Multi-View Cross-Lingual Structured Prediction with Minimum Supervision

Zechuan Hu$^{\diamond\ddagger}$, **Yong Jiang**†*, **Nguyen Bach**†, **Tao Wang**†, **Zhongqiang Huang**†,
Fei Huang†, **Kewei Tu**$^{\diamond*}$
$^{\diamond}$School of Information Science and Technology, ShanghaiTech University
$^{\diamond}$Shanghai Engineering Research Center of Intelligent Vision and Imaging
$^{\diamond}$Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences
$^{\diamond}$University of Chinese Academy of Sciences
†DAMO Academy, Alibaba Group
{huzch,tukw}@shanghaitech.edu.cn, yongjiang.jy@alibaba-inc.com

Abstract

In structured prediction problems, cross-lingual transfer learning is an efficient way to train quality models for low-resource languages, and further improvement can be obtained by learning from multiple source languages. However, not all source models are created equal and some may hurt performance on the target language. Previous work has explored the similarity between source and target sentences as an approximate measure of strength for different source models. In this paper, we propose a multi-view framework, by leveraging a small number of labeled target sentences, to effectively combine multiple source models into an aggregated source view at different granularity levels (language, sentence, or sub-structure), and transfer it to a target view based on a task-specific model. By encouraging the two views to interact with each other, our framework can dynamically adjust the confidence level of each source model and improve the performance of both views during training. Experiments for three structured prediction tasks on sixteen data sets show that our framework achieves significant improvement over all existing approaches, including these with access to additional source language data.

1 Introduction

Structured prediction is the task of mapping input sentences to structured outputs. It is a fundamental task in natural language processing and has many applications, i.e., sequence labeling (DeRose, 1988; Lample et al., 2016; Ma and Hovy, 2016; Hu et al., 2020b), dependency parsing (Chen and Manning, 2014; Dozat and Manning, 2016; Ahmad et al., 2019) and semantic role labeling (van der Plas et al., 2011; Strubell et al., 2018; Cai and Lapata, 2020). To achieve strong performance, structured prediction models mostly require manually labeled data that are costly to obtain in general.

Cross-lingual transfer learning (Yarowsky and Ngai, 2001; Wang and Manning, 2014; Guo et al., 2018; Lin et al., 2019; Hu et al., 2021) recently attracted attention for tackling that problem, by transferring the knowledge from high-resource languages to low-resource ones. Existing works can be categorized into two types: single-source transfer and multi-source transfer. The former is limited to transferring knowledge from one source language and generally results in inferior performance than the latter (McDonald et al., 2011; Rahimi et al., 2019), especially when the target language is similar to multiple source language over various characteristics, i.e., domain, word order, capitalization, and script style. However, in practice, we are more likely to encounter the situation where some source languages are not as similar to the target language and may lead to worse performance (Rosenstein et al., 2005; Rahimi et al., 2019) (we provide an example in the Appendix A). To tackle this challenging problem, most of the previous works do majority voting (Plank and Agić, 2018) and truth inference on hard predictions of multiple sources (Rahimi et al., 2019). To better incorporate target language information, some recent works train a new model on the target unlabeled data with hard/soft predictions from multiple source models, such as mixture-of-experts model (Chen et al., 2019) and knowledge distillation (KD) (Wu et al., 2020), and assign weights to multiple sources based on language similarity. However, these similarity-based approaches are heuristic-based, and cannot well learn the confidence level of multiple source models.

In this paper, we propose to leverage a small number of labeled target data to selectively transfer the knowledge from multiple source models.

[*] Corresponding authors. [‡]Work was done when Zechuan Hu was interning at Alibaba DAMO Academy.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 2661–2674
August 1–6, 2021. ©2021 Association for Computational Linguistics

In many real applications, we are generally easy to obtain a small number of target labeled data. These small amounts of data can reflect the diverse strength and weakness of different source models. Concretely, the (small-size) labeled data can be utilized to learn the aggregation strategy of multiple source models or train a new task-specific model in the target language. Both the aggregation model and target task-specific model can map the inputs to the structured outputs but there exists a trade-off. The aggregation model generally has strong cross-lingual ability since source models are firstly well trained[1], but has lower flexibility since source models are usually frozen. Instead, the target task-specific model tends to be more flexible and has strong capacity but has poor performance since the model is easily over-fitted on the small training sample.

Inspired by previous work on multi/cross-view learning (Clark et al., 2018; Jiang et al., 2019; Fei and Li, 2020), we regard the aggregation model (aggregated source view) and the target task-specific model (target view) as two views since they both can map the input sentence to structured outputs. We propose a novel multi-view framework to achieve a good trade-off between the two views. To capture the diverse strength and weakness of multiple source models, we propose three approaches to obtain the aggregated source view from language/sentence/sub-structure level in a coarse-to-fine manner. By encouraging two views to influence each other, the proposed framework can dynamically learn the confidence level of multiple source models in three coarse-to-fine granularity and make the best use of the small number of labeled data, and make both views improved during training. Benefited from the multi-view framework, our proposed approaches can leverage plenty of target unlabeled data to capture the useful target language information (Wu et al., 2020).

The contributions of this work are:

1. We propose to leverage a small number of target labeled data to better aggregate multiple source models.

2. Our approach contains three novel coarse-to-fine approaches to aggregate multiple source models (section 2.2).

3. We propose a novel multi-view learning framework (section 2.3).

4. By utilizing both the label & unlabeled dataset, our approach improves two views simultaneously (section 2.4).

We extensively experiment on three structured prediction tasks, which are named entity recognition (NER), part-of-speech tagging (POS), and dependency parsing. Our proposed approaches outperform several state-of-the-art approaches.

2 Methodology

The left part of Figure 1 depicts the proposed general framework. Our framework contains two views, a **target view** which is a target structured predictor, and an **aggregated source view** based on multiple pre-trained source models. Both views can map the input sentences to the structured outputs and have diverse statistical properties, and thus can provide complementary information to each other (learned by the **consensus component**).

