Fine-grained Knowledge Fusion for Sequence Labeling Domain Adaptation

Huiyun Yang[1,2] **Shujian Huang**[1,2] **Xinyu Dai**[1,2] **Jiajun Chen**[1,2]
National Key Laboratory for Novel Software Technology, Nanjing, China[1]
Nanjing University, Nanjing, China[2]
yanghy@smail.nju.edu.cn
{huangsj, daixinyu, chenjj}@nju.edu.cn

Abstract

In sequence labeling, previous domain adaptation methods focus on the adaptation from the source domain to the entire target domain without considering the diversity of individual target domain samples, which may lead to negative transfer results for certain samples. Besides, an important characteristic of sequence labeling tasks is that different elements within a given sample may also have diverse domain relevance, which requires further consideration. To take the multi-level domain relevance discrepancy into account, in this paper, we propose a fine-grained knowledge fusion model with the domain relevance modeling scheme to control the balance between learning from the target domain data and learning from the source domain model. Experiments on three sequence labeling tasks show that our fine-grained knowledge fusion model outperforms strong baselines and other state-of-the-art sequence labeling domain adaptation methods. [1]

1 Introduction

Sequence labeling tasks, such as Chinese word segmentation (CWS), POS tagging (POS) and named entity recognition (NER), are fundamental tasks in natural language processing. Recently, with the development of deep learning, neural sequence labeling approaches have achieved pretty high accuracy (Chen et al., 2017; Zhang and Yang, 2018), relying on large-scale annotated corpora. However, most of the standard annotated corpora belong to the news domain, and models trained on these corpora will get sharp declines in performance when applied to other domains like social media, forum, literature or patents (Daume III, 2007; Blitzer et al., 2007), which limits their application in the real world. Domain adaptation aims to exploit the abundant information of well-studied source domains to improve the performance in target domains (Pan and Yang, 2010), which is suitable to handle this issue. Following Daume III (2007), we focus on the supervised domain adaptation setting, which utilizes large-scale annotated data from the source domain and small-scale annotated data from the target domain.

Types	Cases
Strongly	*Ops* **Steve Jobs resigned as CEO of Apple. Share prices are rising** *soooo* **fast**!
Weakly	*Alas* **as time goes by**, *hair's gone. Rock to 204 Section* **next week**!

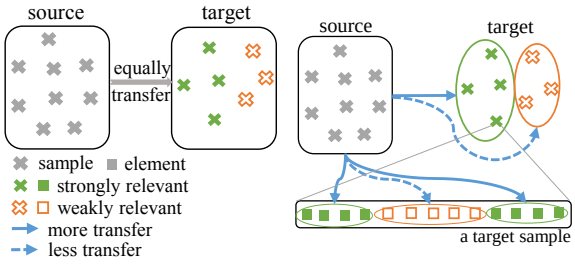
Table 1: Tweets from the social media domain have different degrees of relevance to the source domain (news). Within each case, the bold part is strongly relevant and the italic part is weakly relevant.

For sequence labeling tasks, each sample is usually a sentence, which consists of a sequence of words/Chinese characters, denoted as the element. We notice an interesting phenomenon: different target domain samples may have varying degrees of domain relevance to the source domain. As depicted in Table 1, there are some tweets similar to the news domain (i.e. strongly relevant). But there are also some tweets of their own style, which only appear in the social media domain (i.e. weakly relevant). The phenomenon can be more complicated for the cases where the whole sample is strongly relevant while contains some target domain specific elements, or vice versa, showing the diversity of relevance at the element-level. In the rest of this paper, we use 'domain relevance' to refer to the domain relevance to the source domain, unless specified otherwise.

Conventional neural sequence labeling domain adaptation methods (Liu and Zhang, 2012; Liu et al., 2014; Zhang et al., 2014; Chen et al., 2017; Peng and Dredze, 2017; Lin and Lu, 2018) mainly

[1]Our code is available at https://github.com/yhy1117/FGKF-DA.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4197–4206, Hong Kong, China, November 3–7, 2019.



(a) Previous methods (b) Our method

Figure 1: Previous methods transfer knowledge by the whole sample set, while our method consider diverse domain relevance within the target domain set and within every target sample to transfer knowledge respectively.

focus on reducing the discrepancy between the sets of source domain samples and target domain samples. However, they neglect the diverse domain relevance of individual target domain samples, let alone the element-level domain relevance. As depicted in Figure 1, obviously, strongly relevant samples/elements should learn more knowledge from the source domain, while weakly relevant samples/elements should learn less and keep their characteristics.

In this paper, we propose a fine-grained knowledge fusion model to control the balance between learning from the target domain data and learning from the source model, inspired by the knowledge distillation method (Bucila et al., 2006; Hinton et al., 2015). With both the sample-level and element-level domain relevance modeling and incorporating, the fine-grained knowledge fusion model can alleviate the *negative transfer* (Rosenstein et al., 2005) in sequence labeling domain adaptation.

