This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2931569, IEEE Transactions on Pattern Analysis and Machine Intelligence

Dual Adversarial Transfer for Sequence Labeling

Joey Tianyi Zhou, Hao Zhang, Di Jin and Xi Peng

Abstract—We propose a new architecture for addressing sequence labeling, termed Dual Adversarial Transfer Network (DATNet). Specifically, the proposed DATNet includes two variants, i.e., DATNet-F and DATNet-P, which are proposed to explore effective feature fusion between high and low resource. To address the noisy and imbalanced training data, we propose a novel Generalized Resource-Adversarial Discriminator (GRAD) and adopt adversarial training to boost model generalization. We investigate the effects of different components of DATNet across different domains and languages, and show that significant improvement can be obtained especially for low-resource data. Without augmenting any additional hand-crafted features, we achieve state-of-the-art performances on CoNLL, Twitter, PTB-WSJ, OntoNotes and Universal Dependencies with three popular sequence labeling tasks, i.e. Named entity recognition (NER), Part-of-Speech (POS) Tagging and Chunking.

Index Terms—Sequence labeling, named entity recognition, chunking, part-of-speech tagging, transfer learning, natural language processing, adversarial training.

1 Introduction

Sequence labeling is one type of fundamental pattern recognition task that involves the automatic assignment of a categorical label to each member of a sequence of observed values. This occurs in a number of applications of natural language processing (NLP), bioinformatics, and so on. For example, in document analysis, part of speech (POS) tagging seeks to assign a part of speech to each word (token) in an input sentence or document, e.g., noun, verb, adjective. Named entity recognition (NER) or called entity identification is a sub-task of information extraction which aims to locate and classify elements of texts into pre-defined categories such as the names of persons, organizations, locations and so on. As for genetic databases, researchers seek to build the prediction model for assigning values, e.g., A, G, C and T, to each nucleotide in DNA sequences [1].

The sequence labeling, especially for NLP tasks such as NER, is usually required to detect not only the type of the element, but also the element boundaries. To the end, it is necessary to deeply understand the contextual semantics to disambiguate the different types of same element. To tackle this challenging problem, most early studies were based on hand-crafted rules, which show sub-optimal performance in practice. To achieve better result, some efforts are devoted to developing learning based algorithms, especially neural network based methods, and the state-of-the-art have been consecutively advanced [2]–[7]. These end-to-end models generalize well on new elements based on features which are automatically learned from the training data. However, when the annotated training data is small, especially in the low resource scenario [8], the performance of these methods significantly degrades since the hidden feature representations cannot be adequately learned.

Recently, more and more approaches have been proposed to address low-resource sequence labeling. Early works [9], [10] primarily assumed a large parallel corpus and focused on exploiting them to project information from high resource to low one. Unfortunately, such a large parallel corpus may be unavailable for many low-resource languages. More recently, cross-resource word embedding [11]–[13] was proposed to bridge the low and high resources and enable knowledge transfer. Although the aforementioned transfer-based methods show promising performance in low-resource sequence labeling, there are two issues deserved to be further investigated on:

1) **Representation Difference** they did not consider the representation difference across resources and enforced the feature representation to be shared across languages/domains.
2) **Resource Data Imbalance** the training size of high-resource is usually much larger than that of low-resource.

Almost all existing methods neglect the difference in their models, thus resulting in poor generalization.

In this work, we present a novel approach termed **Dual Adversarial Transfer Network (DATNet)** to address the above issues in a unified framework for low-resource NLP sequence labeling. Specifically, to handle the representation difference, we first investigate two architectures of hidden layers (we use bi-directional long-short term memory (BiLSTM) model as hidden layer) for transfer. The first architecture is that all the units in hidden layers are common units shared across languages/domains. The second one is composed of both private and common units, where the private part preserves the independent language/domain information. Extensive experiments are conducted to show their advantages over each other in different situations. On the top of common units, the adversarial discriminator (AD) loss is introduced to encourage the resource-agnostic representation so that the knowledge from high resource can be more compatible with low resource. To handle the imbalance issue of data resource, we further propose a variant of the AD loss, termed *Generalized Resource-Adversarial Discriminator (GRAD)*, to impose the resource weight during training so that low-resource and hard samples can be paid more attention to. In addition, we create adversarial samples to conduct the *Adversarial Training (AT)* for further improving the generalization and alleviating over-fitting problem. To achieve end-to-end training and obtain prominent

- *J. T. Zhou is with IHPC, A*STAR; E-mail: joey_zhou@ihpc.a-star.edu.sg*
- *H. Zhang is with A*AI, A*STAR; E-mail: zhang_hao@scei.a-star.edu.sg*
- *D. Jin is with CSAIL, MIT; E-mail: jindi15@mit.edu*
- *X. Peng is with College of Computer Science, Sichuan University, Chengdu, China; E-mail: pengx.gm@gmail.com*
- *J. T. Zhou and H. Zhang contributed to this work equally.*

improvements on a series of sequence labeling tasks especially for low-resource data, we unify two kinds of adversarial learning, i.e., GRAD and AT, into one transfer learning model, termed Dual Adversarial Transfer Network (DATNet). With the help of the pretrained language model, our method could advance state of the art on several tasks. Different from prior works, we do *not* use additional hand-crafted features and do *not* use cross-lingual word embeddings, while addressing the cross-language tasks effectively.

2 Related Work

Sequence Labeling Sequence labeling is a type of pattern recognition task which aims at automatically assigning a categorical label to each element of a sequence from free text. For example, NER task tries to detect named entities (e.g. person, organization, and location) in the text. The early works applied CRF, SVM, and perception models with hand-crafted features [14]–[16]. With the development of deep learning, the research focus has been shifting towards deep neural networks (DNN), which requires little feature engineering and domain knowledge [4], [17], [18]. Collobert *et al.* [2] proposed a feed-forward neural network with a fixed sized window for each word, which failed in considering useful relations between long-distance words. To overcome this limitation, Chiu *et al.* [5] presented a bidirectional LSTM-CNNs architecture that automatically detects character- and word-level features. Ma *et al.* [6] further extended it into bidirectional LSTM-CNNs-CRF architecture, where the CRF module was added to optimize the output label sequence. Liu *et al.* [19] proposed task-aware neural language model termed LM-LSTM-CRF, where character-aware neural language models were incorporated to extract character-level embedding under a multi-task framework.

Transfer Learning Transfer learning is a powerful tool of sequence labeling, especially for the low-resources tasks. To bridge the resource differences in domains, languages and high/low resource scenario, transfer learning methods for sequence labeling could be divided into following two categories: the parallel corpora based transfer and the shared representation based transfer. Early works mainly focused on exploiting parallel corpora to project information between the high- and low-resource language [9], [10], [20]–[22]. For example, Chen *et al.* [9] and Feng *et al.* [21] proposed to jointly identify and align bilingual named entities. On the other hand, the shared representation methods do not require the parallel correspondence [23]. For instance, Fang *et al.* [11] proposed cross-lingual word embeddings to transfer knowledge across resources. Yang *et al.* [13] presented a transfer learning approach based on a deep hierarchical recurrent neural network (RNN), where the full/partial hidden features between source and target tasks are shared. Ni *et al.* [24], [25] utilized the Wikipedia entity type mappings to improve low-resource NER. Al-Rfou *et al.* [26] built massive multilingual annotators with minimal human expertise by using language agnostic techniques. Mayhew *et al.* [27] created a cross-language sequence tagging system, which works well for very minimal resources by translating annotated data of high-resource into low-resource. Cotterell *et al.* [28] proposed character-level neural CRFs to jointly train and predict low- and high-resource languages. Pan *et al.* [29] proposes a large-scale cross-lingual named entity dataset which contains 282 languages for evaluation. In addition, multi-task learning [30]–[36] shows that jointly training on multiple tasks/languages helps improve performance. Different from transfer learning methods, multi-task learning aims at improving the performance of all the resources instead of low resource only. Most recently, more and more works [37]–[42] focus on building general language model to learn informatively contextualized features from large-scale language corpus and improve their generalization ability for various tasks through unsupervised or supervised learning.

Adversarial Learning Adversarial learning originates from Generative Adversarial Network (GAN) [43], which shows promising performance in computer vision. Recently, many works have tried to apply adversarial learning to NLP tasks. Chen *et al.* [44] proposed an adversarial deep averaging network for sentiment classification task, which transfers the knowledge learned from the labeled data on a resource-rich source language to low-resource languages where only unlabeled data exists. Liu *et al.* [45] presented an adversarial multi-task learning framework for text classification. Gui *et al.* [46] applied the adversarial discriminator to POS tagging for Twitter. Kim *et al.* [47] proposed a language discriminator to enable language-adversarial training for cross-language POS tagging. Chen *et al.* [48] built a multinomial adversarial network to tackle the text classification problem in the real-world multi-domain setting. Besides adversarial discriminator, adversarial training is another concept originally introduced by [49], [50] to improve the robustness of image classification model by injecting malicious perturbations into input images. Recently, Miyato *et al.* [51] proposed a semi-supervised text classification method by applying adversarial training, where the adversarial perturbations were added onto word embeddings for the first time. Yasunaga *et al.* [52] applied adversarial training to POS tagging. Different from all these adversarial learning methods, our method integrates both the adversarial discriminator and adversarial training in an unified framework to enable end-to-end training.