2.1 The Target View

In the general framework, the target view is a task-specific model. We leverage the multilingual bert (mBERT) (Devlin et al., 2019) as the sentence encoder. We feed the input sentence $\mathbf{x}$ to the mBERT and obtain the contextual internal states $\mathbf{h}$, which are utilized by a task-specific module to produce a structured output $\mathbf{y}$. Specifically, we use a Softmax layer for sequence labeling tasks and a biaffine attention mechanism (Dozat and Manning, 2016) followed by (Wu and Dredze, 2019a) for graph-based tasks like dependency parsing. The conditional probability of the structured output given the input sequence is computed by,

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp(\sum_{u \in \mathbf{y}} s(\mathbf{h}, u))}{\sum_{\mathbf{y}'} \exp(\sum_{u \in \mathbf{y}'} s(\mathbf{h}, u))}$$

where $\mathbf{y}'$ is the candidate structured outputs, $\mathbf{y}$ is the structured outputs and u is the sub-structure of $\mathbf{y}$. Sub-structure is the label of each token for sequence labeling and dependency head for dependency parsing. During training with gold labels, the sequence labeling objective function is the cross entropy between the gold labels and the model's

[1] Following Wu et al. (2020), source models are previously trained on their corresponding labeled training set and frozen during training.

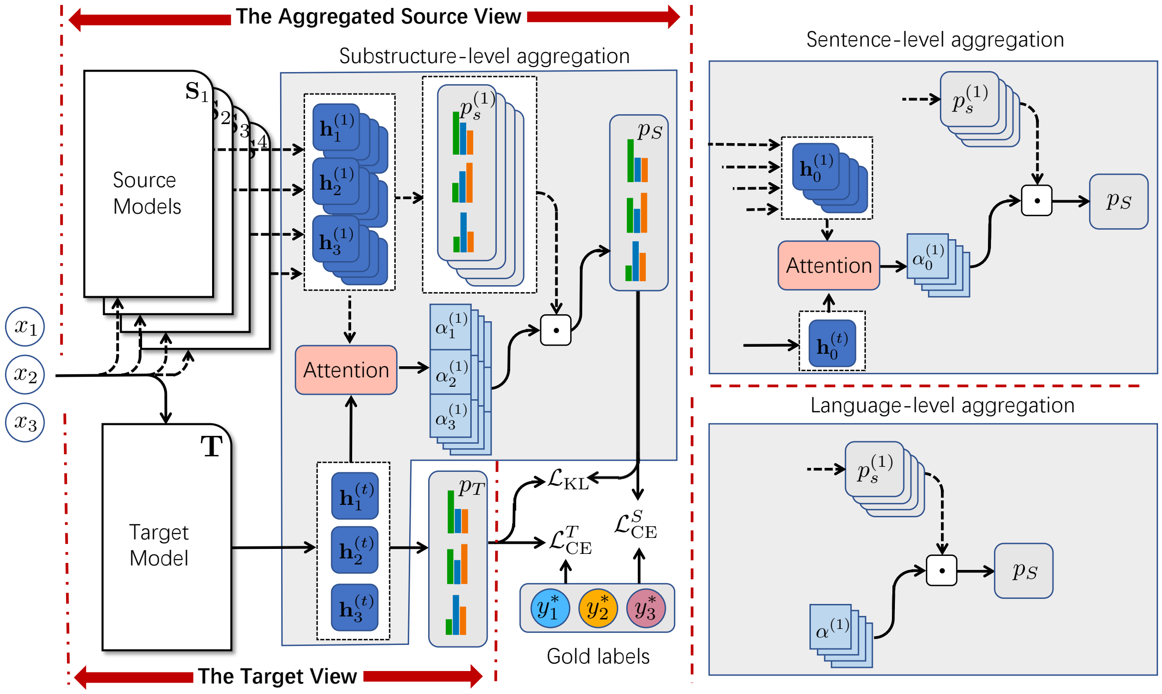

Figure 1: The proposed multi-view framework with K source models ($K = 4$ in this case). The parts with gray background are the aggregation modules of three levels. **Left**: The multi-view framework with substructure-level aggregation as described in section 2.2 and 2.2.3. An input sentence is passed through K source models ($\mathbf{S}$) and one target model ($\mathbf{T}$). K output probabilities from source models are aggregated by a trainable weighting factors α (vector for model/sentence level and matrix for sub-structure level). $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss term on both two views only for labeled data, and $\mathcal{L}_{\text{KL}}$ is the KL-divergence loss between two views for both labeled data and unlabeled data, as described in section 2.1, 2.3, and 2.4. **Right**: The sentence-level aggregation (above) as described in section 2.2.3 and the language-level aggregation as described in section 2.2.1 .

soft predictions [2],

$$\mathcal{L}_{\text{CE}} = -\log p(\mathbf{y}^*|\mathbf{x}) = -\sum_{i=1}^{n} \log p(y_i^*|\mathbf{x})$$

where $\mathbf{y}^*$ is the gold label sequence. In dependency parsing, we use the biaffine parser (Dozat and Manning, 2016) which is one of the state-of-the-art parsers. Following Wu and Dredze (2019a), we replace the BiLSTM encoder with mBERT. Similar to sequence labeling, the biaffine parser models the dependency head separately for each token. Following Anderson and Gómez-Rodríguez (2020), it has two independent distributions, one for head prediction and one for label prediction. The cross-entropy loss for dependency head is,

$$\mathcal{L}_{\text{CE}}(head) = -\log p(\mathbf{t}^*|\mathbf{x}) = -\sum_{i=1}^{n} \log p(h_i^*|\mathbf{x})$$

where h_i^* is the gold head for i-th word of the gold tree $\mathbf{t}^*$. Together with the similar cross-entropy loss of predicted edge labels, the dependency parsing objective function is $\mathcal{L}_{\text{CE}} = \mathcal{L}_{\text{CE}}(head) + \mathcal{L}_{\text{CE}}(label)$.

2.2 The Aggregated Source View

In this section, we take the sequence labeling tasks as an example to introduce our aggregated source view. The source models have the same model structure as the task-specific model of the target view in section 2.1. As presented in figure 1, for a K-source setup, we have K pretrained source models $\mathbf{S}_k$, $k \in \{1, \ldots, K\}$ and the target structured model $\mathbf{T}$. Given a sentence $\mathbf{x} = \{x_0, \ldots, x_n\}$, where x_0 represents the [CLS] token, we feed it to these models and get the internal states $\{\mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(K)}\}$ and the probability distributions $\{p_s^{(1)}, \ldots, p_s^{(K)}\}$ over the structured output of K source models $\mathbf{S}_k$, and $\mathbf{h}^{(t)}$ and p_T of the target model. To aggregate all source models, we propose three novel coarse-to-fine approaches.