We verify the effectiveness of our method on six domain adaptation experiments of three different tasks, i.e. CWS, POS and NER, in two different languages, i.e. Chinese and English, respectively. Experiments show that our method achieves better results than previous state-of-the-art methods on all tasks. We also provide detailed analyses to study the knowledge fusion process.

Contributions of our work are summarized as follows:

- We propose a fine-grained knowledge fusion model to balance the learning from the target data and learning from the source model.
- We also propose multi-level relevance modeling schemes to model both the sample-level and element-level domain relevance.
- Empirical evidences and analyses are provided on three different tasks in two different languages, which verify the effectiveness of our method.

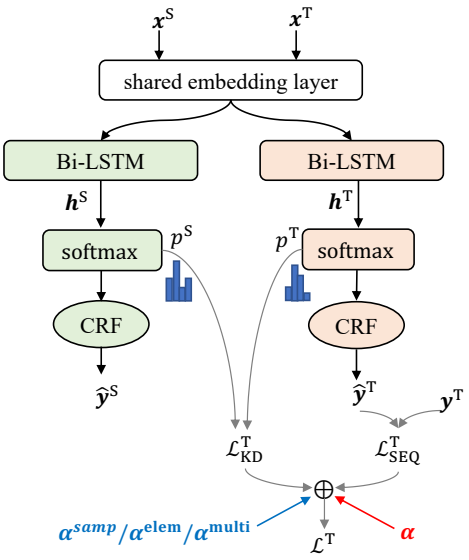


Figure 2: The architecture of basicKD (with the red α, see §2) or fine-grained knowledge fusion model (with the blue α, see §4), where the green part belongs to the source model, the orange part belongs to the target model and the white part is common. Better viewed in color.

2 Knowledge Distillation for Adaptation

Knowledge distillation (KD), which distills the knowledge from a sophisticated model to a simple model, has been employed in domain adaptation (Bao et al., 2017; Meng et al., 2018). Recently, online knowledge distillation(Furlanello et al., 2018; Zhou et al., 2018) is shown to be more effective, which shares lower layers between the two models and trains them simultaneously.

For sequence labeling domain adaptation, we utilize the online knowledge distillation method to distill knowledge from the source model to improve the target model, denoted as basicKD, which is depicted in Figure 2. We use the Bi-LSTM-CRF architecture (Huang et al., 2015), for both the source model and the target model, and share the embedding layer between them.

Notations For the rest of the paper, we use the superscript S and T to denote the source domain and the target domain, respectively. Source domain data is a set of m samples with gold la-

bel sequences, denoted as $(\mathbf{x}_j^S, \mathbf{y}_j^S)_{j=1}^m$. Similarly, target domain data has n samples, denoted as $(\mathbf{x}_i^T, \mathbf{y}_i^T)_{i=1}^n$, where $n \ll m$.

The training loss of the source model is the cross entropy between the predicted label distribution $\hat{\mathbf{y}}$ and the gold label $\mathbf{y}$:

$$\mathcal{L}^S = -\frac{1}{m}\sum_{j=1}^{m} \mathbf{y}_j^S \log \hat{\mathbf{y}}_j^S \tag{1}$$

The training loss of the target model is composed of two parts, namely the sequence labeling loss $\mathcal{L}_{\text{SEQ}}^T$ and the knowledge distillation loss $\mathcal{L}_{\text{KD}}^T$:

$$\mathcal{L}^T = (1-\alpha)\mathcal{L}_{\text{SEQ}}^T + \alpha\mathcal{L}_{\text{KD}}^T \tag{2}$$

$$\mathcal{L}_{\text{SEQ}}^T = -\frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i^T \log \hat{\mathbf{y}}_i^T \tag{3}$$

$$\mathcal{L}_{\text{KD}}^T = -\frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_i^S \log \mathbf{p}_i^T \tag{4}$$

where $\mathcal{L}_{\text{SEQ}}^T$ is similar to $\mathcal{L}^S$, while $\mathcal{L}_{\text{KD}}^T$ is the cross entropy between the probability distributions predicted by the source model and the target model. α is a hyper-parameter scalar, which is used to balance the learning from the target domain data and the learning from the source model.

3 Relevance Modeling

BasicKD provides individual learning goals for every sample and element of the target domain, using a scalar α to weight. As a result, the source model has the same influence on all target samples, in which the diversity of domain relevance is neglected.

Here we present methods to model the domain relevance of target samples and elements, which could then be used to guide the knowledge fusion process (see §4). The overall architecture is shown in Figure 3. The relevance of each sample is a scalar, denoted as the sample-level relevance weight, $\mathbf{w}_i^{\text{samp}}$ for the i^{th} sample, which can be obtained by the sample-level domain classification. The relevance of each element is also a scalar, while the relevance weights of all elements within a sample form a weight vector $\mathbf{w}^{\text{elem}}$, which can be obtained by the similarity calculation.