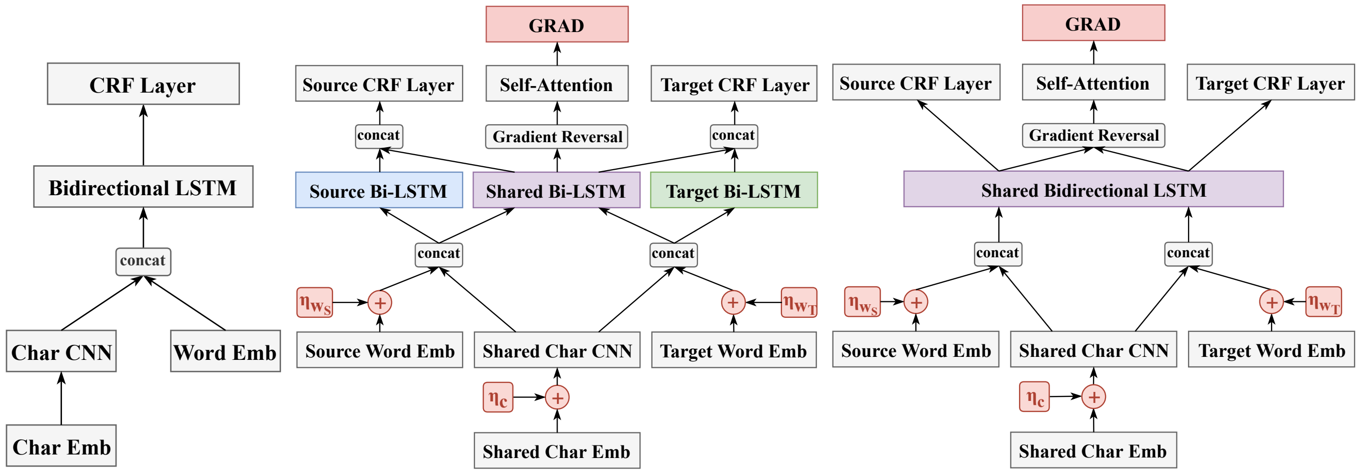
3 Dual Adversarial Transfer Network

In this section, we introduce DATNet in more details. We first describe a base model for NER, and then discuss the proposed two transfer architectures for DATNet.

3.1 Basic Architecture

We follow state-of-the-art models for sequence labeling task [3]–[6], i.e., LSTM-CNNs-CRF based structure, to build the base model. It consists of the following pieces, i.e., character-level embedding, word-level embedding, BiLSTM for feature representation, and CRF as the decoder. The character-level embedding takes a sequence of characters in the word as atomic units input to derive the word representation that encodes the morphological information, such as root, prefix, and suffix. These character features are usually encoded by character-level CNN or BiLSTM, then concatenated with word-level embedding to form the final word vectors. On the top of them, the network further incorporates the contextual information using BiLSTM to output new feature representations, which is subsequently fed into the CRF layer to predict label sequence. Although both of the word-level layer and the character-level layer can be implemented using CNNs or RNNs, we use CNNs for extracting character-level and RNNs for extracting word-level representation. Fig. 1(a) shows the architecture of the base model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2931569, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Base Model (b) DATNet-P (c) DATNet-F

Fig. 1. The general architecture of proposed models.

3.2 Dual Adversarial Transfer Architecture

3.2.1 *Character-level Encoder*

Previous works have shown that character features can boost sequence labeling performance by capturing morphological and semantic information [35]. To obtain high-quality word features from the low-resource dataset, character features learned from other language/domain may provide crucial information for labeling, especially for rare and out-of-vocabulary words. Character-level encoder usually contains BiLSTM [4] and CNN [5], [6] approaches. In practice, Reimers *et al.* [53] observed that the difference between the two approaches is statistically insignificant in the sequence labeling tasks, but character-level CNN is more efficient and has less parameters. Thus, we use character-level CNN and share character features between high- and low-resource tasks to enhance the representations of low-resource.

3.2.2 *Word-level Encoder*

To learn a better word-level representation, we concatenate character-level features of each word with a latent word embedding as $\mathbf{w}_i = [\mathbf{w}_i^{char}, \mathbf{w}_i^{emb}]$, where the latent word embedding $\mathbf{w}_i^{emb}$ is initialized with pre-trained embeddings and fixed during training. One characteristic of sequence labeling is that the historical and future input for a given time step could be useful for label inference. To exploit such a characteristic, we use a bidirectional LSTM architecture [54] to extract contextualized word-level features. In this way, we can gather the information from the past and future for a particular time frame t as follows, $\overrightarrow{\mathbf{h}}_t = \texttt{lstm}(\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{w}_t)$, $\overleftarrow{\mathbf{h}}_t = \texttt{lstm}(\overleftarrow{\mathbf{h}}_{t+1}, \mathbf{w}_t)$. After the LSTM layer, the representation of a word is obtained by concatenating its left and right context representation as follows, $\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t]$.

To consider the resource representation difference on word-level features, we introduce two transferable word-level encoders used in our model, namely DATNet-Full Share (DATNet-F) and DATNet-Part Share (DATNet-P). In DATNet-F, all the BiLSTM units are shared by all resources while word embeddings for different resources are disparate. The illustrative figure is depicted in the Fig. 1(c). Different from DATNet-F, the DATNet-P decomposes the BiLSTM units into the shared component and the resource-related one, which is shown in the Fig. 1(b).

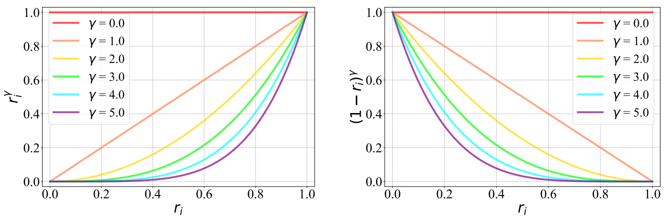
3.2.3 *Generalized Resource-Adversarial Discriminator*

To achieve the compatibility between the feature representation extracted from the source domain and those from the target domain, we encourage the outputs of the shared BiLSTM part to be resource-agnostic by constructing a resource-adversarial discriminator, which is inspired by the Language-Adversarial Discriminator proposed by [47]. Unfortunately, previous works did not consider the imbalanced training size of two resources. Specifically, the target domain consists of very few labeled training data, e.g., 10 sentences. In contrast, labeled training data in the source domain are much richer, e.g., 10k sentences. If such imbalance was not considered during training, the stochastic gradient descent (SGD) optimization would make the model biased to high resource [55]. To address this imbalance problem, we impose a weight α on two resources to balance their influences. However, in the experiment we also observe that the easily classified samples from high resource comprise the majority of the loss and dominate the gradient. To overcome this issue, we further propose Generalized Resource-Adversarial Discriminator (GRAD) to enable adaptive weights for each sample (note that the sample here means each sentence of resource), which enables the model training focusing on the hard samples.

In order to compute the loss of GRAD, the output sequences of shared BiLSTM should be first encoded into a single vector, i.e. sentence representation, and then fed into the discriminator. The common way to create the sentence representation is either using the final hidden state of BiLSTM or the max (or average) pooling from the hidden states. Unfortunately, most of these approaches fail to carry the semantics along the long sequences of a recurrent model [56]. To overcome this challenge, self-attention mechanism [57] could be a feasible solution that utilizes all the local information to construct the sentence representation. Specifically, it computes the weighted summation of all hidden states with different attention weights which indicate the contribution of each hidden state to the whole sentence representation. Then the single vector is projected into a scalar probability r via a linear transformation and activation. The loss function of the resource classifier is formulated as:

$$\begin{aligned}\ell_{GRAD} = -\sum_i \{ & \mathbf{I}_{i\in\mathcal{D}_S}\alpha(1-r_i)^\gamma \log r_i \\ & + \mathbf{I}_{i\in\mathcal{D}_T}(1-\alpha) r_i^\gamma \log(1-r_i)\}\end{aligned} \tag{1}$$

where $\mathbf{I}_{i\in\mathcal{D}_S}, \mathbf{I}_{i\in\mathcal{D}_T}$ are the identity functions to denote whether a sentence is from high resource (source) and low resource (target), respectively; α is a weighting factor to balance the loss contribution from high and low resource. The parameter $(1-r_i)^\gamma$ (or r_i^γ) controls the loss contribution from individual samples by measuring the discrepancy between prediction and true label (easy samples have smaller contribution). $\gamma \geq 0$ is a factor that smoothly adjusts the rate at which easy examples are down-weighted. Figure 2 illustrates how $(1-r_i)^\gamma$ (or r_i^γ) controls the loss contribution for individual samples. For example, for the sample from the high resource $\mathcal{D}_S$, its corresponding loss term is $\mathbf{I}_{i\in\mathcal{D}_S}\alpha(1-r_i)^\gamma \log r_i$, where the controlling factor $(1-r_i)^\gamma$ is inverse proportion to r_i. In other words, when $r_i \to 1$, this well-classified sample is down-weighted due to $(1-r_i)^\gamma$ goes to 0. As γ increases, the approaching speed increases. In this case, a large γ is preferred. For the sample from low resource data, a small γ is preferred. Therefore, the value of γ should be carefully selected to trade off the source and target data. In practice, $\gamma = 2$ always achieves the best performance as discussed in Section 4.5. The resource classifier is optimized by minimizing the resource classification error and the gradients originated from the loss are back-propagated to other model parts. Then the gradient reversal module negates the gradients for parameter updates such that the bottom layers are trained to be resource-agnostic.