2.2.1 Language-level Aggregation

We simply introduce a trainable probability vector α_{lang}, which is depicted on the bottom right part of

[2] This is a common way in the BERT-fintuning setup (Wu and Dredze, 2019a; Wu et al., 2020).

the Figure 1. The final output distribution of the aggregated source view can be computed as,

$$p_S(\mathbf{y}|\mathbf{x}) = \sum_{k=1}^{K} \alpha_{\text{lang}}^{(k)} \cdot p_s^{(k)}(\mathbf{y}|\mathbf{x})$$

We use superscript to represent the index of vector α_{lang}. Note that we use lowercase s, uppercase S, and uppercase T to differentiate the final outputs of the source model, aggregated source view, and target view respectively. In this approach, the k-th source model has the same weight $\alpha_{\text{lang}}^{(k)}$ over all sentences.

2.2.2 Sentence-level Aggregation

In this section, we leverage an attention mechanism (Luong et al., 2015; Vaswani et al., 2017) to learn the weight of each source model on an input sentence, as shown on the top right part of Figure 1. Firstly, we use the internal states of the [CLS] token as sentence representation. Secondly, $\mathbf{h}_0^{(t)}$ from the target model $\mathbf{T}$ is used as a query to attend $\mathbf{h}_0^{(k)}$ from the k-th source model $\mathbf{S}_k$ to produce the probabilities $\alpha_{\text{sent}}(\mathbf{x}) \in \mathcal{R}^K$.

$$\mathbf{K}_0 = [\mathbf{h}_0^{(1)}; \ldots; \mathbf{h}_0^{(K)}]$$
$$\alpha_{\text{sent}}(\mathbf{x}) = \text{Softmax}(\mathbf{h}_0^{(t)} \mathbf{W} \mathbf{K}_0^T)$$

where $\mathbf{K}_0$ is the concatenation of sentence representations from K source models, and $\mathbf{W} \in \mathcal{R}^{d\times d}$ is the bilinear weight matrix. Then the probabilities are utilized to compute the aggregation distribution $p_S(\mathbf{y}|\mathbf{x})$ as follows,

$$p_S(\mathbf{y}|\mathbf{x}) = \sum_{k=1}^{K} \alpha_{\text{sent}}^{(k)}(\mathbf{x}) \cdot p_s^{(k)}(\mathbf{y}|\mathbf{x})$$

In sentence-level aggregation approach, k-th source model has the same weight $\alpha_{\text{sent}}^{(k)}(\mathbf{x})$ over each substructure of a sentence, but different weights over different sentences and thus can capture the diverse strengths of each source on different sentences.

2.2.3 Sub-structure-level Aggregation

We further propose a fine-grained aggregation approach on sub-structure level, which is also based on the attention mechanism. As shown in the left part of Figure 1, for token x_i in a given sentence $\mathbf{x}$, we use its representation $\mathbf{h}_i^{(t)}$ as the query to attend the corresponding representation from each source model. We compute the probabilities $\alpha_{\text{sub}}(x_i)$ for i-th sub-structure as follows,

$$\mathbf{K}_i = [\mathbf{h}_i^{(1)}; \ldots; \mathbf{h}_i^{(K)}]$$
$$\alpha_{\text{sub}}(x_i) = \text{Softmax}(\mathbf{h}_i^{(t)} \mathbf{W} \mathbf{K}_i^T)$$

Then the aggregation distribution becomes,

$$p_S(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \alpha_{\text{sub}}^{(k)}(x_i) \cdot p_s^{(k)}(y_i|\mathbf{x})$$

In this approach, our target model acts as a selector to dynamically assess the multiple source models on sub-structure level.

2.3 Consensus between Two Views

To achieve a good trade-off between the target view and the aggregation view during training, inspired by Clark et al. (2018), we utilize the KL divergence [3] as the metric to encourage the similarity between the two views. For sequence labeling, the objective is,

$$\mathcal{L}_{\text{KL}}(\mathbf{x}) = \mathbb{KL}(p_S(\mathbf{y}|\mathbf{x}) || p_T(\mathbf{y}|\mathbf{x}))$$

2.4 Overall Training Objective

In the model training, for the unlabeled sentences, we only calculate the KL-divergence loss $\mathcal{L}_{\text{U}} = \mathcal{L}_{\text{KL}}$. For the labeled sentences, we train the model with two supervised cross-entropy loss in addition to the KL-divergence loss,

$$\mathcal{L}_{\text{L}} = \lambda_1 \mathcal{L}_{\text{CE}}^S + \lambda_2 \mathcal{L}_{\text{CE}}^T + \lambda_3 \mathcal{L}_{\text{KL}}$$

where λ_1, λ_2 and λ_3 are the interpolation factors. Finally, we introduce an interpolation μ to balance the labeled and unlabeled sentences and the overall learning objective is $\mathcal{L} = \mu\mathcal{L}_{\text{L}} + (1-\mu)\mathcal{L}_{\text{U}}$.

Connections to KD There are mainly four differences between KD (Wu et al., 2020) and our approach:

1. Unlike our approach, KD only utilizes the target unlabeled data, from which it cannot well learn the strength and weakness of different source models (see Sec.1 for more discussion.).

[3] We also try many metrics of measuring the similarity between two probability distributions, e.g., mean squared error (MSE) (Wu et al., 2020), Cosine, and Jensen-Shannon divergence (JS) (Ruder and Plank, 2017), and we find KL perform best.

2. KD assigns equal importance to multiple source models, which can be seen as a fixed uniform vector in our language-level aggregation approach.

3. Besides language-level aggregation, we propose two fine-grained aggregation strategies to dynamically balance the information from source models.

4. To achieve the previously described goal, our approach has trainable parameters in the aggregation component and our multi-view learning framework can jointly learn the parameters of two views.

2.5 Training and Inference Strategies

Following previous work on cross-lingual transfer (Rahimi et al., 2019; Wu et al., 2020), the source models are previously trained on their corresponding labeled training data. During training, we freeze the parameters of the pre-trained source models and only update the parameters of calculating weights α in the aggregated source view, and update all parameters of the target view. In every iteration, we randomly sample a batch of data from the labeled dataset and unlabeled dataset according to the interpolation μ. In the experiments, our model can significantly benefit from this training strategy by controlling the ratio of labeled data and unlabeled data. During the inference phase, we have two options to obtain the predictions: utilizing the aggregated source view or the target view. In our experiments, we use the second one as the main result for its simplicity and better performance.

