languages, there are much less word anchors that can be shared, so Google-based direct translation can better introduce information of the target language.

On the other hand, Google Phrase and Google Word&Phrase are generally less effective than the



(a) Source Model (b) Entity-Enhanced Teacher model (c) Context-Enhanced Teacher model

Figure 3: Clusters of embeddings of models in different stages (Circles correspond to words in the source language, triangles correspond to words in the target language).

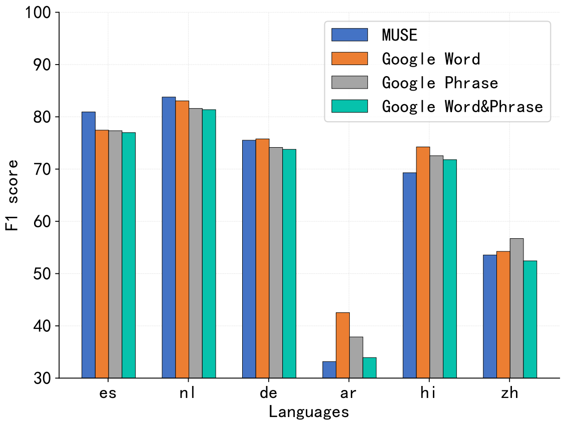


Figure 4: The comparison of different translation strategies for each language.

other two strategies which are based entirely on word-by-word translation. This may be due to the fact that the word-by-word translated data is more compatible with the sentence-level order adversarial training in TransAdv.

B.3 Analysis of Language discriminators

To analysis the effect of language discriminators of different grains, clusters of embeddings of models in different stages trained on CoNLL with Dutch (nl) as the target language are shown in Figure 3.

We find that in the source model Θ_{src}, embeddings corresponding to entity labels of the source and the target languages have been partially fused due to the original language discriminator while embeddings corresponding to context labels are too scattered. Moreover, in the entity-enhanced teacher model Θ_{entity}, embeddings of different languages get further fused thanks to word-by-word translated data and the entity language discriminator while embeddings corresponding to context labels are still relatively scattered. In the context-enhanced teacher model $\Theta_{context}$, due to the context language discriminator, the integration of embeddings corresponding to context labels is basically complete while embeddings corresponding to entity labels are not. The above results together demonstrate the effectiveness of different language discriminators.