(a) r_i vs r_i^γ (b) r_i vs $(1-r_i)^\gamma$

Fig. 2. The effect of γ on sample weights.

3.2.4 Label Decoder

The label decoder induces a probability distribution over the sequences of labels, conditioned on the word-level encoder features. In this paper, we use a linear chain model based on the first-order Markov chain structure, termed the chain conditional random field (CRF) [58], as the decoder. In this decoder, there are two kinds of cliques: local cliques and transition cliques. Specifically, local cliques correspond to the individual elements in the sequence. On the other hand, the transition cliques reflect the evolution of states between two neighboring elements at time $t-1$ and t and we define the transition distribution as θ. Formally, a linear-chain CRF can be written as $p(\mathbf{y}|\mathbf{h}_{1:T}) = \frac{1}{Z(\mathbf{h}_{1:T})}\exp\left\{\sum_{t=2}^{T}\theta_{y_{t-1},y_t} + \sum_{t=1}^{T}\mathbf{W}_{y_t}\mathbf{h}_t\right\}$, where $Z(\mathbf{h}_{1:T})$ is a normalization term and $\mathbf{y}$ is the sequence of predicted labels as follows: $\mathbf{y} = y_{1:T}$. The parameters of our model are optimized to maximize this conditional log likelihood, which acts as the objective function of the model. We define the loss function for the source and target resource as follows, $\ell_S = -\sum_i \log p(\mathbf{y}|\mathbf{h}_{1:T})$, $\ell_T = -\sum_i \log p(\mathbf{y}|\mathbf{h}_{1:T})$.

3.2.5 Adversarial Training

So far our model can be trained in an end-to-end manner with the standard back-propagation by minimizing the following loss:

$$\ell = \ell_{GRAD} + \ell_S + \ell_T. \tag{2}$$

Recent works have demonstrated that deep learning models are fragile to *adversarial examples* [50], [52], [59]. Thus, adversarial samples are widely incorporated into training to improve the generalization and robustness of the model, which is called adversarial training (AT) [51]. It emerges as a powerful regularization tool to stabilize training and enable the model to escape from the local minimum. Recently, Yasunaga *et al.* [52] applied adversarial training to POS tagging. In this paper, we also explore AT in more tasks of sequence labeling. To be specific, we prepare an adversarial sample by adding the original sample with a perturbation bounded by a small norm ϵ to maximize the loss function as follows:

$$\eta_{\mathbf{x}} = \arg \max_{\eta:\|\eta\|_2\leq\epsilon} \ell(\Theta; \mathbf{x}+\eta) \tag{3}$$

where Θ is the current model parameters set. However, we cannot calculate the value of η exactly in general, because the exact optimization with respect to η is intractable in neural networks. Following the strategy in [50], [52], this value can be approximated by linearizing it as follows,

$$\eta_{\mathbf{x}} = \epsilon\frac{\mathbf{g}}{\|\mathbf{g}\|_2}, \quad \text{where} \;\; \mathbf{g} = \nabla\ell(\Theta; \mathbf{x}) \tag{4}$$

where ϵ can be determined on the validation set. In this way, the adversarial examples are generated by adding small perturbations into the inputs along the direction that most significantly increases the loss function of the model. We find such η against the current model parameterized by Θ. Moreover, at each training step, we construct an adversarial example via $\mathbf{x}_{adv} = \mathbf{x} + \eta_{\mathbf{x}}$. Noted that we generate this adversarial example on the word and character embedding layer, respectively, as shown in the Fig. 1(b) and 1(c). Then, the classifier is trained on the mixture of original and adversarial examples to improve the generalization. To this end, we augment the loss in Eqn. 2 and define the loss function for adversarial training via:

$$\ell_{AT} = \ell(\Theta; \mathbf{x}) + \ell(\Theta; \mathbf{x}_{adv}) \tag{5}$$

where $\ell(\Theta; \mathbf{x}), \ell(\Theta; \mathbf{x}_{adv})$ represents the loss from an original example and its adversarial counterpart, respectively. Note that we present the AT in a general form for ease of presentation. For different samples, the loss and parameters should correspond to their counterparts. For example, for the source data with word embedding $\mathbf{w}_S$, the loss for AT can be defined as follows, $\ell_{AT} = \ell(\Theta; \mathbf{w}_S) + \ell(\Theta; \mathbf{w}_{S,adv})$ with $\mathbf{w}_{S,adv} = \mathbf{w}_S + \eta_{\mathbf{w}_S}$ and $\ell = \ell_{GRAD} + \ell_S$. Similarly, we can compute the perturbations $\eta_{\mathbf{c}}$ for char-embedding and $\eta_{\mathbf{w}_T}$ for target word embedding.

4 Experiments

In this section, we evaluate our DATNet method on various benchmark sequence labeling tasks to demonstrate the effectiveness comparing with other competing sequence labeling methods. Furthermore, we study the performance of DATNet under different transferring settings upon different datasets. Meanwhile, we also incorporate state-of-the-art language model into our method to boost the performance.

4.1 Datasets

For a comprehensive comparison, our experiment involves three different sequence labeling tasks, namely, part-of-speech (POS) tagging, chunking and named entity recognition (NER). We conduct experiments on CoNLL [60]–[62], WNUT [63], PTB-WSJ [64], OntoNotes [65] and Universal Dependencies (UD) [66] benchmark datasets as well as cross-lingual named tagging datasets for 282 languages (CLNER) [29]. The statistics of the benchmark datasets are described in Table 1. For CLNER datasets, we choose 9 different lingual datasets in our experiment, as summarized in Table 2. For UD datasets, we choose 21 different lingual datasets to compare with other methods.

Typically, we divide the aforementioned datasets into three groups for different experimental settings:

- CoNLL-2002 & 2003 and WNUT datasets are used to study the transferring performance from the source to the target by simulating various low-resource scenario. Specially, CoNLL-2003 dataset is used as the source, CoNLL-2002 datasets and WNUT datasets are used as the target in cross-language and cross-domain settings, respectively.
- CLNER contains the NER datasets in various language families and branches, which cover from the most major languages (like *English*) to minority languages (like *Marathi*). We conduct experiments on these datasets to investigate the transfer ability among different linguistic families and branches of our method under low- and high-resource scenarios. The UD dataset is also utilized in the experiments for the part-of-speech tagging.
- CoNLL-2000 & 2003, CoNLL-2002 Dutch, PTB-WSJ, OntoNotes and WNUT-2017 datasets are used to study the performance of incorporating language model into DATNets. Specially, we choose the latest pre-trained models ELMo [39] and BERT [40] for evaluation.

Unlike previous works [2], [5], [13], [33], [67], [68] that introduced hand-crafted features (e.g. one-hot gazetteer and orthographic features) as the additional input for further boosting performance, we do *not* adopt these hand-crafted features and only use words and characters as inputs for our method.

4.2 Experimental Setup

We use 50-dimensional publicly available pre-trained word embeddings for English, Spanish and Dutch languages of CoNLL and WNUT datasets in our experiments. The models are trained by word2vec package on the corresponding Wikipedia articles (2017-12-20 dumps) [35]. For the named entity datasets [29], we use 300-dimensional pre-trained word embeddings trained by fastText package on Wikipedia [69]. For the part-of-speech datasets from UD, we use 64-dimensional pre-trained Polyglot word embeddings [70] for fair comparison. The 30-dimensional randomly initialized character embeddings are used for all the datasets. We set the filter number as 20 for char-level CNN and the dimension of hidden states of the word-level LSTM as 200 for both base model and DATNet-F. For DATNet-P, we set the dimension for the source, share, and target LSTMs to 100. Parameters optimization is performed by Adam optimizer [71] with the gradient clipping of 5.0 and learning rate decay strategy.