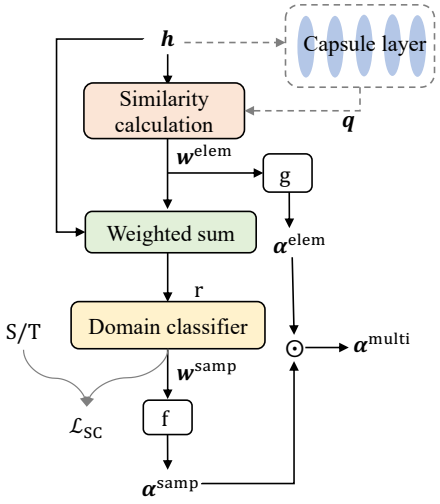


Figure 3: The relevance modeling process (see §3), where the block f denotes Eq.(10) and the block g denotes Eq.(14).

3.1 Element-level Relevance

To acquire the element-level relevance, we employ the domain representation $\boldsymbol{q} \in \mathbb{R}^{2d_h}$ (d_h is the dimension of the Bi-LSTM) and calculate the similarity between the element representation and the domain representation. We incorporate two methods to get $\boldsymbol{q}$: (1) Domain-q: $\boldsymbol{q}$ is a trainable domain specific vector, where every element within a domain share the same $\boldsymbol{q}$; (2) Sample-q: $\boldsymbol{q}$ is the domain relevant feature extracted from each sample, where every element within a sample share the same $\boldsymbol{q}$. Because of the superiority of the capsule network modeling abstract features (Gong et al., 2018; Yang et al., 2018), we use it to capture the domain relevant features within a sample. We incorporate the same bottom-up aggregation process as Gong et al. (2018) and the encoded vector is regarded as $\boldsymbol{q}$:

$$\mathbf{q} = \text{Capsule}(\mathbf{h}) \tag{5}$$

where $\mathbf{h}$ is the hidden state matrix of a sample.

The similarity calculation formula is the matrix dot [2]:

$$\mathbf{w}_j^{\text{elem}} = \mathbf{q}^\top \mathbf{B}\mathbf{h}_j \tag{6}$$

where $\boldsymbol{h}_j$ is the hidden states of the j^{th} element and $\mathbf{w}_j^{\text{elem}}$ is the relevance weight of it. $\mathbf{B} \in \mathbb{R}^{2d_h \times 2d_h}$ is a trainable matrix.

[2]We also try dot and MLP, while matrix dot get better performance with fewer parameters.

3.2 Sample-level Relevance

To acquire the sample-level domain relevance, we make use of the domain label to carry out sample-level text classification (two class, source domain or target domain). The weight $\mathbf{w}^{\text{elem}}$ is normalized across the sample length using the softmax function, then the sample representation can be obtained by the weighted sum of hidden states. The process can be expressed as:

$$\hat{\mathbf{w}}_j^{\text{elem}} = \frac{\exp(\mathbf{w}_j^{\text{elem}})}{\sum_k \exp(\mathbf{w}_k^{\text{elem}})} \tag{7}$$

$$\mathbf{r} = \sum_{j=1}^{L} \hat{\mathbf{w}}_j{}^{\text{elem}} \cdot \mathbf{h}_j \tag{8}$$

$\mathbf{r} \in \mathbb{R}^{2d_h}$ is the sample representation and L is the sample length.

Once the sample representation is obtained, the multi-layer perceptron (MLP) and softmax do sample classification next:

$$[w^{\text{samp}}, 1 - w^{\text{samp}}] = [\text{softmax}(\text{MLP}(\mathbf{r}))]^\top \tag{9}$$

where w^{samp} is the sample relevance weight.

4 Fine-grained Knowledge Fusion for Adaptation

With the relevance modeling, the fine-grained knowledge fusion model is proposed to fusion the knowledge from the source domain and the target domain at different levels. The overall architecture is shown in Figure 2.

4.1 Sample-level Knowledge Fusion

Different samples of target domain tend to show different domain relevance, and as a result, they need to acquire different amount of knowledge from the source domain. Different α is assigned to each target sample based on its domain relevance to achieve the sample-level knowledge fusion. The new α can be computed as:

$$\boldsymbol{\alpha}_i^{\text{samp}} = \sigma(\tau \cdot \mathbf{w}_i^{\text{samp}} + \gamma) \tag{10}$$

where $\boldsymbol{\alpha}_i^{\text{samp}}$ is the α of the i^{th} sample and $\mathbf{w}_i^{\text{samp}}$ is the relevance weight of it; σ denotes the sigmoid function; τ is temperature and γ is bias.

