Learning Label-Relational Output Structure for Adaptive Sequence Labeling

1st Keqing He
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
Beijing, China
keqing@bupt.edu.cn

2nd Yuanmeng Yan
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
Beijing, China
yanyuanmeng@bupt.edu.cn

3rd Hong Xu
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
Beijing, China
xuhong@bupt.edu.cn

4rd Sihong Liu
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
Beijing, China
liusihong@bupt.edu.cn

5th Zijun Liu
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
Beijing, China
liuzijun@bupt.edu.cn

6th Weiran Xu
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
Beijing, China
xuweiran@bupt.edu.cn

***Abstract*—Sequence labeling is a fundamental task of natural language understanding. Recent neural models for sequence labeling task achieve significant success with the availability of sufficient training data. However, in practical scenarios, entity types to be annotated even in the same domain are continuously evolving. To transfer knowledge from the source model pre-trained on previously annotated data, we propose an approach which learns label-relational output structure to explicitly capturing label correlations in the latent space. Additionally, we construct the target-to-source interaction between the source model M_S and the target model M_T and apply a gate mechanism to control how much information in M_S and M_T should be passed down. Experiments show that our method consistently outperforms the state-of-the-art methods with a statistically significant margin and effectively facilitates to recognize rare new entities in the target data especially.**

***Index Terms*—adaptive sequence labeling, transfer learning, label-relational output structure, latent space**

I. Introduction

Slot tagging is a critical component in spoken dialogue systems. It aims to extract semantic concepts as constraints from the natural language. Traditionally, slot tagging is formulated as a sequence labeling task using the IOB representation. Take a movie-related utterance as an example, "find comedies by James Cameron". The task tries to assign the corresponding entity label to each word, "O B-genre O B-dir I-dir", where a B-XXX tag indicates the first word of slot type XXX, and I-XXX is used for subsequent words of slot type XXX. The tag O indicates words outside of any slot types. The example above finally extracts "comedies" as the movie genre to find and "James Cameron" as the director.

In recent years, neural network models have witnessed a notable success in sequence labeling task. However, the development of such models has largely been hindered by the lack of sufficient training data. In various practical scenarios, entity types to be annotated in the same domain are continuously evolving. It remains an unsolved challenge to effectively make use of the previous annotated data or models and alleviate labor costs for new annotated data. Another urgent need is that domain-specific entities can be recognized using models pre-trained on annotated data with common entities, like Location or Date. Both challenges expect to transfer knowledge from annotated entity data available to new entity types to avoid re-annotating all the samples. But for copyright and privacy concerns, industrial scenarios prevent the release of original training data. Therefore, we focus on incorporating source model pre-trained on previously annotated data and facilitating to recognize new entity types in this paper.

Similar to [1], we define a setting for our research consisting of two aspects: (1) a source model, M_S, already trained to recognize a certain number of categories on the source data, D_S; and (2) a transfer learning (TL) task consisting of training a new model, M_T, on the target data, D_T, where new entity categories appear, in addition to those of the D_S (note that D_S is no longer available to perform TL). D_T is typically much

* This work was partially supported by National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artifical Intelligence" Project No. MCM20190701.

* Weiran Xu is the corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

smaller in size compared to D_S. These kinds of problems regard leveraging knowledge about a source model learned on a source dataset, to improve learning a target model on a target dataset.

Existing works [2], [3] transfer the weights of the source model and then finetune the target model. [4] utilizes the outputs of the top layer of the expert models pre-trained on source data when training new domains, allowing for faster training. [1] proposes the neural adapter to learn the difference between the source and the target label distribution. All of these works ignore the structure of the output space and semantic correlation between old entities and new entities. They treat entity labels as independent classes, which makes transfer learning difficult. The reason is that identifying one label is not helped by data for other labels. We aim at addressing this problem by learning output entity label structure to capture the similarity of entities in the output space, so that data for similar labels can help sequence labeling, even to the extent of enabling few-shot or zero-shot sequence labeling.

Motivated by the problems described above, we propose a novel transfer method with label-relational output structure for sequence labeling. Specifically, we first encode the input sequence using the source model M_S and the target model M_T individually. To tackle the issue of feature discrepancy, we construct strong target-to-source interactions and employ the gate mechanism to determine how much information in M_S and M_T should be passed down. Then we encode the semantics of the label entities and learn the label-relational output structure in the joint context-output space via complex nonlinear relationships. Our model can learn to share parameters across output classifiers and input contexts to better capture the similarity structure of the output space and leverage prior knowledge about this similarity. To the best of our knowledge, we are the first to apply learning joint input-output embeddings to solving transfer learning of adaptive sequence labeling.