3 Experiments

We experiment on three structured prediction tasks: NER, POS tagging, and dependency parsing. Following previous work (Rahimi et al., 2019; Wu et al., 2020), we conduct the experiments in a leave-one-out setting in which we hold out one language as the target language and the others as the source languages. To simulate the low-resources scenario, for each training set in a specific target language, we randomly select fifty sentences [4] with the gold annotations and discard the annotations of the remaining sentences to construct the training set. We randomly select six languages from Universal Dependencies Treebanks (v2.2)[5] for dependency parsing and POS tagging tasks. We use the datasets from CoNLL 2002 and CoNLL 2003 shared tasks (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) for NER tasks. We utilize the base cased multilingual BERT (Devlin et al., 2019) as base model for all approaches. We run each approach five times and report the averaged accuracy for POS tagging, f1-score for NER, and unlabelled attachment score (UAS) and labeled attachment score (LAS) for dependency parsing. More details can be found in the Appendix B.1.

3.1 Compared Baselines

We compare the results of the target view of our language/sentence/sub-structure-level approaches which are denoted as Ours-lang/sent/sub respectively, with a large amount of previous state-of-the-art cross-lingual baselines: direct fine-tuning (DT-finetuning), direct transfer (DT), hard knowledge distillation (hard-KD) (Liu et al., 2017), soft knowledge distillation (soft-KD) (Hinton et al., 2015; Wu et al., 2020), unified multilingual model (UMM) which is similar to (Yasunaga et al., 2018; Akbik et al., 2019), and bootstrapping approaches (Yarowsky, 1995; Zhou and Li, 2005; McClosky et al., 2006; Ruder and Plank, 2018) based on UMM.

DT-finetuning We directly fine-tune the task-specific view on fifty labeled data.

DT In DT, there is only test data in the target language. Therefore, we evaluate this approach in three ways: 1) using the mean probability distribution of source models (DT-mean); 2) using the maximal probability distribution of source models over the sub-structure level (DT-max). 3) evaluating each source model and voting on the sub-structure level (DT-vote). We also provide the maximal results of DT on language level (DT-Max(lang)) [6].

Hard-KD The hard knowledge distillation approaches first predict the pseudo labels on target unlabeled training set by using pre-trained source models and then train a new model on the pseudo labeled data (Liu et al., 2017; Rahimi et al., 2019).

[4] We explore the effects of randomness on labeled data in the Appendix C.1 and the results show that our approach is robust to randomness in the selection of labeled data.

[5] https://universaldependencies.org/

[6] We separately evaluate the source language models on the target test data and choose the best score. Since we don't know which source model is the best for DT in practice, the DT-Max(lang) results are only for reference.

With 50 labeled data		CoNLL02/03 NER					POS tagging						
		En	**De**	**Nl**	**Es**	**Avg.**	**En**	**Ca**	**Id**	**Hi**	**Fi**	**Ru**	**Avg.**
✗	*DT-gold*	90.13	84.60	89.09	84.30	87.03	95.71	96.80	94.67	94.39	92.18	97.39	95.19
✗	*DT-max(lang)*	80.85	74.27	81.00	78.42	78.64	87.38	94.26	89.33	87.96	82.47	91.71	88.85
✓	DT-Finetuning	72.71	54.49	57.07	70.82	63.77	85.58	92.90	86.73	86.17	72.57	87.65	85.27
✗	DT-vote	81.81	74.52	81.66	78.51	79.13	89.73	94.26	90.29	89.09	82.82	92.51	89.78
✗	DT-max	82.21	74.98	82.19	78.74	79.53	89.71	94.49	90.13	89.13	83.97	92.78	90.04
✗	DT-mean	82.57	75.33	82.19	78.93	79.76	90.04	94.38	90.40	89.26	83.72	92.86	90.11
✓	hard-KD-cat	83.73	75.56	82.30	79.07	80.17	90.22	94.41	90.60	89.52	84.26	92.80	90.30
✓	hard-KD-vote	83.45	75.80	82.48	79.18	80.23	90.06	94.38	90.52	89.53	83.77	92.65	90.15
✓	hard-KD-max	83.14	75.39	82.27	79.40	80.05	90.16	94.56	90.45	89.41	84.89	92.99	90.41
✓	hard-KD-mean	83.42	75.67	82.306	79.29	80.17	90.32	94.46	90.67	89.61	84.48	92.96	90.41
✓	UMM	78.99	75.26	82.48	78.26	78.75	88.14	93.88	89.65	88.42	83.03	93.26	89.40
✓	Self-training[1]	80.76	75.96	82.91	79.63	79.81	89.68	94.46	90.13	89.16	83.72	94.02	90.19
✓	Tri-training[2]	80.63	76.62	83.14	79.10	79.87	89.83	94.40	90.04	89.69	83.94	**94.05**	90.32
✓	soft-KD-avg[3]	83.52	75.84	82.46	79.24	80.26	90.31	94.62	90.75	89.69	84.55	93.22	90.52
✓	soft-KD-sim[4]	83.58	75.99	82.94	79.63	80.54	89.79	94.80	90.79	89.70	84.55	93.54	90.53
✓	Ours-lang	83.48	75.88	83.02	79.79	80.54	90.27	94.73	90.81	89.62	84.78	93.44	90.61
✓	Ours-sent	83.83	76.13	82.92	80.07	80.74	90.31	94.80	90.91	89.71	84.93	93.51	90.70
✓	Ours-sub	**84.78**	76.56	**84.12**	**80.34**	**81.45**	**91.12**	**95.30**	**91.15**	**90.11**	**85.68**	93.57	**91.16**

[1] Yarowsky (1995); McClosky et al. (2006) [2] Ruder and Plank (2018) [3,4] Wu et al. (2020)

Table 1: Results on CoNLL02/03 NER and POS tagging tasks. The approaches provided for **reference** is marked as *italic*. We compare the best score of our approaches and the best score of the baselines by leveraging almost stochastic dominance (ASD) test (Dror et al., 2019). We mark the the highest score as **bold** if its superiority is significant ($p < 0.05$) and underline otherwise.

We obtain the pseudo labels in four ways: 1) using DT-mean (hard-KD-mean); 2) using DT-max (hard-KD-max); 3) using DT-vote (hard-KD-vote); 4) concatenating all predictions of source models instead of voting (hard-KD-concat). For fairly comparison, we also concatenate the fifty target labeled data into the pseudo labeled data.