The loss functions of the target model can be computed as:

$$\mathcal{L}^T = \mathcal{L}^T_{\text{SEQ}} + \mathcal{L}^T_{\text{KD}} \tag{11}$$

$$\mathcal{L}^T_{\text{SEQ}} = -\frac{1}{n}\sum_{i=1}^{n}(1 - \boldsymbol{\alpha}_i^{\text{samp}})\mathbf{y}_i^T \log \hat{\mathbf{y}}_i^T \tag{12}$$

$$\mathcal{L}^T_{\text{KD}} = -\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\alpha}_i^{\text{samp}}\mathbf{p}_i^S \log \mathbf{p}_i^T \tag{13}$$

The sample classification losses of the source model $\mathcal{L}^S_{sc}$ and target model $\mathcal{L}^T_{sc}$ are both cross entropy.

4.2 Element-level Knowledge Fusion

Besides the sample-level domain relevance, different elements within a sample tend to present diverse domain relevance. In this method, we assign different α to each element based on its domain relevance weight to achieve the element-level knowledge fusion. The new α can be computed as:

$$\boldsymbol{\alpha}_i^{\text{elem}} = \sigma(\mathbf{W}_\alpha \mathbf{w}_i^{\text{elem}} + \mathbf{b}_\alpha)] \tag{14}$$

where $\boldsymbol{\alpha}_i^{\text{elem}} \in \mathbb{R}^L$ is a vector, in which $\boldsymbol{\alpha}_{ij}^{\text{elem}}$ denotes the α of the j^{th} element in the i^{th} sample. $\mathbf{w}_i^{\text{elem}}$ is the relevance weight of the i^{th} sample. $\mathbf{W}_\alpha$ and $\mathbf{b}_\alpha$ are trainable parameters.

The loss functions of the target model can be expressed as:

$$\mathcal{L}^T_{\text{SEQ}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{L}(1 - \boldsymbol{\alpha}_{ij}^{\text{elem}})\mathbf{y}_{ij}^T \log \hat{\mathbf{y}}_{ij}^T \tag{15}$$

$$\mathcal{L}^T_{\text{KD}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{L}\boldsymbol{\alpha}_{ij}^{\text{elem}}\mathbf{p}_{ij}^S \log \mathbf{p}_{ij}^T \tag{16}$$

where $*_{ij}$ denotes the $*$ of the j^{th} element in the i^{th} sample, and the final loss function is the same with Eq.(11).

4.3 Multi-level Knowledge Fusion

In this method, we take both the sample-level and element-level relevance diversities into account to implement the multi-level knowledge fusion, and the multi-level α can be computed as:

$$\boldsymbol{\alpha}^{\text{multi}} = \boldsymbol{\alpha}^{\text{samp}} \odot \boldsymbol{\alpha}^{\text{elem}} \tag{17}$$

where $\odot$ denotes the element-wise product. $\boldsymbol{\alpha}^{\text{multi}} \in \mathbb{R}^{n \times L}$ is a matrix as well.

The loss functions of the target model can be obtained by replacing $\boldsymbol{\alpha}_{ij}^{\text{elem}}$ with $\boldsymbol{\alpha}_{ij}^{\text{multi}}$ in Eq.(15) and Eq.(16).

Task	Language	Source	Target	Domain
CWS	Chinese	CTB6 (Xue et al., 2005)	Zhuxian (Zhang et al., 2014)	news → novels
POS	Chinese	CTB6 (Xue et al., 2005)	Weibo (Qiu et al., 2016)	news → social media
	English	PTB (Marcus et al., 1993)	Twitter (Ritter et al., 2011)	
NER	Chinese	MSRA (Levow, 2006)	WeiboNER (Peng and Dredze, 2015)	
	English	Ontonotes (Ralph et al., 2013)	Twitter (Ritter et al., 2011)	

Table 2: Datasets used in this paper.

Algorithm 1 Training Process of Knowledge Fusion

1. **Input**: source data, target data
2. **Hyper − parameters**: batch size b, teach step I
3. Initialize parameters of the source and target model
4. (optional) Use the source data to pre-train $\boldsymbol{\theta}^S$ and $\boldsymbol{\theta}^T$
5. **repeat**
6. **for** $i = 1$ to I **do**
7. Sample b samples from the source data
8. Compute $\mathcal{L}^S$, and update $\boldsymbol{\theta}^S$
9. Compute $\mathcal{L}^S_{sc}$, and update $\boldsymbol{\theta}^S$
10. **end for**
11. Use $\boldsymbol{\theta}^S$ to test $\mathbf{x}^T_{train}$ and get $\mathbf{p}^S$
12. **while** in an episode:
13. Sample b samples from the target data
14. Use relevance modeling to get $\mathbf{w}^{\text{samp}}$,$\mathbf{w}^{\text{elem}}$
15. Compute $\boldsymbol{\alpha}^{\text{samp}}$/$\boldsymbol{\alpha}^{\text{elem}}$/$\boldsymbol{\alpha}^{\text{multi}}$ and $\mathcal{L}^T_{\text{SEQ}}$
16. Use $\boldsymbol{\theta}^T$ to predict $\mathbf{p}^T$, and compute $\mathcal{L}^T_{\text{KD}}$
17. Compute $\mathcal{L}^T$, and update $\boldsymbol{\theta}^T$
18. Compute $\mathcal{L}^T_{sc}$, and update $\boldsymbol{\theta}^T$
19. **end while**
20. **until** converge