To summarize, our major contribution includes:

- We propose a transfer method which learns label-relational output structure to explicitly capturing label correlations.
- We augment our method with the target-to-source interaction between the source model and the target model.
- Empirically, our model can offer significant improvements over previous models on the CoNLL dataset and enable to recognize rare new entity types in the target dataset.

II. STATE-OF-THE-ART ARCHITECTURE IN SEQUENCE LABELING

A standard sequence labeling task can be defined as follow: given an input sequence $X = (x_1, x_2, ..., x_n)$, we need to predict the output sequence $Y = (y_1, y_2, ..., y_n)$. X and Y represent the input and output space respectively. Typically, the model learns to maximize the conditional probability $P(Y|X)$.

In this section, we introduce the state-of-the-art neural model for sequence labeling, CNN+BiLSTM [5], which is also the baseline model we use to evaluate the capability of our method for transfer learning. The general architecture is described in Figure 1. The model is composed of: a CNN layer at the character level, followed by a BiLSTM at the word level, a fully connected layer, and a CRF/output layer. First, a word in the input sequence is represented by both its word-level and character-level embeddings. Then, a bidirectional LSTM processes the input sequence, followed by a fully connected layer. The final prediction y_t is obtained by applying a softmax over the output feature vector h_t:

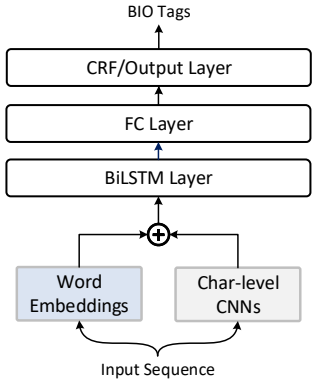


Fig. 1. State-of-the-art baseline architecture in sequence labeling.

$$p\left(y_t|h_t\right) \propto \exp\left(\mathbf{W}^T h_t + b\right) \tag{1}$$

where $\mathbf{W} \in \mathcal{R}^{d_h \times |\mathcal{V}|}$ is a trainable weight matrix and $b \in \mathcal{R}^{|\mathcal{V}|}$ is a bias vector. $\mathcal{V}$ is the label set of sequence labeling. The parameterization in Eq. 1 makes it difficult to learn the structure of the output space or to transfer this information from one label to another because the parameters for output label i, $\mathbf{W}_i^T$, are independent from the parameters for any other output label j, $\mathbf{W}_j^T$. We will dive into the detailed analysis of this problem in the following sections.

Besides, we also implement a Linear Chain CRF [6] over the fully connected layer to improve the prediction ability of the model, by taking the neighboring prediction into account while making the current prediction.

III. PROBLEM FORMALIZATION

Given a source dataset D_S which has E classes and a sequence labeling model M_S trained on D_S, we aim at training a new model M_T on the target dataset D_T which extends original E classes to $E+N$ classes. Note that D_S is no longer available to perform transfer learning and D_T is typically much smaller in size compared to D_S. The core challenge is how we incorporate output features of the source model M_S to facilitate the learning process of the target model M_T without access to the original source dataset D_S.

IV. METHODOLOGY

In this part, we will adequately delineate our transfer method with label-relational output structure for sequence labeling. We start from a brief description of the overall architecture and then dive into the details of each part of the proposed model.

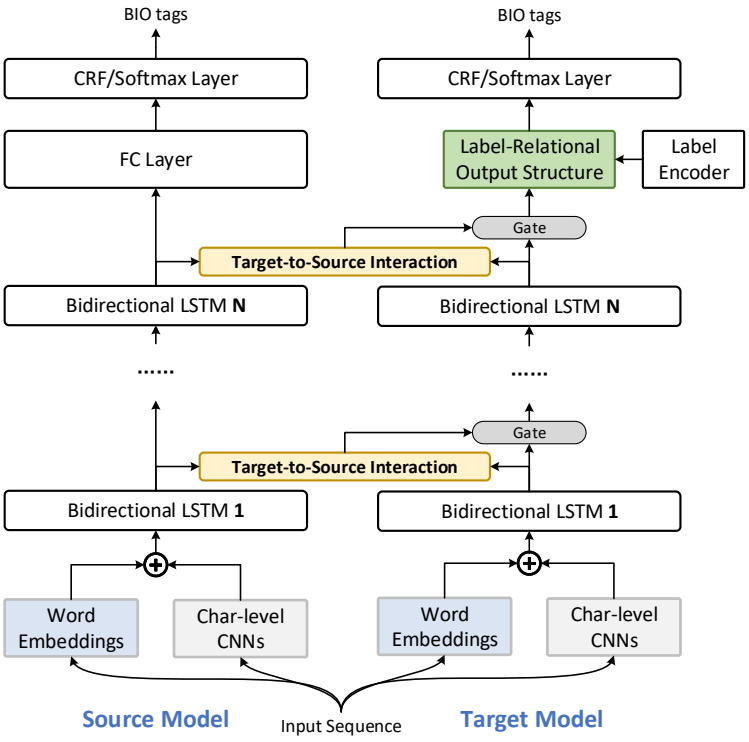


Fig. 2. Our transfer method with label-relational output structure for sequence labeling.