Soft-KD Instead of leveraging hard predictions of source models in hard-KD, the soft-KD leverages soft probability distribution of source models. The original Soft-KD (Wu et al., 2020) only focuses on zero-shot NER tasks. Instead, we modify their training objective to leverage fifty target labeled data and adapt it to POS tagging and dependency parsing tasks. (Refer to section 2.4 for details.) We re-implement their two proposed approaches: 1) uniformly aggregating multiple source models (KD-avg); 2) aggregating source models by fixed weights pre-trained on source unlabeled data based on language similarity (KD-sim)[7].

UMM The UMM is trained on the concatenation of all source languages labeled data and fifty labeled data of target language.

Bootstrapping Bootstrapping approaches firstly train a UMM and then add the most confident sentences of target unlabeled data into the training set every iteration during training. We compare our approaches to Self-Training (Yarowsky, 1995; McClosky et al., 2006) and Tri-Training (Ruder and Plank, 2018).

We provide the upper bound results of DT (DT-gold). We construct the upper bound using the gold label set in test data by selecting the gold label if any prediction of source models appears in the gold set. Besides, unlike UMM, self-training, tri-training, and KD-sim, our approaches do not require extra resources like source language training data.

3.2 Results

We report the results in Table 1 for NER and POS tagging, and 2 for dependency parsing.

Common Results on All Tasks As shown in Table 1 and 2, our three proposed approaches outperform most of the baselines on all tasks, which demonstrates the effectiveness of the proposed multi-view learning framework. When trained on only fifty labeled data, the task-specific model shows significantly poor results especially on dependency parsing which verifies our intuition that the task-specific model is easily over-fitted and only training the task-specific model is not sufficient. Notably, UMM, self-training, and tri-training

[7] For more details of the two approaches, please refer to the original paper.

With 50 labeled data		En		Ca		Id		Hi		Fi		Ru		Avg.	
		UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
✗	*DT-gold*	93.30	87.67	92.80	88.61	89.10	81.80	88.54	80.41	84.65	74.67	92.20	86.20	90.10	83.23
✗	*DT-max(lang)*	77.71	67.90	84.39	76.17	76.86	68.37	70.49	52.64	76.62	56.98	72.31	64.45	76.40	64.42
✓	DT-Finetuning	49.75	41.87	53.59	48.38	47.19	38.39	50.32	41.58	32.88	22.22	35.87	28.78	44.93	36.87
✗	DT-vote	80.94	71.65	83.82	75.81	77.79	66.76	75.98	63.18	68.23	52.82	79.80	69.62	77.76	66.64
✗	DT-max	81.07	71.56	84.29	75.87	77.46	65.78	76.54	63.42	69.13	52.79	79.42	69.22	77.99	66.44
✗	DT-mean	81.79	72.96	84.52	76.68	78.45	67.56	76.80	64.31	68.83	54.11	80.54	70.77	78.49	67.73
✓	hard-KD-cat	82.16	74.29	84.41	77.13	78.28	68.26	77.26	65.56	69.61	55.80	80.28	70.90	78.67	68.66
✓	hard-KD-vote	82.46	74.09	84.47	77.02	78.05	67.99	77.83	65.79	69.39	55.31	80.78	71.44	78.83	68.61
✓	hard-KD-max	82.35	74.16	85.13	77.73	77.62	67.45	78.19	66.42	69.49	54.68	80.79	71.52	78.93	68.66
✓	hard-KD-mean	82.69	74.61	84.85	77.41	78.11	68.45	78.23	66.45	69.88	56.04	81.15	72.08	79.15	69.17
✓	UMM	82.89	73.44	83.02	73.24	78.28	63.21	75.36	61.38	66.85	49.13	80.40	70.84	77.80	65.21
✓	Self-training[1]	83.89	74.64	83.76	74.10	79.01	63.31	77.56	63.31	67.95	50.39	80.78	72.20	78.82	66.33
✓	Tri-training [2]	83.97	74.64	83.80	75.34	79.17	63.49	77.94	63.89	68.35	51.07	80.51	71.84	78.96	66.71
✓	soft-KD-avg[3]	82.07	74.64	84.80	77.82	78.18	68.73	78.27	67.46	68.90	54.84	80.83	72.12	78.84	69.27
✓	soft-KD-sim[4]	81.49	72.46	85.49	78.39	77.59	67.90	78.28	67.38	68.63	54.58	80.93	72.19	78.74	68.82
✓	Ours-lang	82.07	74.67	84.94	78.03	78.26	68.76	78.62	67.78	68.66	54.49	81.10	72.62	78.94	69.39
✓	Ours-sent	82.33	74.89	85.25	78.10	78.62	69.03	78.74	67.91	69.06	56.13	81.19	72.54	79.20	69.77
✓	Ours-sub	83.95	**76.67**	**86.00**	**79.25**	**79.41**	**70.13**	**79.40**	**68.58**	**72.36**	**60.21**	**82.15**	**73.70**	**80.54**	**71.42**

[1] Yarowsky (1995); McClosky et al. (2006) [2] Ruder and Plank (2018) [3,4] Wu et al. (2020)

Table 2: Results on the dependency parsing task. (Refer to the caption of Table 1 for the format detail.)

do not yield improvements compared to hard-KD-*, soft-KD-*, and Ours-*, verifying our motivation that simply concatenating all training data is not sufficient to model the difference between multiple sources. We also observe that our three approaches outperform the two KD approaches consistently, indicating that their simple or heuristic-based aggregation strategies are difficult to assess the diverse quality of source models. It is also worth noticing that with a more fine-grained aggregated source view, the target view has stronger performance, especially for Ours-sub [8]. Even though UMM, self-training, tri-training, and soft-KD-sim all utilize source language training data during training, Ours-sub achieves remarkable advantage over these baselines without the extra resources, especially for dependency parsing.

Other results Although Tri-training achieves the highest score and UAS on De of NER and En of parsing respectively, it is not statistically significant compared to Ours-sub and the gap is very marginal ($< 0.1\%$). For NER task, it is probably due to the difference of the capitalization style between De and other languages on CoNLL NER (Chen et al., 2019), which may lead to the negative transfer problem [9]. Besides, the gaps between the DT-gold and the best transfer approaches suggest the large potential space on multi-source transfer tasks.