We set the initial learning rate by $\beta_0 = 0.001$ for all experiments. At each epoch t, β_t is updated by $\beta_t = \beta_0/(1 + \rho \times t)$, where ρ is the decay rate with 0.05. To reduce over-fitting, we also apply dropout [72] to the embedding layer and the output of the LSTM layer, respectively.

4.3 Comparison with State-of-the-Art Results

In this section, we compare our approach with state-of-the-art (SOTA) methods on a set of benchmark datasets. We first carry out experiment on the CoNLL and WNUT datasets. In the experiment, we exploit all the source data (i.e., CoNLL-2003 English NER) and the target data to improve performance of a specific task. The averaged results with the standard deviation over 10 repetitive runs are summarized in Table 3, and we also report the best results on each task for fair comparison with other SOTA methods. From the results, one could observe that incorporating the additional resource is helpful to improve performance. DATNet-P model achieves the highest F1 score on CoNLL-2002 Spanish and second F1 score on CoNLL-2002 Dutch dataset. Moreover, DATNet-F model beats the other methods on WNUT datasets. Different from other state-of-the-art models, DATNets do *not* use any addition features[1].

Table 4 summarizes the results of our methods under different cross-language transfer settings and shows the comparison with Cotterell *et al.* [28]. In this experiment, we study the transferability between languages not only from the same linguistic family and branch, but also from different linguistic families or branches. According to the results, DATNets outperform the transfer learning method of Cotterell *et al.* [28] for both low- and high-resource scenarios within the same linguistic family and branch (i.e., in-family in-branch) transfer case. One could observe that:

1) For the low-resource scenario, transfer learning significantly improves the performance of target datasets within both the same and different linguistic family or branch (i.e., in/cross-family in/cross-branch), while the improvements are more prominent for the in-family in-branch case.
2) For the high-resource scenario, when the target language data is sufficient, the improvement of transfer learning is not as distinct as that for low-resource scenario under the in-family in-branch case. We also find little improvement by transferring knowledge from *Arabic* to either of *Galician* and *Ukrainian*. Since *Arabic* and *Galician* are from totally different linguistic families, the improvement may be limited by the great linguistic differences between the source and target languages.

The experiment results on the UD multilingual part-of-speech tagging datasets are given in Table 5. Our DATNets show distinct improvements over the base model on all the 21 languages, which outperforms the state-of-the-art methods on 16 different languages. The overall performance of DATNet-F is comparable to Yasunaga *et al.* [52], which achieves the SOTA performance on the UD datasets. Moreover, DATNet-P achieves 0.21% absolute improvement on the average accuracy. All the experiments in Table 5 follow the cross-language transferring setting, and one could observe that DATNet-P method always outperforms DATNet-F method. The superior performance of DATNet-P may result from language-specific features in the model architecture. In contrast, DATNet-F only learns the language-agnostic features from both source and target languages.

[1] It is not sure whether Feng *et al.* [21] has incorporated the validation set into training. And if we merge training and validation sets, we can push the F1 score to **88.71**. In addition to the aforementioned features, Aguilar *et al.* [73] also incorporated the International Phonetic Alphabet (IPA), phonological features, and subword information to handle noisy text and out-of-vocabulary (OOV) words.

TABLE 1
Statistics of Benchmark Sequence Labeling Datasets.

Benchmark	Task	Language	# Training Tokens (# Entities)	# Dev Tokens (# Entities)	# Test Tokens (# Entities)
PTB-WSJ	POS	English	912,344	131,768	129,654
OntoNotes	POS	English	1,088,503	147,724	152,728
OntoNotes	NER	English	1,088,503 (81,828)	147,724 (11,066)	152,728 (11,257)
WNUT-2016	NER	English	46,469 (2,462)	16,261 (1,128)	61,908 (5,955)
WNUT-2017	NER	English	62,730 (3,160)	15,733 (1,250)	23,394 (1,740)
CoNLL-2000	Chunking	English	211,727	-	47,377
CoNLL-2002	NER	Spanish	207,484 (18,797)	51,645 (4,351)	52,098 (3,558)
CoNLL-2002	NER	Dutch	202,931 (13,344)	37,761 (2,616)	68,994 (3,941)
CoNLL-2003	NER	English	204,567 (23,499)	51,578 (5,942)	46,666 (5,648)

TABLE 2
Statistics of Selected Named Entity Recognition Datasets from Pan *et al.* [29].

Language	Resource	Linguistic Family	Linguistic Branch	# Training Sentences	# Dev Sentences	# Test Sentences
Spanish (es)	Source	Indo-European	Romance	10,000	-	-
Galician (gl / gl-h)	Target	Indo-European	Romance	100 / 10,000	1,000	1,000
Dutch (nl)	Source	Indo-European	Germanic	10,000	-	-
West Frisian (fy)	Target	Indo-European	Germanic	100	1,000	1,000
Russian (ru)	Source	Indo-European	Slavic	10,000	-	-
Ukrainian (uk / uk-h)	Target	Indo-European	Slavic	100 / 10,000	1,000	1,000
Hindi (hi)	Source	Indo-European	Indo-Aryan	10,000	-	-
Marathi (mr)	Target	Indo-European	Indo-Aryan	100	1,000	1,000
Arabic (ar)	Source	Afro-Asiatic	Semitic	10,000	-	-

gl-h and uk-h denote the high-resource settings for Galician and Ukrainian respectively.

TABLE 3
Comparison with State-of-the-art Results in CoNLL and WNUT datasets (F1-score).

Mode	Methods		Additional Features			CoNLL Datasets		WNUT Datasets	
			POS	Gazetteers	Orthographic	Spanish	Dutch	WNUT-2016	WNUT-2017
Mono-language /domain	Gillick *et al.* [74]		×	×	×	82.59	82.84	-	-
	Lample *et al.* [4]		×	√	×	85.75	81.74	41.77*	34.53*
	Partalas *et al.* [67]		√	√	√	-	-	46.16	-
	Limsopatham *et al.* [68]		×	×	√	-	-	52.41	-
	Lin *et al.* [75]		√	√	×	-	-	-	40.42
	Our Base Model	Best	×	×	×	85.53	85.55	44.96	35.20
		Mean & Std				85.35±0.15	85.24±0.21	44.37±0.31	34.67±0.34
Cross-language /domain	Yang *et al.* [13]		×	√	×	85.77	85.19	47.19*	40.83*
	Ying *et al.* [35]		×	√	×	85.88	86.55	46.53*	40.79*
	Feng *et al.* [21]		√	×	×	86.42	**88.39**	-	-
	Von *et al.* [76]		×	√	×	-	-	-	40.78
	Aguilar *et al.* [33]		√	×	√	-	-	-	41.86
	Aguilar *et al.* [73]		√	√	√	-	-	-	**45.55**
	DATNet-P	Best	×	×	×	**88.16**	88.32	50.85	41.12
		Mean & Std				<u>87.89±0.18</u>	88.09±0.13	50.41±0.32	40.52±0.38
	DATNet-F	Best	×	×	×	87.04	87.77	**53.43**	42.83
		Mean & Std				86.79±0.20	87.52±0.19	<u>53.03±0.24</u>	<u>42.32±0.32</u>

The scores with "*" denote produced results by the corresponding official tools/codes.

4.4 Transfer Learning Performance

In this section, we investigate the improvements with transfer learning under multiple low-resource settings on partial target data. To simulate a low-resource setting, we randomly obtain some subsets of target data with varying data ratio at 0.05, 0.1, 0.2, 0.4, 0.6, and 1.0. For example, $20,748$ training tokens are sampled from the training set under a data ratio of $r = 0.1$ for the dataset CoNLL-2002 Spanish NER (Cf. Table 1). The results for cross-language and cross-domain transfer are shown in Fig. 3(a) and 3(b), respectively, where we compare the results with each part of DATNet under various data ratios. From those figures, we have the following observations:

1) Both adversarial training and adversarial discriminator in DATNet consistently contribute to the performance improvement;
2) The transfer learning component in the DATNet consistently improves over the results of the base model and the improvement margin is more distinct when the target data ratio is lower.

Specifically, when the data ratio is 0.05, DATNet-P model outperforms the base model by more than 4% in F1-score on Spanish NER. Furthermore, DATNet-F model improves around 13% absolutely in F1-score compared to base model on WNUT-2016 NER.