A. Model overview

Figure 2 illustrates the architecture of our method with label-relational output structure. The left represents the source model and the right represents the target model. First, we encode an input word by concatenating its character embeddings and word embeddings. Then, a multi-layer bidirectional LSTM processes the word representation. To incorporate the output features of the source model to the target model, we construct a stronger interaction between the source model and the target model rather than direct concatenation. Besides, we employ a gate mechanism to control how much information in the source model and target model should be passed down. Moreover, instead of using a common softmax output layer, our method aims at explicitly capturing label correlations to facilitate transfer learning by learning label-relational output structure.

B. Target-to-source interaction

Our method encodes the input sequence just like the state-of-the-art architecture in Figure 1. A word in the input sequence is represented by both its word-level and character-level embeddings. Then, a multi-layer bidirectional LSTM processes the input sequence. We assume the hidden states of the i-th BiLSTM layer are represented as $(h_{s_1}^{(i)}, ..., h_{s_n}^{(i)})$ for the source model M_S and $(h_{t_1}^{(i)}, ..., h_{t_n}^{(i)})$ for the target model M_T.

In our work, we aim to transfer knowledge in an incremental, progressive way from the source model to the target model. The intuitive way of fusing two features from M_S and M_T is to concatenate them directly. However, we must tackle the issue of feature discrepancy between the source and target data because of the different label spaces. For example, many words corresponding to new tag categories can already appear in the source data, but they are annotated as O[1] since their labels are not part of the source data annotation yet. [1] uses an RNN-based neural adapter to encode the context information of the source model individually but ignores target-to-source interaction. Thus, we propose to construct a stronger interaction between the source model and target model with an attention-based matching module, which has shown its effectiveness in recent natural language inference models [7]–[9].

Formally consider k-th hidden state $h_{t_k}^{(i)} \in \mathcal{R}^{h_c}$ at i-th RNN layer of the target model and hidden states $(h_{s_1}^{(i)}, ..., h_{s_n}^{(i)})$ of the source model, where n indicates the number of tokens in the input sequence and h_c denotes dimensions. Our model attends on the hidden states of M_S with k-th hidden state $h_{t_k}^{(i)}$ of M_T to capture corresponding information between the source and target models:

$$\alpha_j = (h_{t_k}^{(i)})^T \cdot h_{s_j}^{(i)}, \quad j = 1, 2, ..., n \tag{2}$$

$$\beta_j = \frac{exp(\alpha_j)}{\sum_{j=1}^{n} exp(\alpha_j)} \tag{3}$$

$$\widetilde{h_{t_k}}^{(i)} = \sum_{j=1}^{n} \beta_j \cdot h_{s_j}^{(i)} \tag{4}$$

[1] An O tag indicates that a token belongs to no label.

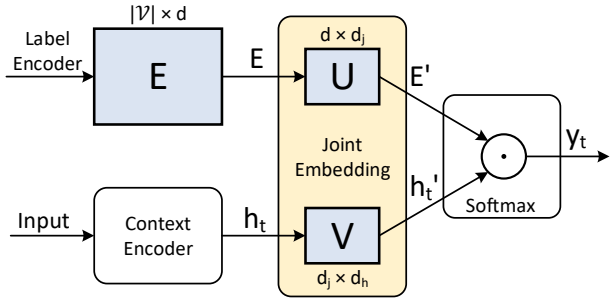


Fig. 3. Learning label-relational output structure for sequence labeling via joint context-output space.

Then we define the following interaction operators:

$$g = \sigma(\mathbf{W}_g[h_{t_k}^{(i)}; \widetilde{h_{t_k}}^{(i)}; h_{t_k}^{(i)} - \widetilde{h_{t_k}}^{(i)}; h_{t_k}^{(i)} * \widetilde{h_{t_k}}^{(i)}]) \quad (5)$$

$$o^{(i)} = g * h_{t_k}^{(i)} + (1-g) * \widetilde{h_{t_k}}^{(i)} \quad (6)$$

where $\sigma(\cdot)$ indicates a sigmoid function and g is the resulting gating function, which control how much information in the source model and target model should be passed down. We expect the model to focus on the relevant parts of M_S and M_T and incorporate them automatically. $o^{(i)}$ will be pass on to the $(i+1)$-th BiRNN layer as input. Besides, we employ multi-level target-to-source interactions on all the layers of BiRNN to leverage multi-grained linguistic knowledge.