4.4 Training Process

Both the source model and the target model can be pre-trained on the source domain data (warm up, optional). In the fine-grained knowledge fusion method, the source model and the target model are trained alternately. Within an episode, we use I steps to train the source model ahead, then the soft target ($\mathbf{p}^S$) can be obtained and the target model will be trained. During the training of the target model, the parameters of the source model are fixed (gradient block). Every training step includes the sequence labeling training and the sample classification training. We conduct early stopping according to the performance of the target model. The whole training process is shown in Algorithm 1.

5 Experiments

5.1 Datasets

We conduct three sequence labeling tasks: CWS, POS and NER, and the latter two tasks containing both Chinese and English settings. Detailed datasets are shown in Table 2. There are two kinds of source-target domain pairs: news-novels and news-social media. To be consistent with the setting where there is only small-scale target domain data, we use 5% training data of Weibo for both CWS and POS. For the different NER tag sets, we only focus on three types of entities: Person (PER), Location (LOC) and Organization (ORG) and regard other types as Other (O).

5.2 Settings

For each task, hyper-parameters are set via grid search on the target domain development set. Embedding size and the dimension of LSTM hidden states is set to 100. Batch size is set to 64. Learning rate is set to 0.01. We employ the dropout strategy on the embedding and MLP layer with the rate of 0.2. The l_2 regularization term is set to 0.1. The gradient clip is set to 5. The teach step I is set to 100. The routing iteration is set to 3 and the number of the output capsules is set to 60. The temperature τ is initialized to 1 and the probability bias γ is initialized to 0.5. We set the α of the basicKD method to 0.5 according to Hinton et al. (2015). We randomly initialize the embedding matrix without using extra data to pre-train, unless specified otherwise.

5.3 Baselines

We implement several baseline methods, including: **source only** (training with only source domain data), **target only** (training with only target domain data) and **basicKD** (see §2).

We also re-implement state-of-the-art sequence labeling domain adaptation methods, following their settings except for unifying the embedding size and the dimension of LSTM hidden states:

- **Pre-trained methods: Pre-trained embedding** incorporates source domain data with its gold label to pre-train context-aware character embedding (Zhou et al., 2017), which is used to initialize the target model; **Pre-trained model** trains the model on the source domain and then finetune it on the target domain.

Methods	CWS				POS		NER	
	Zhuxian		5% Weibo					
	F	R_{OOV}	F	R_{OOV}	zh	en	zh	en
Target only	92.80	65.81	84.01	64.12	93.03	86.83	46.49	59.58
BasicKD	94.23	74.08	89.21	76.26	95.69	89.96	49.92	62.15
Pre-trained embedding	93.70	70.44	87.62	72.27	94.96	89.70	52.53	61.36
Pre-trained model	94.43	74.30	89.50	76.27	96.10	90.05	54.25	62.88
Linear projection	94.14	72.75	88.77	75.85	95.92	89.36	52.71	62.27
Domain mask	94.30	75.20	88.84	75.03	96.01	89.81	54.12	62.64
NAL	94.47	74.62	88.63	75.77	96.19	90.48	54.70	63.32
AMCL	94.62	74.46	89.42	76.16	94.13	89.12	51.47	61.57
FGKF	95.01	77.26	90.45	77.27	**96.60**	91.33	55.60	63.81
+ Pre-trained embedding	**95.09**	**77.56**	**90.73**	**77.87**	96.36	**91.66**	**57.57**	**65.51**

Table 3: Results of domain adaptation on three tasks, where zh denotes the Weibo datasets (in Chinese), and en denotes the Twitter dataset (in English).

- **Projection methods: Linear projection** (Peng and Dredze, 2017) uses the domain-relevant matrix to transform the learned representation from different domains into the shared space; **Domain mask** (Peng and Dredze, 2017) masks the hidden states of Bi-LSTM to split the representations into private and public regions to do the projection; **Neural adaptation layer** (NAL) (Lin and Lu, 2018) incorporates adaptation layers at the input and output to conduct private-public-private projections.

- **Adversarial method: Adversarial multi-criteria learning** (AMCL) (Chen et al., 2017) uses the shared-private architecture with the adversarial strategy to learn the shared representations across domains.