In the second experiment, we further investigate DATNet on the extremely low resource cases, e.g., the number of training

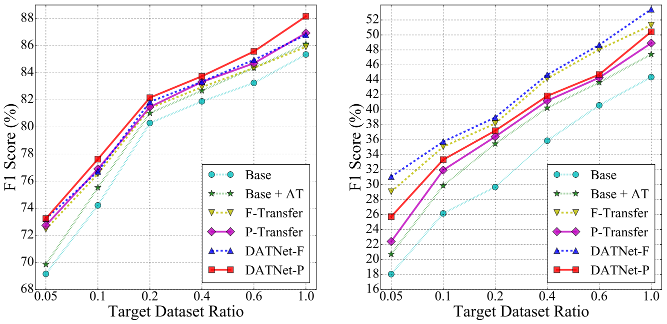
TABLE 4
Results of Varying Cross-language Transfer Settings in [29] Datasets (F1-Score).

Language		Transfer Strategy	Cotterell *et al.* [28]		Our Methods		
Source	Target		Base Model	Transfer	Base Model	DATNet-P	DATNet-F
Dutch (nl)	West Frisian (fy)	In-Family & In-Branch	58.43	72.12	57.47	75.08	76.05
Hindi (hi)	West Frisian (fy)	In-Family & Cross-Branch	-	-	57.47	69.25	68.44
Arabic (ar)	West Frisian (fy)	Cross-Family & Cross-Branch	-	-	57.47	67.89	66.05
Hindi (hi)	Marathi (mr)	In-Family & In-Branch	39.02	60.92	43.55	68.55	64.87
Dutch (nl)	Marathi (mr)	In-Family & Cross-Branch	-	-	43.55	63.83	60.50
Arabic (ar)	Marathi (mr)	Cross-Family & Cross-Branch	-	-	43.55	63.28	59.76
Spanish (es)	Galician (gl)	In-Family & In-Branch	49.19	76.40	49.94	79.60	86.01
Hindi (hi)	Galician (gl)	In-Family & Cross-Branch	-	-	49.94	60.57	61.68
Arabic (ar)	Galician (gl)	Cross-Family & Cross-Branch	-	-	49.94	59.18	60.43
Spanish (es)	Galician (gl-h)	In-Family & In-Branch	89.42	89.46	92.78	93.14	93.02
Arabic (ar)	Galician (gl-h)	Cross-Family & Cross-Branch	-	-	92.78	92.63	92.21
Russian (ru)	Ukrainian (uk)	In-Family & In-Branch	60.65	76.74	61.48	79.02	80.76
Hindi (hi)	Ukrainian (uk)	In-Family & Cross-Branch	-	-	61.48	72.73	73.84
Arabic (ar)	Ukrainian (uk)	Cross-Family & Cross-Branch	-	-	61.48	71.55	72.24
Russian (ru)	Ukrainian (uk-h)	In-Family & In-Branch	87.39	87.42	93.29	93.62	93.51
Arabic (ar)	Ukrainian (uk-h)	Cross-Family & Cross-Branch	-	-	93.29	92.83	92.42

* Base model denotes the model is trained by using target language dataset only.

TABLE 5
Comparison with State-of-the-art Results in Universal Dependencies (UD) Part-of-speech Tagging Datasets (%).

Language		Berend *et al.* [77]	Plank *et al.* [78]	Nguyen *et al.* [79]	Yasunaga *et al.* [52]	Base	DATNet-P	DATNet-F
bg	Bulgarian	95.63	97.97	97.4	98.53	98.34	**98.59**	98.47
cs	Czech	95.83	97.89	-	98.81	97.75	**98.87**	98.75
da	Danish	93.32	96.35	95.8	96.74	96.38	**96.82**	96.46
de	German	90.73	93.38	92.7	94.35	93.48	**94.58**	94.35
en	English	93.47	95.16	94.7	95.82	95.42	**95.94**	95.69
es	Spanish	94.69	95.74	95.9	96.44	96.41	**96.80**	96.64
eu	Basque	90.63	**95.51**	93.7	94.71	94.47	95.15	94.92
fa	Persian	96.11	97.49	96.8	97.51	97.00	**97.64**	97.38
fi	Finnish	89.19	95.85	94.6	95.40	95.27	**95.89**	95.78
fr	French	94.96	96.11	96.0	96.63	96.52	**97.34**	97.15
he	Hebrew	95.28	96.96	-	97.43	97.22	**97.46**	97.32
hi	Hindi	96.09	97.10	96.4	97.21	96.92	**97.25**	97.11
hr	Croatian	93.53	96.82	-	96.32	96.56	**97.18**	97.12
id	Indonesian	92.02	93.41	93.1	**94.03**	93.68	93.89	93.75
it	Italian	96.28	97.95	97.5	98.08	98.02	**98.23**	98.10
nl	Dutch	85.10	**93.30**	91.4	93.09	92.68	93.04	92.85
no	Norwegian	95.67	98.03	97.4	**98.08**	97.57	98.01	97.77
pl	Polish	93.95	97.62	96.3	97.57	97.35	**97.95**	97.80
pt	Portuguese	95.50	97.90	97.5	**98.07**	97.60	97.92	97.71
sl	Slovenian	92.70	96.84	97.1	98.11	97.50	**98.17**	98.05
sv	Swedish	94.62	96.69	-	96.70	96.45	**97.31**	97.13
Avg. accuracy		93.59	96.40	95.55	96.65	96.31	**96.86**	96.68



(a) CoNLL-2002 Spanish NER (b) WNUT-2016 Twitter NER

Fig. 3. Comparison with Different Target Data Ratio, where AT stands for adversarial training, F(P)-Transfer denotes the DATNet-F(P) without AT.

target sentences is 10, 50, 100, 200, 500 or 1,000. The setting is quite challenging and fewer works have studied before. The results are summarized in Table 6. We have two interesting observations[2]:

1) DATNet-F outperforms DATNet-P on cross-language transfer when the target resource is extremely low, however, this situation is reversed when the target dataset size is large enough (i.e., more than 100 sentences);
2) DATNet-F is generally superior to DATNet-P on cross-domain transfer.

The factor for the first observation may be because DATNet-F with more shared hidden units is more efficient than DATNet-P to transfer knowledge when the data size is extremely small. For the second observation, the possible reason is that the cross-domain

[2] For other tasks/languages we have the similar observation, so we only report CoNLL-2002 Spanish and WNUT-2016 Twitter results.

transfer are in the same language, more knowledge is common between the source and target domains, requiring more shared hidden features to carry with these knowledge compared to cross-language transfer. Therefore, for cross-language transfer with an extremely low resource and cross-domain transfer, we suggest using DATNet-F model for better performance. As for the cross-language transfer with relatively more training data, DATNet-P model shows better result.

TABLE 6
Experiments on Extremely Low Resource (F1-score).

Tasks	CoNLL-2002 Spanish NER					
# Target sentences	10	50	100	200	500	1000
Base	21.53	42.18	48.35	63.66	68.83	76.69
+ AT	19.23	41.01	50.46	64.83	70.85	77.91
+ P-Transfer	29.78	61.09	64.78	66.54	72.94	78.49
+ F-Transfer	39.72	63.00	63.36	66.39	72.88	78.04
DATNet-P	39.52	62.57	64.05	**68.95**	**75.19**	**79.46**
DATNet-F	**44.52**	**63.89**	**66.67**	68.35	74.24	78.56
Tasks	WNUT-2016 Twitter NER					
# Target sentences	10	50	100	200	500	1000
Base	3.80	14.07	17.99	26.20	31.78	36.99
+ AT	4.34	16.87	18.43	26.32	35.68	41.69
+ P-Transfer	7.71	16.17	20.43	29.20	34.90	41.20
+ F-Transfer	15.26	20.04	26.60	32.22	38.35	44.81
DATNet-P	9.94	17.09	25.39	30.71	36.05	42.30
DATNet-F	**17.14**	**22.59**	**28.41**	**32.48**	**39.20**	**45.25**

4.5 Ablation Study of DATNet

In the proposed DATNet, both GRAD and AT play important roles in low resource NER. In this experiment, we further investigate how GRAD and AT help transfer knowledge across different languages/domains. In the first experiment[3], we used t-SNE [80] to visualize the feature distribution of BiLSTM outputs without AD, with the normal AD (GRAD without considering data imbalance) and the proposed GRAD in Figure 4. From the figure, one could see that the GRAD in DATNet makes the distribution of extracted features from the source and target datasets much more similar by considering the data imbalance, which indicates that the outputs of BiLSTM are resource-invariant.

To better understand the working mechanism, Table 7 further reports the quantitative performance comparison between models with different components. We observe that GRAD shows the stable superiority over the normal AD regardless of other components. There are no always winner between DATNet-P and DATNet-F on different settings. The DATNet-P architecture is more suitable to cross-language transfer whereas DATNet-F is more suitable to cross-domain transfer.

TABLE 7
Quantitative Performance Comparison between Models with Different Components.