C. Label-relational output structure

For approaches directly using the output softmax layer like Eq.1, the underlying label correlations cannot be explicitly considered. If we denote the feature extracted by any arbitrary neural model as $f \in \mathcal{R}^{d_f}$, then the probability of being any given label is calculated by:

$$p = \sigma\left(\mathbf{W}_o^T f\right), \mathbf{W}_o^T \in \mathcal{R}^{|\mathcal{V}| \times d_f} \quad (7)$$

where $\mathcal{V}$ is the label set of sequence labeling. For simplicity, we omit the bias parameter and normalization operation. We can see that every row vector of $\mathbf{W}_o^T$ is responsible for predicting the probability of one particular label. We will refer the row vectors as label vectors for the rest of this paper. As these label vectors are independent, the label correlations are only implicitly captured by sharing the model parameters that are used to extract f. The paradigm of parameter sharing is not enough to impose strong label dependencies and the values of label vectors should be better constrained.

Previous works [10]–[13] substantiate that learning the structure of the output space can help the transfer of learned information across output labels. So data for similar labels can help classification, even to the extent of enabling few-shot or zero-shot classification. We first introduce our label encoder network and then delineate the process of learning the output structure.

Label Encoder Network For the sequence labeling task, the output labels are annotated with IOB format, such as *B-EntityName*, where *EntityName* is usually written as a text phrase. We exploit the semantics provided by pre-trained word embeddings by simply averaging the embeddings of all tokens in the *EntityName* of the label. In general, there may be additional information about each label, such as dictionary entries, cross-lingual resources, or contextual information, in which case we can add an initial encoder for these descriptions which outputs a label embedding matrix. We leave the investigation of additional label information to future work. To distinguish *B-EntityName* from *I-EntityName* of the same entity, we add a two-dimensional vector to the start of the word embedding. Formally we consider the label embedding matrix $\mathbf{E} \in \mathcal{R}^{|\mathcal{V}| \times d}$, where each row vector represents a label embedding.

Learning Output Structure Inspired by recent neural language generation works [13], [14], we expect to learn the joint context-output space and capture the label-relational structure. Let $g_{in}(\cdot)$ and $g_{out}(\cdot)$ be two non-linear projections of d_j dimensions of any context h_t and any embedded output e_j, where e_j is the j-th row vector from the label embedding matrix $\mathbf{E}$. We assume $g_{in}(\cdot)$ and $g_{out}(\cdot)$ could capture the context and output structure respectively and combine them in a joint way. So we change the Eq.1 to the following Eq.8:

$$p\left(y_t|h_t\right) \propto \exp\left(g_{out}(\mathbf{E}) g_{in}(h_t) + b\right) \quad (8)$$

In the experiments, we employ the following forms of $g_{in}(\cdot)$ and $g_{out}(\cdot)$:

$$g_{in}(h_t) = \sigma(\mathbf{V}h_t + b_v) \quad (9)$$

$$g_{out}(\mathbf{E}) = \sigma(\mathbf{E}\mathbf{U} + b_u) \quad (10)$$

where $\sigma(\cdot)$ is a nonlinear activation function such as Relu or Tanh. The matrix $\mathbf{U} \in \mathcal{R}^{d \times d_j}$, and bias $b_u \in \mathcal{R}^{d_j}$ are the linear projection of the encoded outputs, and the matrix $\mathbf{V} \in \mathcal{R}^{d_j \times d_h}$ and bias $b_v \in \mathcal{R}^{d_j}$ are the linear projection of the context.

From the above formula we can learn the joint context-output space which incorporates both components for learning output and context structure:

$$\underbrace{\sigma(\mathbf{E}\mathbf{U} + b_u)}_{\text{Output structure}} \quad \underbrace{\sigma(\mathbf{V}h_t + b_v)}_{\text{Context structure}} \quad (11)$$

where $\mathbf{U} \in \mathcal{R}^{d \times d_j}$ and $\mathbf{V} \in \mathcal{R}^{d_j \times d_h}$ are the dedicated projections for learning output and context structure respectively (which correspond to $\mathbf{W}$ and f projections in Eq. 7). Figure 3 gives a detailed explanation of joint context-output space. $\mathbf{E}$ represents output label space and h_t represents input context vector. $\mathbf{U}, \mathbf{V}$ align output and input to the same dimension space. By joint learning the non-linear projection g_{in} and g_{out}, our method aims to learn complex relationships between input context and output. Since the output label encoding $\mathbf{E}$ exploits label semantics, Eq. 8 can also measure some degree of label dependency in the label-relational output structure. As will be shown in the experiment section, this label-relational output structure provides a further improvement over our state-of-the-art baseline model.