4 Analysis

4.1 Why the Multi-View Framework Works?

In this section, we study the reason why the proposed framework works. We show the performance of the aggregated source view in Figure 2. It can be seen that with a more fine-grained strategy, the performance of the aggregated source view becomes stronger. It demonstrates the effectiveness of more fine-grained aggregation strategies in the multi-source transfer. The only counter case is language and sentence level on NL, and the performance of the target view drops accordingly. Connecting to Table 1, the target view has the same trends. The reason is probably that the stronger aggregated source view can lead to a stronger target view and vice versa, and the framework achieves a good trade-off to make them both improved.

4.2 Ablation Study

To further understand the proposed framework we investigate the component contributions. We gradually remove some components of our sub-structure-level model, i.e., $\mathcal{L}_{\text{CE}}^{S}$, $\mathcal{L}_{\text{CE}}^{T}$ and $\mathcal{L}_{\text{KL}}$, and evaluate

[8] This is mainly due to the stronger cross-lingual ability of the aggregated source view. We further analyze this in section 4.1.

[9] We speculate that KD-based approaches also suffer from this problem and lead to low results. Our sub-structure-level approach is the second-best system in this case, indicating that it can alleviate this problem by better leveraging labeled data to access the confidence level of source models on more fine-grained-level property.

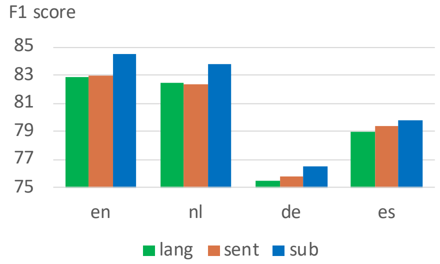

Figure 2: Performance of the aggregated source view on CoNLL02/03 NER tasks.

	En	De	Nl	Es	Avg.
full model	84.95	76.16	83.85	79.79	81.19
$w/o\ \mathcal{L}_{\text{CE}}^{S}$	84.92	75.89	83.38	79.49	80.92
$w/o\ \mathcal{L}_{\text{CE}}^{T}$	84.88	75.59	83.33	79.32	80.78
$w/o\ \mathcal{L}_{\text{CE}}^{S}$ & $\mathcal{L}_{\text{CE}}^{T}$	84.34	74.07	83.07	79.19	80.17
$w/o\ \mathcal{L}_{\text{KL}}$	72.59	54.60	57.19	71.06	63.86

Table 3: Ablation study of Ours-sub model on CoNLL02/03 NER task. w/o denotes ‘without’.

on the NER task. We report the average results of twenty-five runs [10] in Table 3. Without $\mathcal{L}_{\text{KL}}$ the approach degenerates into supervised training with only fifty labeled data and it leads to the largest drop in performance. It is because the model is easily over-fitted. Though the performance drops without one of $\mathcal{L}_{\text{CE}}^{S}$ and $\mathcal{L}_{\text{CE}}^{T}$, it still outperform KD-* baselines of Table 1. $w/o\ \mathcal{L}_{\text{CE}}^{S}$ leads to less drops than $w/o\ \mathcal{L}_{\text{CE}}^{T}$, which suggest that the labeled data influence more in the target model. Besides, without both cross-entropy loss of labeled data, the approach degenerates into a zero-shot manner and results in inferior performance.

4.3 Different Sizes of Unlabeled Data or Labeled Data

In this section, we study the impact of the sizes of labeled data and unlabeled data on the target language for the ours-sub model. We randomly select $\{10, 50, 200, 1000\}$ labeled data and $\{1000, 2000, 4000, \text{All}\}$ unlabeled data. We repeat each experiment five times and report the average results of both two views [11]. It can be seen that with more labeled data or unlabeled data, the results both become higher and the labeled data shows higher influence than the unlabeled data. Unlike the aggregated source view, the target view gains significantly larger boosts when the size of unlabeled data or labeled data increases (the aggregation view generally shows comparable or even superior results to the target view with fewer data). This verifies our motivation that there exists a trade-off between two views. With #0 unlabeled data, the task-specific model is over-fitted when only trained on #200 or less labeled data.

Target unlabeled data	Target labeled data: #10		#50		#200		#1000	
#0	1.22	—	49.13	—	68.53	—	77.01	—
#1000	68.95	70.38	73.42	75.59	75.75	76.11	77.47	77.18
#2000	70.94	71.09	75.18	76.15	76.54	76.37	78.48	77.66
#4000	71.78	71.89	76.49	76.41	77.52	76.59	78.77	77.69
All	74.61	74.66	76.56	76.44	78.26	77.07	79.18	77.76

Table 4: Results on different sizes of target unlabeled data and labeled data on De of NER tasks. In each cell, the right (underlined) and left part denote the results of the aggregated source view and target view respectively.

5 Related Work

Cross-lingual Structured Prediction Comparing to single-source transfer, the multi-source transfer shows superior performance by leveraging multi-source language knowledge (McDonald et al., 2011; Rahimi et al., 2019; Hu et al., 2021). However, the diverse quality of source models sorely hurt the target model. To tackle this challenging problem, Ammar et al. (2016) leverage language embeddings to model language topological similarities. Rahimi et al. (2019) utilize truth inference to obtain the best labeling over multiple unreliable predictors. Hu et al. (2021) models the relations between the predicted labels from the source models and the true labels.Approaches based on the similarity of source and target data are widely studied (Chen et al., 2019; Wu et al., 2020).

Multi/Cross-view Learning Multi-view learning learns multiple representations for the target data. Tri-training approaches (Zhou and Li, 2005; Ruder and Plank, 2018) leverage voting on three separate models to select confident sentences. Jiang et al. (2019); Cai and Lapata (2020) utilize similarity metrics to regularize source-target language pairs. Multi-view learning can also be utilized in training NER models with different kinds of input components (Wang et al., 2021). Cross-view learning (Clark et al., 2018) is a semi-supervised approach that aims to boost the monolingual model's performance. It learns only one model with several auxiliary prediction modules which are treated

[10] We randomly select five different copies of labeled data and run five times for each copy.

[11] We only show the De results due to the space limitation. The results of the other three languages can be found in the Appendix C.2.

as different views. In contrast to it, we focus on the cross-lingual scenario and our two views are a target task-specific model and the aggregation of multiple pre-trained source models.