Model	F1-score	Model	F1-score
CoNLL-2002 Spanish NER			
Base	85.35	+AT	86.12
+P-T (no AD)	86.15	+AT +P-T (no AD)	86.90
+F-T (no AD)	85.46	+AT +F-T (no AD)	86.17
+P-T (AD)	86.32	+AT +P-T (AD)	87.19
+F-T (AD)	85.58	+AT +F-T (AD)	86.38
+P-T (GRAD)	86.93	**DATNet-P**	**88.16**
+F-T (GRAD)	85.91	**DATNet-F**	87.04
WNUT-2016 Twitter NER			
Base	44.37	+AT	47.41
+P-T (no AD)	47.66	+AT +P-T (no AD)	48.44
+F-T (no AD)	49.79	+AT +F-T (no AD)	50.93
+P-T (AD)	48.14	+AT +P-T (AD)	49.41
+F-T (AD)	50.48	+AT +F-T (AD)	51.84
+P-T (GRAD)	48.91	**DATNet-P**	50.85
+F-T (GRAD)	51.31	**DATNet-F**	**53.43**

* AT: Adversarial Training; P-T: P-Transfer; F-T: F-Transfer; AD: Adversarial Discriminator; GRAD: Generalized Resource-Adversarial Discriminator.

Moreover, adversarial training (AT) acts as a regularizer, which is related to other regularization methods that add noise to data such as dropout and its variants. Dropout is data-independent and it randomly mutes neurons with a certain probability to reduce overfitting. In contrast, AT is a data-driven regularization, which generates adversarial examples by adding perturbations to the inputs in the direction that most significantly increases the loss. In this way, AT forces the model to adapt the noise and improves its generalization ability. To further illustrate the superiority of AT, we also compare AT with dropout by adding each of them into the word/char embedding layers of base model, respectively. The results are summarized in Table 8, which demonstrate that AT is more effective regularization and it always outperforms the dropout.

TABLE 8
Comparison between AT and Dropout Regularizer.

Method	CoNLL-2002 NER		WNUT NER	
	Spanish	Dutch	2016	2017
Base	85.35	85.55	44.37	34.67
Base + dropout	85.51	85.84	45.95	36.72
Base + AT	**86.12**	**86.76**	**47.41**	**38.48**

TABLE 9
Analysis of Maximum Perturbation $\epsilon_{\mathbf{w}_T}$ in AT with Varying Data Ratio ρ (F1-score).

$\epsilon_{\mathbf{w}_T}$	1.0	3.0	5.0	7.0	9.0
Ratio	CoNLL-2002 Spanish NER				
$\rho = 0.1$	75.90	76.23	77.38	77.77	**78.13**
$\rho = 0.2$	81.54	81.65	81.32	**81.81**	81.68
$\rho = 0.4$	83.62	83.83	83.43	**83.99**	83.40
$\rho = 0.6$	84.44	84.47	**84.72**	84.04	84.05

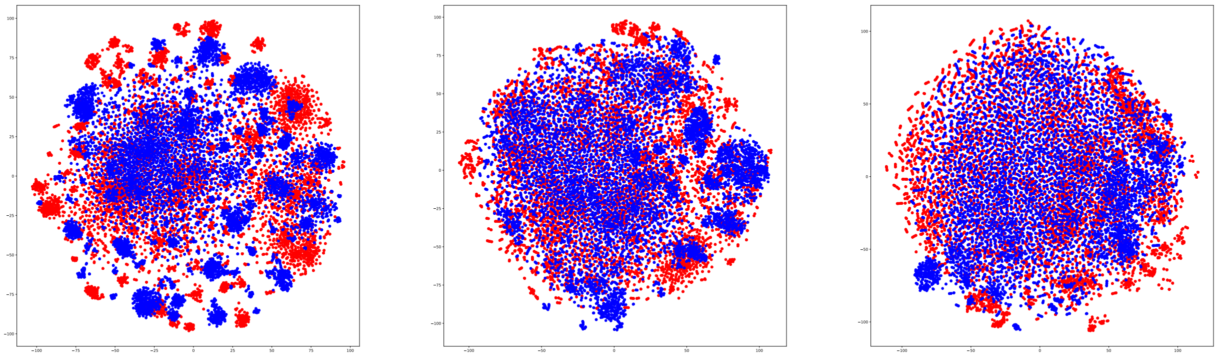
From the aforementioned results, one could know that AT helps to enhance the overall performance by adding perturbations into inputs with the limit of $\epsilon = 5$, i.e., $\|\eta\|_2 \leq 5$. In this experiment, we further investigate how the target perturbation $\epsilon_{\mathbf{w}_T}$ with the fixed source perturbation $\epsilon_{\mathbf{w}_S} = 5$ in AT affects knowledge transfer and the results on Spanish NER are summarized in Table 9. The results generally indicate that less training data require a larger ϵ to prevent over-fitting, which further validates the necessity of AT in the case of low resource data.

Figure 5 shows the mean and standard deviation score of both DATNet-F and DATNet-P on CoNLL-2002 Spanish NER dataset with different γ. One could observe that both DATNet-F and DATNet-P achieve the best results when $\gamma = 2$. The experiments on the other used datasets also support this observation, and thus we give the recommendation of $\gamma = 2$ in practical use.

Finally, the discriminator weight α in GRAD and results are summarized in Table 10. From the results, one could find that α is

[3] We used data ratio $\rho = 0.5$ for training model and randomly selected 10k testing data for visualization.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2931569, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) no AD

(b) AD

(c) GRAD

Fig. 4. The visualization of extracted features from shared bidirectional-LSTM layer. The left, middle, and right figures show the results when no Adversarial Discriminator (AD), AD, and GRAD is performed, respectively. Red points correspond to the source CoNLL-2003 English examples, and blue points correspond to the target CoNLL-2002 Spanish examples.

TABLE 10
Analysis of Discriminator Weight α in GRAD with Varying Data Ratio ρ (F1-score).

α	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8
Ratio	CoNLL-2002 Spanish NER														
$\rho = 0.1$	78.37	78.63	**78.70**	78.32	77.96	77.92	77.88	77.78	77.85	77.90	77.65	77.57	77.38	77.49	77.29
$\rho = 0.2$	80.99	81.71	**82.18**	81.57	81.53	81.55	81.44	81.25	81.32	81.16	81.02	81.16	80.63	80.79	80.54
$\rho = 0.4$	83.76	83.73	84.18	**84.48**	84.26	84.12	83.54	83.40	83.52	84.18	83.42	83.47	83.28	83.33	83.19
$\rho = 0.6$	85.18	85.24	85.85	85.68	85.84	**86.10**	85.71	85.74	85.42	85.60	85.20	85.40	85.26	85.24	84.98

TABLE 11
Comparison with State-of-the-art Language Models (%).

Method	Chunking	Named Entity Recognition				POS Tagging	
	CoNLL-2000	CoNLL-2003	OntoNotes	WNUT-2017	CoNLL-2002 Dutch	OntoNotes	PTB-WSJ
Liu *et al.* [19]	96.13	91.85	87.89*	39.61*	86.03*	97.66*	97.59
Peters *et al.* [37]	96.37	91.93	-	-	-	-	-
Peters *et al.* [39]	96.41*	92.22	88.25*	42.58*	-	97.74*	97.68*
Devlin *et al.* [40]	96.66*	92.40	88.76*	46.88*	89.55*	98.11*	97.69*
Akbik *et al.* [41]	96.72	**93.18**	89.30	50.24	90.44	-	97.85
Base	94.71	91.23	87.75	35.20	85.55	97.61	97.49
Base + ELMo	96.50	92.21	88.80	43.93	-	97.83	97.70
Base + BERT	96.68	91.90	88.83	47.05	89.73	98.13	97.69
DATNet-P	95.70	91.63	88.15	41.12	88.32	97.69	97.57
DATNet-F	96.10	92.16	88.33	42.83	87.77	97.79	97.68
DATNet-P + ELMo	96.64	92.48	88.96	47.91	-	97.85	97.74
DATNet-F + ELMo	96.96	92.88	89.24	49.58	-	98.03	97.86
DATNet-P + BERT	96.93	92.45	**89.56**	50.25	**91.20**	98.25	97.74
DATNet-F + BERT	**97.01**	92.63	89.39	**50.63**	90.91	**98.28**	**97.88**

The scores with "*" denote produced results by the corresponding official tools/codes. Results with BERT-base in [40] are reported for fair comparison.

directly proportional to the data ratio ρ. In other words, more target training data requires larger α to achieve better performance, i.e., smaller $1 - \alpha$ to reduce training emphasis on the target domain.