Algorithm 1 Target Model Training

Require: $\{(X^{(n)}, Y^{(n)})\}_{n=1}^{N_\mathrm{T}}$: target training data
$\mathcal{L}$: loss function.
θ^S: parameters of the source model $M_S(X;\theta^S)$
θ^T: parameters of the target model $M_T(X;\theta^T)$
Ensure: $\{y^{(n)}\}_{n=1}^{N_\mathrm{T}}$: predictions
 # initialization period
train the source model $M_S(X;\theta^S)$ using the source data D_S
initialize the parameters θ^T with θ^S
 # training period
for $e = 1 \rightarrow n\_epochs$ **do**
  for $n = 1 \rightarrow N_T$ **do**
    $y^{(n)} = \mathcal{M}(X^{(n)};\theta^S,\theta^T)$
    $\theta^\mathrm{T} := \theta^\mathrm{T} - \alpha\nabla_{\theta^\mathrm{T}}\mathcal{L}[y^{(n)},\theta^\mathrm{T};X^{(n)},Y^{(n)}]$
    θ^S keep fixed
  end for
end for

D. Training procedure

In Algorithm 1, the training procedure for our methods consists of two stages: initialization and training. In the initialization stage, we first train the source model $M_S(X;\theta^S)$ using the source data D_S then initialize the parameters θ^T with θ^S. For the new parameters of M_T in the output layer, we initialize them with weights drawn from the normal distribution, $X \sim \mathcal{N}(\mu, \sigma^2)$, where μ and σ are the mean and standard deviation of the pre-trained weights in the same layer of M_S. In the training stage, we send the same input sequence to M_S and M_T simultaneously, and get the prediction from the output layer of M_T. We use the negative log-likelihood of the data as our training objective. During backpropagation, we keep θ^S fixed and only update θ^T.

V. EXPERIMENTS

A. Dataset

We adopt the dataset from [1] based on CoNLL 2003 NER dataset in our experiments. The original dataset includes four types of named entities: organization, person, location and miscellaneous (represented by *ORG*, *PER*, *LOC*, and *MISC*, respectively). For our experiments, we divide the CoNLL train set into 80%/20% as D_S and D_T. Please note that in the subsequent step, D_S is no longer available for training M_T. We make *LOC* the new label to be detected in the subsequent step. Hence we replace all the *LOC* label annotations with *O* when they appear in D_S. Instead, we keep *LOC* as it is in D_T. We repeat this process for all four categories to obtain four datasets for our TL setting. Please refer to more detailed statistics in the original dataset paper [1].

B. Implementation details

We implement all sequence labeling models within the AllenNLP framework [15] and use the negative log-likelihood of the data as our training objective. We train models for 100 epochs and select the model that performs best on the development set via early stopping. We use 300 dimension Glove pre-trained embedding to initialize the weights of the embedding layer. We use the Adam optimizer [16] with a learning rate of 0.0001 and gradient clipping of 10.0. We use dropout of 0.2 in the BiLSTM and set the size of the BiLSTM layers to 300 dimensions. The models are evaluated with the F1 score as in the official CoNLL 2003 shared task [17].

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON CONLL DATASET. WE REPORT THE F1 SCORES OF THE TEST SETS BOTH ON THE SOURCE DATA D_S AND THE TARGET DATA D_T. "ORI" INDICATES THE ORIGINAL 3 NE CATEGORIES IN THE SOURCE DATA, WHILE "NEW" INDICATES THE NEW NE CATEGORY IN THE TARGET DATA. "ALL" IS THE OVERALL TEST F1 FOR ALL 4 NE CATEGORIES IN THE TARGET DATA.

Model	M_S	M_T		
	Ori	Ori	New	All
baseline [5]	91.35	89.66	90.20	89.79
baseline w/o CRF	91.06	86.52	87.19	86.68
baseline+finetuning	91.35	90.83	88.96	90.39
baseline+finetuning w/o CRF	91.06	90.42	89.39	90.18
adapter [1]	91.35	91.08	90.73	90.99
adapter w/o CRF	91.06	90.94	89.33	90.56
our method	91.36	**91.55***	**93.62***	**92.83***
our method w/o CRF	91.08	91.40	93.01*	92.66*

C. Evaluation on public datasets

Table I shows the results of the CoNLL dataset where we make *LOC* the new label in the target data. We report the F1 scores of the test sets both on the source data D_S and the target data D_T. The numbers with * indicate that the improvement of our model over all baselines is statistically significant with $p < 0.05$ under t-test. Baseline [5] follows the same architecture as Figure 1 and we train M_S on D_S and M_T on D_T, respectively. Since we do not perform weight transfer between M_S and M_T, baseline [5] can be used to substantiate the effect of transfer learning. Based on the baseline [5], baseline+finetuning first trains M_S on D_S with the same model architecture, then initializes M_T with the weights of M_S, finally finetunes M_T on the target dataset D_T. [1] proposes an RNN-based neural adapter to help transfer learning. Note that we aim to confirm the effects of our transfer technique. Therefore, we employ the same sequence labeling model with different transfer settings in the experiments.