Contextual Multilingual Language Model Trained on massive unlabeled data of hundreds of monolingual corpus, the contextual multilingual models (Devlin et al., 2019; Conneau et al., 2020) learn common representations for multiple languages. Though cross-lingual transfer learning significantly benefits from these models (Pires et al., 2019; Wu and Dredze, 2019b), large gaps still remain between low and high-resources setups (Hu et al., 2020a; Wu and Dredze, 2020).

6 Conclusion

We propose a novel multi-view framework to selectively transfer knowledge from multiple sources by utilizing a small amount of labeled dataset. Experimental results show that our approaches achieve state-of-the-art performances on all tasks. Moreover, even compared to approaches with extra resources like source language data, our substructure-level approach still shows significant improvements.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (61976139) and by Alibaba Group through Alibaba Innovative Research Program. We thank Yuting Zhen for her support in processing datasets and conducting significance tests.

References

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. 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 2440–2452, Minneapolis, Minnesota. Association for Computational Linguistics.

A. Akbik, T. Bergmann, and Roland Vollgraf. 2019. Multilingual sequence labeling with one model. In *NLDL 2019, Northern Lights Deep Learning Workshop*.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Mark Anderson and Carlos Gómez-Rodríguez. 2020. Distilling neural networks for greener and faster dependency parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 2–13, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7375–7388, Online. Association for Computational Linguistics.

Rui Cai and Mirella Lapata. 2020. Alignment-free cross-lingual semantic role labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3883–3894, Online. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2019. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Steven J. DeRose. 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31–39.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.

Hongliang Fei and Ping Li. 2020. Cross-lingual unsupervised sentiment classification with multi-view transfer learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5759–5771, Online. Association for Computational Linguistics.

Jiang Guo, Darsh Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4694–4703, Brussels, Belgium. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020a. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Zechuan Hu, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020b. An investigation of potential function designs for neural CRF. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2600–2609, Online. Association for Computational Linguistics.

Zechuan Hu, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Risk Minimization for Zero-shot Sequence Labeling. In *the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (**ACL-IJCNLP 2021**)*. Association for Computational Linguistics.

Yong Jiang, Wenjuan Han, and Kewei Tu. 2019. A regularization-based framework for bilingual grammar induction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1423–1428, Hong Kong, China. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Liyuan Liu, Xiang Ren, Qi Zhu, Shi Zhi, Huan Gui, Heng Ji, and Jiawei Han. 2017. Heterogeneous supervision for relation extraction: A representation learning approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 46–56.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Pro-*

ceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Barbara Plank and Željko Agić. 2018. Distant supervision from disparate sources for low-resource part-of-speech tagging. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 614–620, Brussels, Belgium. Association for Computational Linguistics.

Lonneke van der Plas, Paola Merlo, and James Henderson. 2011. Scaling up automatic cross-lingual semantic role annotation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 299–304, Portland, Oregon, USA. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. 2005. To transfer or not to transfer. In *In NIPS'05 Workshop, Inductive Transfer: 10 Years Later*. Citeseer.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054, Melbourne, Australia. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

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.

Mengqiu Wang and Christopher D. Manning. 2014. Cross-lingual projected expectation regularization for weakly supervised learning. *Transactions of the Association for Computational Linguistics*, 2:55–66.

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. In *the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (**ACL-IJCNLP 2021**)*. Association for Computational Linguistics.

Qianhui Wu, Zijia Lin, Börje Karlsson, Jian-Guang Lou, and Biqing Huang. 2020. Single-/multi-source cross-lingual NER via teacher-student learning on unlabeled data in target language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6505–6514, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019a. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844.

Shijie Wu and Mark Dredze. 2019b. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. 2018. Robust multilingual part-of-speech tagging via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 976–986, New Orleans, Louisiana. Association for Computational Linguistics.

Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541.

A Examples Mentioned in The Introduction

Example #1 in practice, we are more likely to encounter the situation where some source languages are not as similar to the target language and may lead to worse performance (Rosenstein et al., 2005; Rahimi et al., 2019). We show the example in Table 5. The results show that for a target language, the gap between the score of different source models can be large ($> 10\%$).

Source \ Target	EN	DE	NL	ES
EN	—	72.77	79.47	75.13
DE	75.96	—	78.47	70.74
NL	69.38	72.35	—	74.16
ES	68.55	63.37	69.12	—

Table 5: Direct bilingual transfer results on the CoNLL02/03 NER task measured in F1 scores (%). We use the multilingual BERT (mBERT) (Devlin et al., 2019) stacked by a Softmax layer to train a source model. Each source model is pre-trained on the labeled training data of the source language and directly evaluated on the target language test data. For a target language, the gap between the highest and lowest scores ranges from 4.4%-10.4%.

Example #2 The model/language level weights can not well capture the diverse strength and weakness of multiple source models. For example in Table 6 [12], none of the three source models predict correctly on the whole sequence, but selecting predictions based on the sub-structure level can obtain the correct label sequence.

	LOCKERBIE	-	JUICIO	CHAVEZ	PIDE	AYUDA	A	...
En	🟥 O	O	🟥 B-PER	🟥 I-PER	O	🟥 B-LOC	O	...
De	B-LOC	O	🟥 B-PER	🟥 I-PER	O	O	O	...
Nl	🟥 O	O	O	B-PER	O	O	O	...
Mean	🟥 O	O	🟥 B-PER	🟥 I-PER	O	O	O	...
Best	🟥 O	O	O	B-PER	O	O	O	...
Gold	B-LOC	O	O	B-PER	O	O	O	...

Table 6: A negative transfer example on Spanish target language. The three pre-trained source models are obtained in the same way in Table 5. Except the sentence of the first row, each row represent predictions from English (En), German (De), Dutch (Nl) source models and gold labels respectively. $\mathbf{Mean}$ and $\mathbf{Best}$ denote the predictions from the uniform and the best weights of three sources' distributions on sentence level respectively. Labels with red background denote wrong predictions. Each source has its advantages on sub-structure level.

B Experimental Details

B.1 Tasks

Dependency Parsing We randomly select five languages together with the English dataset from Universal Dependencies Treebanks (v2.2) for dependency tasks. The whole datasets are English (En), Catalan (Ca), Finnish (Fi), Indonesian (Id), Hindi (Hi), and Russian (Ru). We do not use syntactic information like gold POS tags as many supervised dependency parsers do since we can't assume they are accessible in practice especially for low-resource languages. Even though we can obtain pseudo tags by pre-trained POS taggers of high-resource language, it may introduce unexpected noises and disturb the experiments.

