Multi-task Multi-domain Representation Learning for Sequence Tagging

Nanyun Peng* and **Mark Dredze*#**
*Human Language Technology Center of Excellence
Center for Language and Speech Processing
Johns Hopkins University, Baltimore, MD, 21218
Bloomberg LP, New York, NY
npeng1@jhu.edu, mdredze@cs.jhu.edu

Abstract

Representation learning with deep models have demonstrated success in a range of NLP. In this paper we consider its use in a multi-task multi-domain setting for sequence tagging by proposing a unified framework for learning across tasks and domains. Our model learns robust representations that yield better performance in this setting. We use shared CRFs and domain projections to allow the model to learn domain specific representations that can feed a single task specific CRF. We evaluate our model on two tasks – Chinese word segmentation and named entity recognition – and two domains – news and social media – and achieve state-of-the-art results for both social media tasks.

1 Introduction

Natural language processing tasks require effective representations of natural language inputs. These representations can arise from feature design or from learning directly from data. New deep learning models have quickly become standard methods for learning representations from data, and have yielded improvements in many tasks, including speech processing (Deng et al., 2010; Mohamed et al., 2012; Dahl et al., 2012), language modeling (Bengio, 2008; Mikolov et al., 2011; Mikolov and Zweig, 2012) and sequence tagging tasks (Collobert and Weston, 2008; Collobert et al., 2011; Zheng et al., 2013; Pei et al., 2014; Chen et al., 2015; Lample et al., 2016; Peng and Dredze, 2016; Yang et al., 2016; Ma and Hovy, 2016). The success of these methods opens new opportunities for exploring how representation learning can benefit different NLP learning settings.

In this paper we consider two settings: multi-task learning and multi-domain learning. The goal of multi-task learning (MTL) is to improve performance on several different tasks by jointly learning across them (Caruana, 1997). By using multiple tasks, we can more readily identify informative features (Ando and Zhang, 2005) or better learn representations of the data (Collobert et al., 2011; Liu et al., 2016; Peng and Dredze, 2016; Yang et al., 2016) that lead to better generalization.

A second setting we consider is multi-domain learning (MDL), in which we learn a predictor for a single task that we wish to produce for multiple different input domains. Normally, we would train a model on all available data for a given task, but in this case, distribution changes in the different input domains makes generalizing across them difficult. While there have been numerous papers on multi-domain learning, and its variant domain adaptation, most fall under two approaches: parameter tying (Dredze and Crammer, 2008; Daumé III, 2007; Daumé III, 2009; Finkel and Manning, 2009; Kumar et al., 2010; Dredze et al., 2010) and learning cross domain representations (Blitzer et al., 2006; Blitzer et al., 2007; Glorot et al., 2011; Chen et al., 2012; Yang and Eisenstein, 2015). This line of work demonstrates that in both multi-task and multi-domain settings, learning better representations effectively improves accuracy across both tasks and domains.

We propose a unified framework that jointly learns representations with deep architectures for both multiple tasks and domain simultaneously. We consider sequence labeling tasks since there have been several demonstrations that they benefit from learning representations. Our framework learns a singe model to produce representations for all domains and tasks based on a Recurrent Neural Network (RNN), which is then used as the input to a Conditional Random Fields (CRF) (Lafferty et al., 2001) to predict the output sequence. We propose both the use of a single CRF for each task across domains and a domain projection layer that enables the RNN to learn domain specific representations. We consider two tasks – Chinese word segmentation and named entity recognition (NER) – and two domains – news and social media. We consider these tasks together since performance of the latter often relies on accurate predictions of the former (Peng and Dredze, 2016). Furthermore, our framework is generally applicable for structured learning problems in NLP which have shown benefit from learning representations with deep network architectures.

Our contributions are threefold: 1) We propose a novel deep architecture for multi-task multi-domain sequence tagging; 2) We evaluate learning simultaneously across tasks and domains, as opposed to just across tasks or domains. We also observed improvements individually in a multi-task and multi-domain setting, as well as a new mismatch setting. 3) We achieve state-of-the-art results on both tasks in the Chinese social media domain.