Comparing the baseline with the other transfer methods, we can see that transferring weights from the initial model always boosts the performance of original 3 NE categories but fails in new target NE for the standard pre-training and fine-tuning TL paradigm. We assume that label disagreement of new NE in the source and target data is adverse to the recognition capability of the target model. Therefore, our transfer method tackles this discrepancy and improve F1-score in new target NE with a statistically significant margin up to 2.44% by learning the label-relational output structure and constructing multi-grained target-to-source interaction. Moreover, our method outperforms state-of-the-art transfer method [1] with a margin of 1.84% in overall F1-score, which substantiates the neural adapters only on the top layer of RNN are insufficient to mitigate the discrepancies between the source and target label distribution. Taking a deep insight into the F1 scores of the new NE category, we find that our method

TABLE II

PERFORMANCE COMPARISON OF DIFFERENT METHODS ON ATIS DATASET WHERE WE SELECT THE TOP 58 ENTITY TYPES AND IGNORE CLASSES WITH SAMPLE NUMBERS BELOW 10. WE RANDOMLY SELECT 10 CLASSES AS THE NEW TARGET CATEGORIES IN THE REST OF 58 CLASSES. HENCE D_S CONTAINS 48 ENTITY TYPES AND D_T CONTAINS 58 TYPES. WE PERFORM ALL THE EXPERIMENTS WITH CRF LAYER. THE OTHER SETTINGS ARE SIMILAR TO THE CONLL DATASET.

Model	M_S	M_T		
	Ori	Ori	New	All
baseline [5]	95.67	95.15	92.99	95.06
baseline+finetuning	95.67	95.72	91.91	95.52
adapter [1]	95.67	95.66	93.38	95.58
our method	95.67	**95.98**	**93.80***	**95.86**

TABLE III

OVERALL F1 SCORES IN RECOGNIZING DIFFERENT TARGET NE CATEGORIES OF THE TEST SET OF THE SUBSEQUENT STEP.

New Target NE Catagory	Model			
	baseline	finetuning	adapter	our method
LOC	89.79	90.39	90.99	**92.83**
PER	88.33	90.23	90.36	**92.41**
ORG	88.77	89.28	90.16	**92.72**
MISC	87.64	90.30	90.34	**92.49**

achieves much more improvements of 4.66% compared to the baseline+finetuning, which further confirms the effectiveness of our method. Besides, we also find that transfer methods result in faster and more stable convergence than the baseline without weight sharing.

To validate the generalization capability of our method, we also perform experiments on another public dataset ATIS. The results are shown in Table II. The numbers with * indicate that the improvement of our model over all baselines is statistically significant with $p < 0.05$ under t-test. Although the baseline model has achieved great results, our method still makes further improvements, especially on new entity types. The results substantiate the effectiveness and generalization capability of our method for new entities in the target dataset.

VI. QUALITATIVE ANALYSIS

A. Results of all NE categories

It is important to ensure that the improvement in the performance is not specific to a target NE category. Thus, we performed additional experiments on CoNLL dataset, using other NEs as the target in the subsequent step.

In Table III, we present the overall F1 scores while recognizing four different new NE categories respectively on four CoNLL datasets. The first column in the table identifies different target NE categories as explained in the Dataset section. The other four columns present the results of the models without any TL method (baseline), with the transferred parameters (baseline+finetuning), and with the neural adapter [1] and our method respectively.

The results of Table III indicate those transfer methods achieve a consistent improvement in the overall F1 score, especially for our method with label-relational output structure. On average, our method gains 3.98 points of improvement in F1 score on the CoNLL dataset compared to the baseline and 2.15 points of improvement to the neural adapter. Our proposed method improves the overall performance of recognizing NEs in the target data, regardless of target NE types, which substantiates the effectiveness of our method.

TABLE IV

COMPARISON WITH DIFFERENT MODEL COMPONENTS ON THE CONLL DATASET. WE REPORT THE F1 SCORES OF THE TEST SETS ON THE TARGET DATASET D_T. "ORI" INDICATES THE ORIGINAL 3 NE CATEGORIES IN THE SOURCE DATA, WHILE "NEW" INDICATES THE NEW NE CATEGORY IN THE TARGET DATA. "ALL" IS THE OVERALL TEST F1 FOR ALL 4 NE CATEGORIES IN THE TARGET DATA.