Named Entity Recognition We use the datasets from CoNLL 2002 and CoNLL 2003 shared tasks (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003), which consist of four languages: En, German (De), Dutch (Nl), and Spanish (Es). Each dataset contains four named entity types: Organization, Person, Location, and Miscellaneous. We use the standard splits with the BIO annotation scheme.

POS Tagging For the POS tagging task, we use the same six datasets as the dependency parsing task.

[12] In this example, three pseudo predictions are from three source models pre-trained on En, De, and Nl training set respectively. The three pre-trained source models are obtained in the same way in Table 5.

B.2 Model Configuration

We utilize the base cased multilingual BERT [13] (Devlin et al., 2019) which has 12 transformer blocks, 12 attention heads, and 768 hidden units. Before model training, the K source models are pre-trained with the corresponding source language training sets [14].

Evaluation We select the best hyper-parameters based on the score of the development set on high-resources language, which is English in practice, and adopt the hyper-parameters to other languages. This may lead to sub-optimal results for other languages but is more realistic (Artetxe et al., 2020).

B.3 Hyper-parameters

we select hyper-parameters based on the performance on the English development set and apply them to other target languages. We search the best learning rate for the mBERT model of all the approaches in the range of $\{2e{-}5, 3e{-}5, 5e{-}5\}$, and set it to $2e{-}5$ for its best performance. We list the important hyper-parameters as follows.

Learning Rate for The Top Layer The top layer's learning rate is generally larger than that of mBERT. We search the best learning rate in the range of $\{2e{-}3, 2e{-}4, 2e{-}5\}$.

Interpolations There are three interpolation hyper-parameters in our framework: λ_1, λ_2, and λ_3 in section 2.4 of the main paper. We tune it in the range of $\{0.5, 1, 3, 10\}$.

Sample Ratio There is a hyper-parameter μ for controlling the ratio of the labeled data and unlabeled data. We tune it in the range of $\{0.05, 0.1, 0.3, 0.5, 0.7\}$.

C Additional Analysis

Linguistic Diversity When all the source languages are different from the target language, the source models generally have poor quality and the target model cannot benefit much from the source models. In this case, the cross-lingual transfer is more difficult. Intuitively, our approaches can dynamically learn the confidence level of multiple source models and still facilitate cross-lingual transfer in this case. We experiment on the dependency parsing task over the languages that are drastically different from each other. The sources are English, Mandarin, Arabic, and Vietnamese, and the target is Turkish. In this setting, the source languages are drastically different from the target language. Our results show that Ours-sub (UAS 59.11, LAS 45.02) still outperforms the strongest baseline (KD, UAS 58.69, LAS 44.36).

	English	**German**	**Dutch**	**Spanish**	**Avg.**
Multilingual	77.13	75.08	81.95	77.60	77.94
Self-training[1]	80.57	75.77	82.44	78.49	79.32
Tri-training[2]	80.99	75.99	82.54	78.28	79.45
KD-avg[3]	83.69	75.91	82.59	79.20	80.35
KD-sim[4]	83.70	75.92	82.76	79.39	80.44
Ours-sub	84.95	76.16	83.85	79.79	81.19

[1] Yarowsky (1995); McClosky et al. (2006)
[2] Ruder and Plank (2018) [3,4] Wu et al. (2020)

Table 7: The average results (twenty-five runs) of randomness test of fifty labeled data on CoNLL02/03 NER task. We randomly select five different copies of fifty labeled data (five runs for each copy).

C.1 Effects of Random Seeds on Labeled Data

We further explore the effects of randomness on labeled data as mentioned in section 3 of the main paper. We randomly select five different copies of fifty labeled data to validate its influence. We compare our sub-structure-level model to KD-* and UMM based approaches on CoNLL02/03 NER task. The results are shown in Table 7. Ours-sub still consistently outperforms the second-best baseline, which demonstrates that our approach is robust to randomness in the selection of labeled data.

C.2 Different Sizes of unlabeled data or labeled data

In Table 8, we provide the whole analysis results mentioned in section 4.3 of the main paper in this section.

[13] https://huggingface.co/bert-base-multilingual-cased

[14] In practice, we can obtain the released pre-trained source models on the open-source community, and thus there is no need to use source language data.

Target labeled data:

Target unlabeled data	En								De							
	#10		#50		#200		#1000		#10		#50		#200		#1000	
#0	0.84	—	71.13	—	81.85	—	87.07	—	1.22	—	49.13	—	68.53	—	77.01	—
#1000	79.45	79.38	82.71	82.56	85.06	84.37	87.01	85.73	68.95	70.38	73.42	75.59	75.75	76.11	77.47	77.18
#2000	82.59	82.13	84.23	84.12	85.43	84.87	87.40	85.77	70.94	71.09	75.18	76.15	76.54	76.37	78.48	77.66
#4000	83.92	83.99	84.27	84.22	85.64	85.00	87.45	85.80	71.78	71.89	76.49	76.41	77.52	76.59	78.77	77.69
All	84.73	84.71	84.78	84.72	86.34	85.4	87.46	85.78	74.61	74.66	76.56	76.44	78.26	77.07	79.18	77.76

Target unlabeled data	Nl								Es							
	#10		#50		#200		#1000		#10		#50		#200		#1000	
#0	1.91	—	35.97	—	73.53	—	85.21	—	0.31	—	63.45	—	78.36	—	82.55	—
#1000	78.30	80.66	79.68	81.61	81.50	83.05	84.95	83.64	77.85	78.34	77.86	78.72	79.42	79.4	82.12	80.26
#2000	81.51	82.23	81.64	82.60	82.70	83.41	85.13	84.00	78.70	78.67	79.00	79.13	80.01	79.49	82.24	80.57
#4000	83.15	82.87	82.67	83.11	83.83	83.75	85.43	84.27	79.15	79.13	79.71	79.27	80.14	79.42	82.57	80.61
All	83.32	83.31	84.14	83.56	85.01	84.06	86.59	84.35	79.71	79.22	80.20	79.35	80.15	79.65	82.56	80.57

Table 8: Results on different sizes of target unlabeled data and labeled data. The numbers of vertical and horizontal axis denote the unlabeled data sizes and labeled data sizes respectively. In each cell, the right (underlined) and left part denote the results of the aggregated source view and target view respectively.