4.6 Incorporate Language Model into DATNet

In this section, we further augment DATNets with pre-trained language models and conduct experiments on six English sequence labeling tasks and one Dutch NER task. Different from the preceding experiments [4], [21], [35], [74] which directly use the pre-trained word embeddings as the input, we use pre-trained

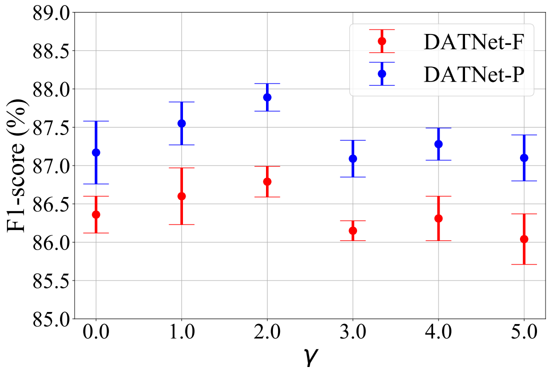


Fig. 5. Analysis of γ in GRAD on CoNLL-2002 Spanish NER.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2931569, IEEE Transactions on Pattern Analysis and Machine Intelligence

ELMo [39] and BERT [40] as the feature encoder[4]. Since BERT could learn the contextual information from input sequence well, we replace the bi-LSTM either of base model and DATNets with feed-forward layer. Moreover, we replace the top CRF decoder with a classifier layer, i.e., to achieve the consistency with the BERT settings. Here we first feed the input sequences into the language model to generate contextualized feature representations and then use those features as the input of our models. We show that our approach is able to beat or be comparable to a series of SOTA methods[5]. The experimental results are summarized in Table 11.

Since only the pre-trained English ELMo model is available, we use it on six English sequence labeling tasks except Dutch dataset. From Table 11, we observe that there are 1.79%, 0.98% and 1.05% improvements with ELMo for base model on CoNLL-2000 Chunk, CoNLL-2003 NER and OntoNotes NER, respectively. It is interesting to note that ELMo improves the performance of base model on WNUT-2017 NER by a large performance margin, i.e., increases of 8.73% in F1-score. The performance improvement on DATNets are also significant, namely 0.86% on CoNLL-2000 Chunk, 0.72% on CoNLL-2003, 0.91% on OntoNotes NER and 6.75% on WNUT-2017 NER.

Comparing with ELMo, BERT appears to a more powerful tool for natural language processing tasks. More specifically, BERT has a more complex and deeper model structure, which is trained on larger language corpus. In this experiment, we use BERT for all the English and Dutch datasets. As shown in Table 11, with the help of pre-trained BERT, DATNets further advances the state-of-the-art performance on the sequence labeling tasks and show distinct improvements over Akbik *et al.* [41]. BERT is generally superior to ELMo except CoNLL-2003 dataset. For example, DATNet-F+BERT achieves 50.63% on WNUT-2017 NER, which is with the improvements of 1.05% over DATNet-F+ELMo, and DATNet-P+BERT obtains 91.20% on CoNLL-2002 Dutch NER. Note that the reported results of BERT-base on CoNLL-2003 English NER is 92.40% in [40]. Unfortunately, our reproduced result of BERT base model is only 91.63%, which is slightly lower than the reported results. Meanwhile, we notice a very recent work [81] that is also fine-tuned the BERT on the CoNLL-2003 NER task and only 91.07% F1-score could be achieved. Therefore, we assume that some practical tricks used in the original paper may not be publicly released or the environment difference may lead to the performance difference.

From Table 11, one could also observe that DATNet-F approach is generally more suitable to the cross-domain transfer while DATNet-P prefers the cross-language transfer after introducing the language model, which is also aligned with the discussion in Section 4.5.

Generally, we see that with the help of language models, i.e., ELMo and BERT, significant improvements could be achieved for both base and DATNet models on different tasks. Such an improvement also relies on the knowledge transferred from external very large-scale language corpus. The result again supports the effectiveness of transfer learning in solving low-resource sequence labeling tasks. It also explains why the improvement of using language model on DATNets is slightly smaller than that of base model not augmenting language models. Nevertheless, DATNet is a general framework that can be adapted with existing language models for further improvement.

5 Conclusion

In this paper we develop a transfer learning model DATNet for sequence labeling tasks, which aims at addressing two problems remained in existing works, namely representation difference and resource data imbalance. To be specific, we introduce two variants of DATNet, DATNet-F and DATNet-P, which can be chosen for use according to the cross-language/domain user case and the target dataset size. To improve the model generalization, we propose dual adversarial learning strategies, i.e., AT and GRAD. Extensive experiments show the superiority of DATNet over existing models and our method achieves significant improvements on the benchmark datasets. By incorporating language model, DATNet further advances the state-of-the-art performance on several challenging tasks.

References

[1] T. Yue and H. Wang, “Deep learning for genomics: A concise overview,” *CoRR*, vol. abs/1802.00810, 2018.

[2] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *JMLR*, pp. 2493–2537, 2011.

[3] Z. Huang, W. Xu, and K. Yu, “Bidirectional lstm-crf models for sequence tagging,” *CoRR*, vol. abs/1508.01991, 2015.

[4] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” in *NAACL HLT*, 2016, pp. 260–270.

[5] J. 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] X. Ma and E. Hovy, “End-to-end sequence labeling via bi-directional lstm-cnns-crf,” in *ACL*, 2016, pp. 1064–1074.

[7] J. T. Zhou, H. Zhang, D. Jin, X. Peng, Y. Xiao, and Z. Cao, “Roseq: Robust sequence labeling,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2019.

[8] B. Zhang, X. Pan, T. Wang, A. Vaswani, H. Ji, K. Knight, and D. Marcu, “Name tagging for low-resource incident languages based on expectation-driven learning,” in *NAACL HLT*, 2016, pp. 249–259.

[9] Y. Chen, C. Zong, and K.-Y. Su, “On jointly recognizing and aligning bilingual named entities,” in *ACL*, 2010, pp. 631–639.

[10] Q. Li, H. Li, H. Ji, W. Wang, J. Zheng, and F. Huang, “Joint bilingual name tagging for parallel corpora,” in *CIKM*, 2012, pp. 1727–1731.

[11] M. Fang and T. Cohn, “Model transfer for tagging low-resource languages using a bilingual dictionary,” in *ACL*, 2017, pp. 587–593.

[12] O. Adams, A. Makarucha, G. Neubig, S. Bird, and T. Cohn, “Cross-lingual word embeddings for low-resource language modeling,” in *EACL*, 2017, pp. 937–947.

[13] Z. Yang, R. Salakhutdinov, and W. W. Cohen, “Transfer learning for sequence tagging with hierarchical recurrent networks,” in *ICLR*, 2017.

[14] L. Ratinov and D. Roth, “Design challenges and misconceptions in named entity recognition,” in *CoNLL*, 2009, pp. 147–155.

[15] A. Passos, V. Kumar, and A. McCallum, “Lexicon infused phrase embeddings for named entity resolution,” *CoRR*, vol. abs/1404.5367.

[16] G. Luo, X. Huang, C.-Y. Lin, and Z. Nie, “Joint entity recognition and disambiguation,” in *EMNLP*, 2015, pp. 879–888.

[17] W. Ling, C. Dyer, A. W. Black, I. Trancoso, R. Fermandez, S. Amir, L. Marujo, and T. Luís, “Finding function in form: Compositional character models for open vocabulary word representation,” in *EMNLP*, 2015, pp. 1520–1530.

[18] A. Zukov Gregoric, Y. Bachrach, and S. Coope, “Named entity recognition with parallel recurrent neural networks,” in *ACL*, 2018, pp. 69–74.

[19] L. Liu, J. Shang, F. Xu, X. Ren, H. Gui, J. Peng, and J. Han, “Empower sequence labeling with task-aware neural language model,” in *AAAI*, 2018.

[4] The BERT-base model is used in our experiments. The pre-trained English model and pre-trained multilingual model are used for the English sequence labeling tasks and Dutch named entity recognition task, respectively.

[5] For the experiments on CoNLL-2002 and WNUT-2017, we use CoNLL-2003 as the source data; For the experiments on PTB-WSJ and CoNLL-2003, OntoNotes is used as the source data; For the experiments on OntoNotes and CoNLL-2000, PTB-WSJ is as the source data.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2931569, IEEE Transactions on Pattern Analysis and Machine Intelligence

[20] D. Yarowsky, G. Ngai, and R. Wicentowski, "Inducing multilingual text analysis tools via robust projection across aligned corpora," in *HLT '01*, 2001, pp. 1–8.

[21] X. Feng, X. Feng, B. Qin, Z. Feng, and T. Liu, "Improving low resource named entity recognition using cross-lingual knowledge transfer," in *IJCAI*, 2018, pp. 4071–4077.

[22] J. T. Zhou, H. Zhao, X. Peng, M. Fang, Z. Qin, and R. S. M. Goh, "Transfer hashing: From shallow to deep," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, p. 61916201, 2018.

[23] M. Rei and A. Søgaard, "Zero-shot sequence labeling: Transferring knowledge from sentences to tokens," in *NAACL HLT*, 2018.

[24] J. Ni and R. Florian, "Improving multilingual named entity recognition with wikipedia entity type mapping," in *EMNLP*, 2016, pp. 1275–1284.