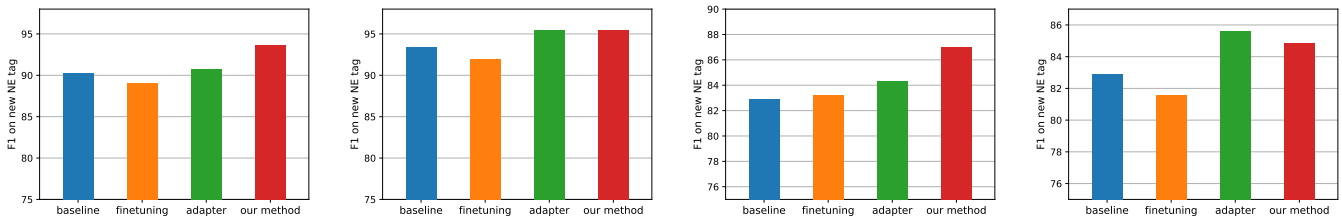
Model	M_T		
	Ori	New	All
baseline+finetuning	90.83	88.96	90.39
our method	**91.55**	**93.62**	**92.83**
- w/o label-relational output structure	91.12	92.08	91.56
- w/o target-to-source interaction	91.36	92.68	91.89
- w/o gate mechanism	91.57	92.81	91.99

We have explained the issue of label discrepancy between the source and target dataset in the previous section. Therefore, we aim to figure out whether our method can make a difference to tackle the issue. Figure 4 shows F1 scores of the new target NE categories on the four datasets, where each subgraph represents a target NE category. We can see that baseline+finetuning always results in a drop of F1 score compared to the baseline without any weight sharing because of the label discrepancy. In all cases, our method significantly outperforms baseline+finetuning with an average improvement of 3.83% for the target entity categories. Moreover, in most cases, we also outperform the state-of-the-art transfer method, adapter [1] by 1.21% points. We argue that our method can better help in resolving the annotation disagreement between the source and the target data.

B. Ablation studies

To quantify the effects of different model components, we report the performance of model variants in Table IV. We implement three model variants: w/o label-relational output structure, w/o target-to-source interaction, and w/o gate mechanism. For the variant w/o label-relational output structure, we simply use a common softmax layer. For the variant w/o target-to-source interaction, we use the original output features of M_S directly. For the variant w/o gate mechanism, we concatenate two output features of M_S and M_T.

From the experiment results, all the components we propose effectively improve the performance of transfer learning for sequence labeling. Among the three components, label-relational output structure is the most essential one. Without learning output structure, our method only achieves F1 score of 91.56%, much lower than our full model by 1.27 absolute points. The other model variants get lower F1 scores by 0.94%(w/o target-to-source interaction) and 0.84%(w/o gate mechanism). Meanwhile, target-to-source interaction also



(a) New category(*LOC*) (b) New category(*PER*) (c) New category(*ORG*) (d) New category(*MISC*)

Fig. 4. New target category Test F1 of baseline, finetuning, adapter, and our method. Figure (a) (b) (c) (d) represent the F1 scores of new target category *LOC*, *PER*, *ORG*, *MISC*, respectively.

makes a significant difference. In summary, all of these components we propose can boost the performance of transfer learning for sequence labeling and efficiently tackle the challenge of label discrepancy.

VII. Related Work

Learning Output Space Learning the structure of the output space improves a wide variety of tasks, such as object recognition and novelty detection in images [18], [18]–[20], zero-shot prediction in texts [21], [22]. The approach has been particularly successful in natural language generation tasks, where word embeddings give a useful similarity structure for next-word-prediction in tasks such as machine translation [23] and language modeling [24]. Recent works have shown great improvements over vanilla architectures by sharing parameters across outputs through a bilinear mapping on neural language modeling [12] or a dual nonlinear mapping on neural machine translation [13], which can boost the classifier.

Transfer Learning Neural networks based TL has proven very effective for various NLP tasks [25]. [26] applies TL to NER with different NE categories by pre-training a linear-chain CRF. [27] constructs label embeddings to automatically map the source and target label types to help improve the transfer between similar domains, where the label types are semantically similar. [28] proposes progressive neural networks to solve reinforcement learning tasks while being immune to parameter forgetting. The networks leverage knowledge from previous trained models with an adapter realized by a feed-forward neural layer with non-linear activation. The adapter is actually an additional connection between new model and trained models. [1] employs BiLSTM based adapters in a sequence-to-sequence way to map the output sequence in the source space to the output sequence in the target space.

Sequence Labeling Recent state-of-the-art models of sequence labeling are recurrent neural network models, which incorporate character-level and word-level embeddings and/or additional morphological features. [29] uses BiLSTM+CRF to achieve state-of-the-art performance (90.10 in terms of test F1 on CoNLL 2003 NER dataset). [5] also implements a similar BiLSTM model with convolutional filters as character feature extractor, achieving 91.62 in the F1 score (BiLSTM+CNN+lexical features). In this work, we choose to use the BiLSTM and BiLSTM+CRF for sequence labeling, to confirm whether our proposed transfer method can improve transfer learning of sequence labeling task.