2 Model

We propose a general framework for multi-domain and multi-task learning focused on sequence labeling tasks. The core of our model is a bi-directional LSTM (BiLSTM) with a CRF layer that uses the learned representations as features to output a sequence of labels. This setup has been shown to learn robust representations for state-of-the-art results on sequence labeling tasks (Lample et al., 2016; Peng and Dredze, 2016; Ma and Hovy, 2016). We will consider how variations on this base configuration can benefit multi-domain and multi-task learning.

Figure 1 gives an overview of our complete model. We begin with a quick review of our base architecture BiLSTM-CRF, and then will introduce a two simplified cases: one for multi-task learning and one for multi-domain learning. We will then present the full model.

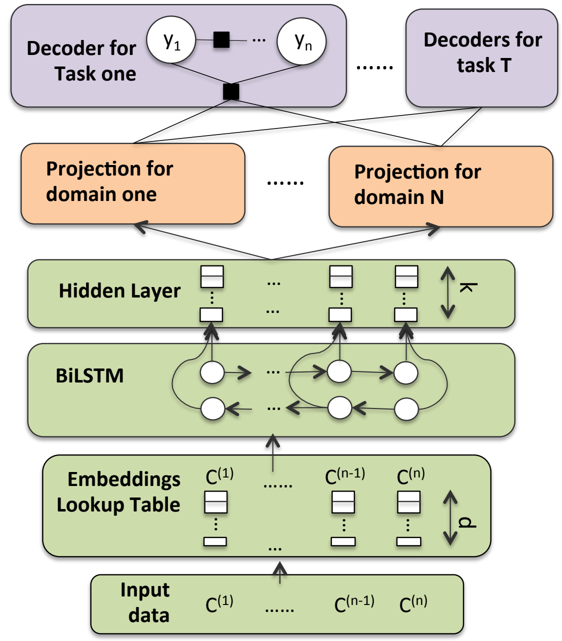

Figure 1: An overview of our proposed model framework. Colors indicate the degree to which parameters are shared. The first few layers (green) are shared by all tasks and domains. The domain projections (orange) are shared within tasks and the decoders (purple) are shared for domains.

2.1 BiLSTM-CRF for Sequence Tasks

Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a recurrent neural network (RNN) that models interactions between input vectors and hidden layers. The hidden layers are optimized to minimize a loss function based on some labeled data, which may be automatically derived from raw text (such as in language modeling). Since LSTMs learn a representation for each input in a sequence, they are popular a choice for sequence labeling tasks.

LSTMs use a series of gates (input, forget and output gate) to control how memory is propagated in the hidden states of the model. The primary motivation for LSTMs is to combat the vanish and exploding gradient problems (Bengio et al., 1994; Pascanu et al., 2013) of standard RNNs. As a result, LSTMs are effective at capturing long-distance dependencies between the inputs by building up context as they scan an input sentence, and then using that context to influence the hidden state of the next token.

Since most sequence tagging tasks benefit from both a left and right context when deciding on the label for a given token, most LSTM sequence taggers use a bi-directional model (Dyer et al., 2015). A bi-directional LSTM (BiLSTM) uses two LSTMs, one that scans left-to-right and one that operates right-to-left, and concatenates the hidden outputs of both LSTMs to form the final output vector. BiLSTMs have become a common building block for sequence tagging tasks in NLP (Lample et al., 2016; Yang et al., 2016; Ma and Hovy, 2016). The BiLSTM serves as the core of our model.

2.1.1 Neural-CRF Decoders

Although BiLSTMs capture long-distance dependencies in the input and make comprehensive local predictions, they lack a principal way of modeling interactions between the output labels, an essential component of sequence tagging tasks. Previous work addressed this problem by introducing a transition matrix for the output labels (Collobert et al., 2011; Zheng et al., 2013; Pei et al., 2014; Chen et al., 2015), or by using a traditional sequential tagger, such as a Conditional Random Field (CRF) (Lafferty et al., 2001), on top of the BiLSTM layer (Lample et al., 2016; Peng and Dredze, 2016; Yang et al., 2016; Ma and Hovy, 2016). While similar in appearance, these two approaches are conceptually different: the former is a pure neural network model that added a patch to enable effective modeling of output dependencies, while the latter utilized existing state of the art sequence taggers but substituted learned representations from a neural network in place of traditional hand crafted features. We note that in the latter case, there is no reason that the CRF cannot consider both learned representations as well as hand crafted features. Such an approach has typically yielded improved results (Peng and Dredze, 2016). We adopt the second approach, which motivates our proposal for domain projections below.