5.4 Overall Results on CWS

We use the F1-score (F) and the recall of out-of-vocabulary words (R_{oov}) to evaluate the domain adaptation performance on CWS. We compare methods with different relevance modeling schemes and different levels of knowledge fusion, without warm up. And we denote our final model as **FGKF**, which is the multi-level knowledge fusion with the sample-q relevance modeling and warm up.

The results in Table 4 show that both the basicKD method and fine-grained methods achieve performance improvements through domain adaptation. Compared with the basicKD method, FGKF behaves better (+1.1% F and +2.8% R_{oov} v.s. basicKD on average), as it takes multi-level relevance discrepancies into account. The sample-q method performs better than the domain-q method, which shows the domain feature is better represented at the sample level, not at the domain level. As for the granularity of α, the performances of $\boldsymbol{\alpha}^{\text{elem}}$ is better than $\boldsymbol{\alpha}^{\text{samp}}$, showing the necessity of modeling element-level relevance. And there isn't a distinct margin between $\boldsymbol{\alpha}^{\text{elem}}$ and $\boldsymbol{\alpha}^{\text{multi}}$ as most of the multi-level domain relevance can be included by the element level. Results of FGKF with warm up indicate that starting from sub-optimal point is better than starting from scratch for the target model.

Methods	Zhuxian		5% Weibo	
	F	R_{OOV}	F	R_{OOV}
Source only	83.86	62.40	83.75	70.74
Target only	92.80	65.81	84.01	64.12
BasicKD	94.23	74.08	89.21	76.26
Domain-q $\boldsymbol{\alpha}^{\text{samp}}$	94.55	74.02	89.63	75.93
Domain-q $\boldsymbol{\alpha}^{\text{elem}}$	94.81	74.75	89.99	**77.59**
Domain-q $\boldsymbol{\alpha}^{\text{multi}}$	94.75	74.96	90.06	77.25
Sample-q $\boldsymbol{\alpha}^{\text{samp}}$	94.57	74.47	89.77	76.81
Sample-q $\boldsymbol{\alpha}^{\text{elem}}$	94.78	74.52	90.07	76.94
Sample-q $\boldsymbol{\alpha}^{\text{multi}}$	94.91	75.56	90.20	77.46
FGKF	**95.01**	**77.26**	**90.45**	77.27

Table 4: Results of baselines and fine-grained knowledge fusion methods on CWS.

Among related works (Table 3), AMCL and Pre-trained model methods have better performances in CWS. Compared with other methods, FGKF achieves the best results in both F and R_{OOV}. Results demonstrate the effectiveness of our fine-grained knowledge fusion architecture for domain adaptation, and also show the significance of considering sample-level and element-level relevance discrepancies.

5.5 Overall Results on POS and NER

To further verify the effectiveness of FGKF, we conduct experiments on POS and NER tasks, using F1-score as the evaluation criterion. Detailed results are shown in Table 3. In these tasks, FGKF

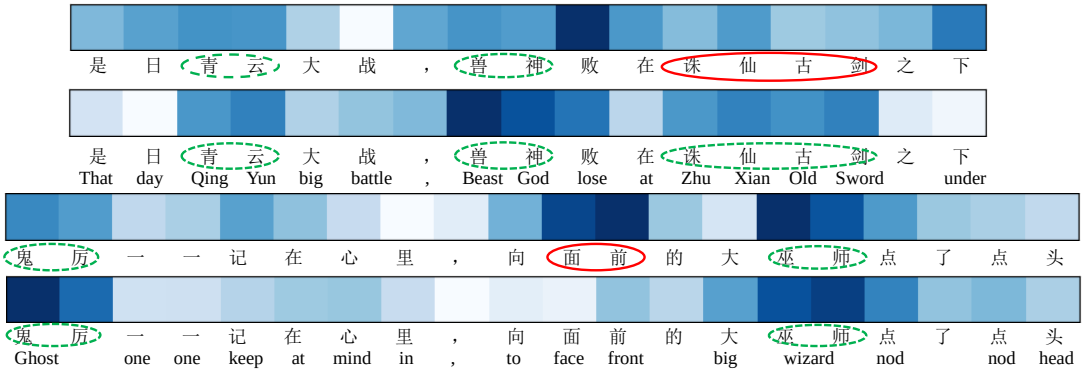


Figure 4: Two cases of the element-level relevance modeling visualization, where the upper one belongs to the domain-q method and the lower one belongs to the sample-q method. The green dotted circle indicates the correct domain relevant element and the red solid circle indicates the ignored or mistaken extracted element.

achieves better results than other adaptation methods. Extra gain could be obtained by using pre-trained embedding. These results also verify the generalization of our method over different tasks and languages.

6 Analysis

In this section, we will display and discuss the domain adaptation improvements provided by our fine-grained knowledge fusion method.