Pre-trained LM Unsupervised pre-trained models(BERT [30], ELMo [31]) achieve huge success. BERT transfers linguistic knowledge from unlabeled corpus while our paper focuses on transfer learning from labeled only data. We can see that they leverage knowledge from two disentangled sources with different techniques and could be combined to enhance each other. In terms of computational complexity, elaborate supervised transfer methods are superior to BERT, especially compared to pre-training on the unlabeled corpus. For performance, our method also achieves good results on sequence labeling. In our experiment scenario, we highlight the label discrepancy between D_S and D_T and propose label-relational output structure and target-to-source interaction. Only using BERT embeddings can't completely solve the issue because it only captures context semantics and neglects label relations. In this paper, we aim to confirm the effects of our method on transfer learning only using labeled data. Thus, we employ the same baseline model as [1] for fair comparison.

VIII. Conclusion and Future Work

In this paper, we propose a novel approach which learns label-relational output structure to explicitly capturing label correlations in the latent space for adaptive sequence labeling. Along with the target-to-source interaction between the source model M_S and the target model M_T, we achieve significant improvements over previous methods on public CoNLL dataset. Besides, our method effectively facilitates to recognize rare new entities in the target dataset especially. We plan to apply our method to cross-domain scenarios and incorporate deep semantic knowledge of entity labels to the output structure.

IX. Acknowledgment

We thank the anonymous reviewers for their helpful comments and suggestions.

References

[1] L. Chen and A. Moschitti, "Transfer learning for sequence labeling using source model and target data," *ArXiv*, vol. abs/1902.05309, 2019.

[2] Y.-B. Kim, K. Stratos, and R. Sarikaya, "Frustratingly easy neural domain adaptation," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 387–396.

[3] S. Upadhyay, M. Faruqui, G. Tür, H.-T. Dilek, and L. Heck, "(almost) zero-shot cross-lingual spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6034–6038.

[4] Y.-B. Kim, K. Stratos, and D. Kim, "Domain attention with an ensemble of experts," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 643–653.

[5] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016.

[6] J. D. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.

[7] L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin, "Natural language inference by tree-based convolution and heuristic matching," in *ACL*, 2015.

[8] X. Y. Guo, X.-D. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced lstm for natural language inference," in *ACL*, 2016.

[9] Q. Q. Chen and W. Wang, "Sequential attention-based network for noetic end-to-end response selection," *ArXiv*, vol. abs/1901.02609, 2019.

[10] H. Inan, K. Khosravi, and R. Socher, "Tying word vectors and word classifiers: A loss framework for language modeling," *arXiv preprint arXiv:1611.01462*, 2016.

[11] O. Press and L. Wolf, "Using the output embedding to improve language models," *arXiv preprint arXiv:1608.05859*, 2016.

[12] K. Gulordava, L. Aina, and G. Boleda, "How to represent a word and predict it, too: Improving tied architectures for language modelling," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2936–2941.

[13] N. Pappas, L. M. Werlen, and J. Henderson, "Beyond weight tying: Learning joint input-output embeddings for neural machine translation," *arXiv preprint arXiv:1808.10681*, 2018.

[14] N. Pappas and J. Henderson, "Deep residual output layers for neural language generation," 2019.

[15] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer, "Allennlp: A deep semantic natural language processing platform," *arXiv preprint arXiv:1803.07640*, 2018.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[17] E. F. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *arXiv preprint cs/0306050*, 2003.

[18] J. Weston, S. Bengio, and N. Usunier, "Wsabie: Scaling up to large vocabulary image annotation," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[19] Y. Zhang, B. Gong, and M. Shah, "Fast zero-shot image tagging," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 5985–5994.

[20] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang, "Zero-shot visual recognition using semantics-preserving adversarial embedding networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1043–1052.

[21] J. Nam, E. L. Mencía, and J. Fürnkranz, "All-in text: Learning document, label, and word representations jointly," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[22] A. Rios and R. Kavuluru, "Few-shot and zero-shot multi-label learning for structured label spaces," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, vol. 2018. NIH Public Access, 2018, p. 3132.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[24] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing lstm language models," *arXiv preprint arXiv:1708.02182*, 2017.

[25] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin, "How transferable are neural networks in nlp applications?" *arXiv preprint arXiv:1603.06111*, 2016.

[26] L. Qu, G. Ferraro, L. Zhou, W. Hou, and T. Baldwin, "Named entity recognition for novel types by transfer learning," *arXiv preprint arXiv:1610.09914*, 2016.

[27] Y.-B. Kim, K. Stratos, R. Sarikaya, and M. Jeong, "New transfer learning techniques for disparate label sets," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 473–482.

[28] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[29] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019.

[31] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. of NAACL*, 2018.