[25] J. Ni, G. Dinu, and R. Florian, "Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection," in *ACL*, 2017, pp. 1470–1480.

[26] R. Al-Rfou', V. Kulkarni, B. Perozzi, and S. Skiena, "Polyglot-ner: Massive multilingual named entity recognition," in *SDM*, 2015.

[27] S. Mayhew, C.-T. Tsai, and D. Roth, "Cheap translation for cross-lingual named entity recognition," in *EMNLP*, 2017, pp. 2536–2545.

[28] R. Cotterell and K. Duh, "Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields," in *IJCNLP*, 2017, pp. 91–96.

[29] X. Pan, B. Zhang, J. May, J. Nothman, K. Knight, and H. Ji, "Cross-lingual name tagging and linking for 282 languages," in *ACL*, 2017.

[30] Z. Yang, R. Salakhutdinov, and W. W. Cohen, "Multi-task cross-lingual sequence tagging from scratch," *CoRR*, vol. abs/1603.06270, 2016.

[31] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," in *ICLR*, 2016.

[32] M. Rei, "Semi-supervised multitask learning for sequence labeling," in *ACL*, 2017, pp. 2121–2130.

[33] G. Aguilar, S. Maharjan, A. P. López Monroy, and T. Solorio, "A multi-task approach for named entity recognition in social media data," in *WNUT*, 2017.

[34] K. Hashimoto, c. xiong, Y. Tsuruoka, and R. Socher, "A joint many-task model: Growing a neural network for multiple nlp tasks," in *EMNLP*, 2017, pp. 1923–1933.

[35] Y. Lin, S. Yang, V. Stoyanov, and H. Ji, "A multi-lingual multi-task architecture for low-resource sequence labeling," in *ACL*, 2018.

[36] J. T. Zhou, I. W. Tsang, S. J. Pan, and M. Tan, "Multi-class heterogeneous domain adaptation," *Journal of Machine Learning Research*, vol. 20, no. 57, pp. 1–31, 2019.

[37] M. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," in *ACL*, 2017.

[38] B. Bohnet, R. McDonald, G. Simões, D. Andor, E. Pitler, and J. Maynez, "Morphosyntactic tagging with a meta-bilstm model over context sensitive token encodings," in *ACL*, 2018, pp. 2642–2652.

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

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[41] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *COLING*, 2018, pp. 1638–1649.

[42] J. T. Zhou, M. Fang, H. Zhang, C. Gong, X. Peng, Z. Cao, and R. S. M. Goh, "Learning with annotation of various degrees," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2019.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[44] X. Chen, B. Athiwaratkun, Y. Sun, K. Q. Weinberger, and C. Cardie, "Adversarial deep averaging networks for cross-lingual sentiment classification," *TACL*, vol. 6, pp. 557–570, 2016.

[45] P. Liu, X. Qiu, and X. Huang, "Adversarial multi-task learning for text classification," in *ACL*, 2017.

[46] T. Gui, Q. Zhang, H. Huang, M. Peng, and X. Huang, "Part-of-speech tagging for twitter with adversarial neural networks," in *EMNLP*, 2017.

[47] J.-K. Kim, Y.-B. Kim, R. Sarikaya, and E. Fosler-Lussier, "Cross-lingual transfer learning for pos tagging without cross-lingual resources," in *EMNLP*, 2017, pp. 2832–2838.

[48] X. Chen and C. Cardie, "Multinomial adversarial networks for multi-domain text classification," in *NAACL HLT*, 2018, pp. 1226–1240.

[49] C. Szegedy, W. Zaremba, D. E. I. G. Ilya Sutskever, Joan Bruna, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[50] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[51] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *ICLR*, 2017.

[52] M. Yasunaga, J. Kasai, and D. Radev, "Robust multilingual part-of-speech tagging via adversarial training," in *NAACL HLT*, 2018.

[53] N. Reimers and I. Gurevych, "Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging," in *EMNLP*, 2017, pp. 338–348.

[54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.

[55] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *ICCV*, 2017.

[56] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *ICLR*, 2017.

[57] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

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

[59] C. Pin-Yu, S. Yash, Z. Huan, Y. Jinfeng, and C.-J. Hsieh, "Ead: Elastic-net attacks to deep neural networks via adversarial examples," in *AAAI*, 2018.

[60] E. F. Tjong Kim Sang and S. Buchholz, "Introduction to the conll-2000 shared task: Chunking," in *Proceedings of the 2Nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning*, ser. CoNLL, 2000, pp. 127–132.

[61] S. E. F. T. Kim, "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)*, 2002.

[62] S. E. F. T. Kim and M. F. De, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *NAACL HLT*, 2003.

[63] D. e. a. Zeman, "Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies," in *CoNLL 2017*, 2017, pp. 1–19.

[64] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The penn treebank: Annotating predicate argument structure," in *Proceedings of the Workshop on Human Language Technology*, 1994, pp. 114–119.

[65] S. S. Pradhan, E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "Ontonotes: A unified relational semantic representation," in *ICSC*, 2007, pp. 517–526.

[66] J. Nivre, Ž. Agić, M. J. Aranzabe, M. Asahara, A. Atutxa, M. Ballesteros, J. Bauer, K. Bengoetxea, R. A. Bhat, C. Bosco, S. Bowman, G. G. A. Celano, M. Connor, M.-C. de Marneffe, A. Diaz de Ilarraza, K. Dobrovoljc, T. Dozat, T. Erjavec, R. Farkas, J. Foster, D. Galbraith, F. Ginter, I. Goenaga, K. Gojenola, Y. Goldberg, B. Gonzales, B. Guillaume, J. Hajič, D. Haug, R. Ion, E. Irimia, A. Johannsen, H. Kanayama, J. Kanerva, S. Krek, V. Laippala, A. Lenci, N. Ljubešić, T. Lynn, C. Manning, C. Mărănduc, D. Mareček, H. Martínez Alonso, J. Mašek, Y. Matsumoto, R. McDonald, A. Missilä, V. Mititelu, Y. Miyao, S. Montemagni, S. Mori, H. Nurmi, P. Osenova, L. Øvrelid, E. Pascual, M. Passarotti, C.-A. Perez, S. Petrov, J. Piitulainen, B. Plank, M. Popel, P. Prokopidis, S. Pyysalo, L. Ramasamy, R. Rosa, S. Saleh, S. Schuster, W. Seeker, M. Seraji, N. Silveira, M. Simi, R. Simionescu, K. Simkó, K. Simov, A. Smith, J. Štěpánek, A. Suhr, Z. Szántó, T. Tanaka, R. Tsarfaty, S. Uematsu, L. Uria, V. Varga, V. Vincze, Z. Žabokrtský, D. Zeman, and H. Zhu, "Universal dependencies 1.2," 2015, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[67] I. Partalas, C. Lopez, N. Derbas, and R. Kalitvianski, "Learning to search for recognizing named entities in twitter," in *WNUT*, 2016, pp. 171–177.

[68] N. Limsopatham and N. Collier, "Bidirectional lstm for named entity recognition in twitter messages," in *WNUT*, 2016, pp. 145–152.

[69] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *TACL*, vol. 5, pp. 135–146, 2017.

[70] R. Al-Rfou', B. Perozzi, and S. Skiena, "Polyglot: Distributed word representations for multilingual NLP," in *CoNLL*, 2013, pp. 183–192.

[71] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[72] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *JMLR*, pp. 1929–1958, 2014.

[73] G. Aguilar, A. P. López-Monroy, F. A. González, and T. Solorio, "Modeling noisiness to recognize named entities using multitask neural networks on social media," in *NAACL-HLT*, 2018.

[74] D. Gillick, C. Brunk, O. Vinyals, and A. Subramanya, "Multilingual language processing from bytes," in *NAACL HLT*, 2016, pp. 1296–1306.

[75] B. Y. Lin, F. Xu, Z. Luo, and K. Zhu, "Multi-channel bilstm-crf model for emerging named entity recognition in social media," in *WNUT*, 2017, pp. 160–165.

[76] P. von Däniken and M. Cieliebak, "Transfer learning and sentence level features for named entity recognition on tweets," in *WNUT*, 2017, pp. 166–171.

[77] G. Berend, "Sparse coding of neural word embeddings for multilingual sequence labeling," *TACL*, vol. 5, pp. 247–261, 2017.

[78] B. Plank, A. Søgaard, and Y. Goldberg, "Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss," in *ACL*, 2016, pp. 412–418.

[79] D. Q. Nguyen, M. Dras, and M. Johnson, "A novel neural network model for joint POS tagging and graph-based dependency parsing," in *CoNLL*, 2017, pp. 134–142.

[80] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *JMLR*, vol. 9, pp. 2579–2605, 2008.

[81] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual bert?" *CoRR*, vol. abs/1906.01502, 2019.