6.1 Performances of Elements with Different Relevance

To further probe into the experimental results of the fine-grained knowledge fusion, we classify the target test data (in element level) into two classes: strongly relevant and weakly relevant, based on their relevance degrees to the source domain. The partition threshold is according to the average relevance score of the target training data. Detailed results on Twitter are depicted in Table 5.

Methods	POS		NER	
	Strong	Weak	Strong	Weak
Source only	87.47	82.48	68.27	46.30
Target only	86.46	87.41	62.01	56.29
BasicKD	91.92	83.82	70.20	52.63
FGKF	**92.55**	**89.93**	**71.81**	**57.92**

Table 5: Results of the strongly/weakly relevant elements on the Twitter test set.

It is reasonable that both the basicKD and FGKF enhance the performance of the strongly relevant part, while FGKF get larger improvements because it is able to enhance the knowledge fusion by learning more from the source model. For the weakly relevant part, the basicKD method damages the performance on it (from 87.41 to 83.82 for POS and from 56.29 to 52.63 for NER), which indicate the negative transfer. On the contrary, FGKF improves the performance of the weakly relevant part compared with the target only baseline with a large margin. It is shown that the fine-grained domain adaptation method can reduce the negative transfer on the weakly relevant part and contribute to the transfer on the strongly relevant one.

6.2 Relevance Weight Visualization

We carry out the visualization of the element-level relevance weight to illustrate the effects of the two relevance modeling schemes (domain-q and sample-q). Figure 4 exhibits two cases of element-level relevance modeling results, from which we can explicitly observe that the two schemes capture different domain relevance within a sample. In the first case, the sample-q method extracts more domain relevant elments, like “Qingyun”, “Beast God” and “Zhuxian Old Sword”, while the domain-q method ignores the last one. In the second case, the domain-q method extracts “front” incorrectly. These results indicate that the sample-q method can implement better relevance modeling than the domain-q method to some extent, and prove that the domain relevant feature is better represented at the sample level, not at the domain level.

6.3 Case Study

We take two samples in Twitter test set as examples to show how the element-level relevance affects the adaptation. Results in Table 6 show that both basicKD and FGKF can improve the performance of strongly relevant elements, e.g. “got (VBD)”, “Lovis (B-PER)”. However, only FGKF

Tasks	POS					NER				
Sentence	I	got	u	next	week	Louis	interview	with	The	Sun
Source only	PN	VBD	NN	JJ	NN	B-PER	O	O	O	O
Target only	PN	VBZ	PN	JJ	NN	O	O	O	B-ORG	I-ORG
BasicKD	PN	VBD	NN	JJ	NN	B-PER	O	O	O	O
FGKF	PN	VBD	PN	JJ	NN	B-PER	O	O	B-ORG	I-ORG

Table 6: Two cases of domain adaptation, where the underlined tags are wrong.

reduces the transfer of source domain errors, e.g. "u (NN)", "The (B-ORG) Sun (I-ORG)".

6.4 Ablation Study

We conduct the ablation study on Twitter dataset (Table 7). Results show the gradient block and the multi-level knowledge fusion are of vital importance to FGKF. The embedding sharing and warm up also make contributions.

Methods	POS		NER	
	F	Δ	F	Δ
FGKF	91.33	-	63.81	-
w/o share embedding	90.75	-0.58	62.47	-1.34
w/o gradient block	88.48	-2.85	58.83	-4.98
w/o α^{samp}	90.94	-0.39	63.52	-0.30
w/o α^{elem}	90.23	-1.10	62.43	-1.38
w/o α^{multi}	90.12	-1.21	62.32	-1.49
w/o warm up	90.89	-0.44	63.17	-0.64

Table 7: Ablation results of the Twitter test set.

6.5 Influence of Target Data Size

Here we investigate the impact of the target domain data size on FGKF. As is depicted in Figure 5, when the size is small (20%), the gap is pretty huge between FGKF and basicKD, which verifies the significance of fine-grained knowledge fusion in the low-resource setting. Even with the size of target data increasing, there are still stable margins between the two methods.

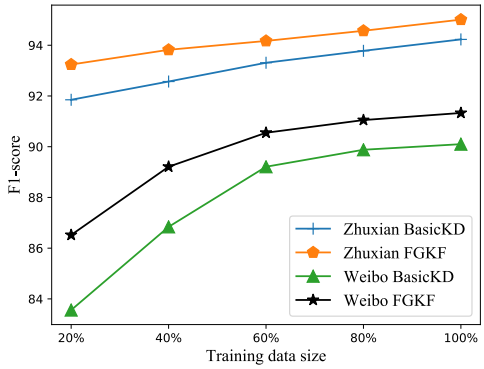


Figure 5: Results of CWS target test set with varying target training data size. Only 10% training data of Weibo is utilized.