The generalized neural-CRF maximizes the conditional log-likelihood of the training labels given the data $\{(\boldsymbol{X}, \boldsymbol{Y})\}$ just as in CRFs. However, rather than rely on a given input representation, the model relies on a conversion between a sequence of tokens (or one-hot embeddings) to a sequence of low-dimensional dense vectors: $\boldsymbol{H} = \psi(\boldsymbol{X})^1$. In our case, $\psi()$ is a BiLSTM. The log-likelihood of the whole dataset is written as:

$$\mathcal{L}(\boldsymbol{Y}|\boldsymbol{X};\boldsymbol{W}) = \sum_k \log p\big(\boldsymbol{y}^k|\psi(\boldsymbol{x}^k);\boldsymbol{W}\big), \quad (1)$$

where k indexes instances, and

$$\begin{aligned} & p(\boldsymbol{y}^k|\psi(\boldsymbol{x}^k);\boldsymbol{W}) \\ & = \frac{\prod_{i=1}^n \exp\big(\boldsymbol{W}_i^T F_i(y_{i-1}^k, y_i^k, \psi(\boldsymbol{x}^k))\big)}{Z^k}, \end{aligned} \quad (2)$$

defines the probability of the kth instance given the parameters $\boldsymbol{W}$. i indexes the position in the sequence, F_i is the feature function, Z^k is the partition function. The feature function depends on the learned hidden representations $\boldsymbol{h} = \psi(\boldsymbol{x})$ produced by a BiLSTM.

2.2 A Model for Multi-task Learning

Yang et al. (2016) proposed a multi-task learning framework based on the BiLSTM-CRF model. They train a single BiLSTM for several tasks and use the learned representations with task specific CRFs. The shared BiLSTM is trained with the feedback provided by all the task specific CRFs. The parameters of the word embeddings, BiLSTM and the CRFs are jointly learned through back-prorogation.

Their setting can be viewed as a special case of ours by removing the projection layer (orange) from our model and adding more decoders for the same task from different domains. We extend their model to the setting of multi-domain learning, where the task remains the same but the input data changes.

2.3 A Model for Multi-domain Learning

The goal in multi-domain learning, where training data is distributed across domains, and domain adaptation, where training data comes mostly or exclusively for one domain, is to learn a single model that can produce accurate predictions for different input domains. An important characteristic of learning across domains is that each domain represents data drawn from a different distribution. The larger the difference between these distributions the larger the

[1] The function $\psi()$ denotes an element-wise operation.

generalization error when learning a single model across domains (Ben-David et al., 2010; Mansour et al., 2009).

As a result, a long line of work in multi-domain learning has been on learning shared representations, such as through identifying alignments between features (Blitzer et al., 2007; Blitzer et al., 2006), learning shared representations with deep networks (Glorot et al., 2011), using transfer component analysis (Pan et al., 2011), learning feature embeddings (Yang and Eisenstein, 2015) and kernel methods for learning low dimensional domain structures (Gong et al., 2012), among many others. Given the importance of learning representations that map domains to the same shared space, we seek a mechanism that encourages our model to infer a mapping when learning across domains.

We can directly apply the multi-task model from §2.2 to the multi-domain setting. where we use multiple domain specific CRFs. However, we suggest two extensions to the model to better handle properties of the multi-domain setting: sharing CRF decoders and domain projections. In this view, the multi-domain learning setting is a special case of our model achieved by sharing all parameters in the CRF decoder layer (purple).

2.4 Sharing CRF Decoders

In multi-task learning we use a separate CRF decoder for each task since different tasks will naturally have different label spaces. However, when we want to learn a model across domains for the same task, which share the label space, we have a choice. Since each domain exhibits its own behaviors, we could use a separate CRF for each domain. As in multi-task learning, the CRFs will benefit from sharing the same representation even though they learn their own output parameters.

Instead, we favor using a single CRF decoder for a task, regardless of the number of input domains. This draws from the tradition in multi-domain learning where we favor learning a single model across domains to increase the amount of training data, even though it introduces errors in learning due to shifts in domain representations (Dredze et al., 2010; Li and Zong, 2008; Daumé III, 2007; Joshi et al., 2012). However, since in our model the representations are themselves learned, we can rely on the BiLSTM model to learn an effective shared representation across the different domains. By back-propagating errors from the CRF through the BiLSTM for both domains, we encourage the BiLSTM to generalize its representation.

2.5 Domain Projections

As discussed in the previous section, by sharing a CRF decoder across domains we can encourage the BiLSTM to learn an effective cross domain representation. However, this may place a heavy burden on the BiLSTM; it does not know the identity of each domain yet must still learn how to map two different input types to the same representation.

Instead, we consider a strategy of domain projections where we rely on an explicit domain specific transformation function to produce a shared representation. We place this transformation layer between the BiLSTM and the CRF, which alleviates pressure on the BiLSTM to learn a cross domain representation. With this added layer, the BiLSTM is free to produce distinct representations for each domain as it sees fit. For example, it may use part of the hidden states for representing domain A, another part for domain B, and a third part for sharing between the two. Then, the domain specific projection will map these hidden states to a single representation for learning. We rely on back-propagation to encourage the model to find effective projections.

We experiment with two strategies for domain projections.

2.5.1 Domain Masks

The first strategy is inspired by Daumé III (2007) and Yang and Hospedales (2015), which split the features (representations) into different regions, with one region shared among domains, and others specific for each domain. We implement this strategy by defining several masks, where $\boldsymbol{m}_d$ is a vector for the dth domain. The mask $\boldsymbol{m}_d$ has value 1 for the effective dimensions of domain d, and 0 for all other values. We can then apply this mask directly to the hidden states $\boldsymbol{h}$ learned by the BiLSTM to produce a projected hidden state $\hat{\boldsymbol{h}}$:

$$\hat{\boldsymbol{h}} = \boldsymbol{m}_d \odot \boldsymbol{h}, \tag{3}$$

where $\odot$ denotes element-wise multiplication.

For example, assume we have two domains A and B and a $3k$ dimensional hidden state space. The first k-dimensions will be shared between the two domains, while the $k+1$ to $2k$ dimensions are used only for domain A, and the remaining dimensions for domain B. The mask for domain A would be $\boldsymbol{m}_1 = [\vec{1}, \vec{1}, \vec{0}]$, and for domain B $\boldsymbol{m}_2 = [\vec{1}, \vec{0}, \vec{1}]$. Since only a subset of the dimensions are included in a projection for each domain, the BiLSTM is free to use those exclusively for each individual domain. In other words, the training prediction errors for domain A will only influence gradients that propagate through a subset of the BiLSTM parameters, allowing the BiLSTM to partition its parameters among the input domains.

2.5.2 Linear Projection

Our first strategy forced a division of parameters between the domains. Another strategy relaxes this assumption and provides the model greater freedom to chose how to use each domain. In this approach, we apply a linear transformation for each domain, denoted as T_d. Given a k-dimensional vector representation $\boldsymbol{h}$, T_d is a $k \times k$ matrix that projects the learned BiLSTM hidden state to a domain specific representation as:

$$\hat{\boldsymbol{h}} = T_d \boldsymbol{h}. \tag{4}$$

We learn T_d for each domain through backpropagation with the other model parameters. This projection works just like the previous domain mask approach, except that the model has greater freedom in choosing how to use each of the model parameters for learning across domains.

2.6 A Model for Multi-task Multi-domain Learning

While our model can be used for the multi-task or multi-domain setting, it is straightforward to combine them to form our full model that handles both settings jointly. The model relies on the shared BiLSTM to learn representations for different domains and tasks. The full model has several projections (one for each domain), and several decoders (one for each task). Thus, it is flexible in both the number of tasks and domains. Increasing the number of domains linearly increases the number of domain projection parameters, but the number of other model parameters remain constant. Increasing the number of tasks linearly increases the number of CRF parameters, with the number of other parameters unchanged. Our experiments will explore a variety of multi-task and multi-domain settings.