7 Related Work

Besides the source domain data, some methods utilize the target domain lexicons (Liu et al., 2014; Zhang et al., 2014), unlabeled (Liu and Zhang, 2012) or partial-labeled target domain data (Liu et al., 2014) to boost the sequence labeling adaptation performance, which belong to unsupervised or semi-supervised domain adaptation. However, we focus on supervised sequence labeling domain adaptation, where huge improvement can be achieved by utilizing only small-scale annotated data from the target domain.

Previous works in domain adaptation often try to find a subset of source domain data to align with the target domain data (Chopra et al., 2013; Ruder and Plank, 2017) which realizes a kind of source data sample or construct a common feature space, while those methods may wash out informative characteristics of target domain samples. Instance-based domain adaptation (Jiang and Zhai, 2007; Zhang and Xiong, 2018) implement the source sample weighting by assigning higher weights to source domain samples which are more similar to the target domain. There are also some methods (Guo et al., 2018; Kim et al., 2017; Zeng et al., 2018) explicitly weighting multiple source domain models for target samples in multi-source domain adaptation. However, our work focuses on the supervised single source domain adaptation, which devote to implementing the knowledge fusion between the source domain and the target domain, not within multiple source domains. Moreover, considering the important characteristics of sequence labeling tasks, we put more attention to the finer-grained adaptation, considering the domain relevance in sample level and element level.

8 Conclusion

In this paper, we propose a fine-grained knowledge fusion model for sequence labeling domain adaptation to take the domain relevance diversity of target data into account. With the relevance modeling on both the sample level and element

level, the knowledge of the source model and target data can achieve multi-level fusion. Experimental results on different tasks demonstrate the effectiveness of our approach, and show the potential of our approach in a broader range of domain adaptation applications.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. Shujian Huang is the corresponding author. This work is supported by National Science Foundation of China (No. U1836221, No. 61772261), National Key RD Program of China (No. 2019QY1806).

References

Zuyi Bao, Si Li, Weiran Xu, and Sheng Gao. 2017. Neural regularized domain adaptation for Chinese word segmentation. In *AFNLP*.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *SIGKDD*.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-criteria learning for Chinese word segmentation. In *ACL*.

Sumit Chopra, Suhrid Balakrishnan, and Raghuraman Gopalan. 2013. Dlid: Deep learning for domain adaptation by interpolating between domains. In *ICML*.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *ACL*.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *ICML*.

Jingjing Gong, Xipeng Qiu, Shaojing Wang, and Xuanjing Huang. 2018. Information aggregation via dynamic routing for sequence encoding. In *COLING*.

Jiang Guo, Darsh Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. In *EMNLP*.

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

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. arXiv:1508.01991. Version 1.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. In *ACL*.

Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. Domain attention with an ensemble of experts. In *ACL*.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *AFNLP*.

Bill Yuchen Lin and Wei Lu. 2018. Neural adaptation layers for cross-domain named entity recognition. In *EMNLP*.

Yang Liu and Yue Zhang. 2012. Unsupervised domain adaptation for joint segmentation and pos-tagging. In *COLING*.

Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan Wu. 2014. Domain adaptation for CRF-based Chinese word segmentation using free annotations. In *EMNLP*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The penn treebank. *Computational Linguistics*.

Zhong Meng, Jinyu Li, Yifan Gong, and Biing-Hwang Juang. 2018. Adversarial teacher-student learning for unsupervised domain adaptation. In *ICASSP*.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for Chinese social media with jointly trained embeddings. In *EMNLP*.

Nanyun Peng and Mark Dredze. 2017. Multi-task domain adaptation for sequence tagging. In *Repl4NLP*.

Xipeng Qiu, Peng Qian, and Zhan Shi. 2016. Overview of the NLPCC-ICCPOL 2016 shared task: Chinese word segmentation for micro-blog texts. In *ICCPOL*.

Weischedel Ralph, Palmer Martha, and Marcus et al. Mitchell. 2013. Ontonotes release 5.0 ldc2013t19. Linguistic Data Consortium.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *EMNLP*.

Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. 2005. To transfer or not to transfer. *NIPS*.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization. In *EMNLP*.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*.

Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018. Investigating capsule networks with dynamic routing for text classification. In *EMNLP*.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *EMNLP*.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Type-supervised domain adaptation for joint segmentation and pos-tagging. In *ACL*.

Shiqi Zhang and Deyi Xiong. 2018. Sentence weighting for neural machine translation domain adaptation. In *COLING*.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. In *ACL*.

Guorui Zhou, Ying Fan, Runpeng Cui, Weijie Bian, Xiaoqiang Zhu, and Kun Gai. 2018. Rocket launching: A universal and efficient framework for training well-performing light net. In *AAAI*.

Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, XIN-YU DAI, and Jiajun Chen. 2017. Word-context character embeddings for Chinese word segmentation. In *EMNLP*.