To train the model, we optimize the log-probabilities of the labels in the training data and have the gradients of CRFs backpropagate through the entire network to learn the parameters of the CRFs, the domain projections, the BiLSTM and the word embeddings, all jointly.

3 Parameter Estimation

Model training is relatively straightforward and follows standard applications of gradient based learning and backpropogation.

We have several different types of model output to use for computing the gradients: we have a different CRF for each task and we have multiple sets of training examples, one for each domain. Therefore, we use an alternating optimization strategy, alternating between each dataset. We use stochastic gradient descent (SGD) with a separate learning rate tuned for dataset. We decay the learning rate when results on development data do not improve after 10 consecutive epochs. We train for 50 epochs and use early stopping (Giles, 2001; Graves et al., 2013) as measured on development data. We learn a model for each dataset based on hyperparameter tuning for that dataset. We use dropout between the word embeddings and the BiLSTM.

Initialization We pre-train the word embeddings as described Peng and Dredze (2015), where we use Chinese character embeddings (Peng and Dredze, 2015). These embeddings are updated during learning by fine-tuning. We use a single set of embeddings for all tasks, in this case those trained on Weibo (social) data. We plan to consider more effective word representation learning for this setting in future work. All other model parameters are initialized uniformly at random in the range $[-1, 1]$.

Inference For training the CRFs we use marginal inference and maximize the marginal probabilities of the labels in the training data. At test time, the label sequence with highest conditional probability $y^* = \arg\max p(y|x; \Omega)$ is obtained by MAP inference. Both of these can be efficiently computed with dynamic programming.

Subsampling The news data is significantly ($\sim$100 times) larger than data from the social domains. Training on the raw data would heavily skew the model towards the news domain. Therefore, during training we subsample the number of instances in the news domain based on the fraction λ, which is tuned as a hyper-parameter on development data. We use different randomly selected subsamples for each epoch of training.

Hyper-parameters Our hyper-parameters include the initial learning rate (per dataset), the dropout rate for the input embedding and the hidden vectors, and the subsample coefficient for the news domain. We tune these hyper-parameter using beam search on development data, where different datasets may use different hyper-parameters to train the model. The embedding and the hidden state dimensions are set to 100 and 150 respectively, following previous work (Chen et al., 2015; Peng and Dredze, 2016).

4 Experimental Setup

To demonstrate the flexibility and effectiveness of our framework to learn across domains and tasks, we experiment on two Chinese domains (news and social) and two tasks (word segmentation and named entity recognition (NER)). We split each of the four datasets into train, development and test. We use the development set for hyper-parameter tuning and model development, and report the F1 results for test data.

We begin by describing each dataset, followed by the baselines. We then explain each experimental setting, which correspond to different configurations of our model.

4.1 Datasets

We experiment with four Chinese datasets.

SighanSeg (news) comes from the SIGHAN 2005 shared task on Chinese word segmentation (Emerson, 2005). We used the PKU portion, which is simplified Chinese and includes 43,963 sentences as training and 4,278 sentences as test. Within the training data, we hold out the last $\frac{1}{10}$ for development. We did not apply any special preprocessing.

SighanNER (news) comes from SIGHAN 2006 shared task for Chinese NER (Levow, 2006), which contains three entity types (person, organization and location). We used the MSR portion, the only corpus in simplified Chinese, which includes 18,683 sentences as training and 4,636 sentences as test. Within the training data, we hold out the last $\frac{1}{10}$ for development.

WeiboSeg (social) is a word segmentation dataset created by Zhang et al. (2013a). It contains 2,000 Sina Weibo messages[2]. They used the entire dataset for test so we split it ourselves using an 8:1:1 split.

WeiboNER (social) is a NER dataset created by Peng and Dredze (2015) that contains 1,890 Sina Weibo messages annotated with four entity types (person, organization, location and geo-political entity), including named and nominal mentions. To match SIGHAN 06 we only use named mentions and merge geo-political entities and locations. We use the same split as Peng and Dredze (2015).

4.2 Baseline

As is standard in multi-task and multi-domain work, we compare our model with a baseline that only considers a single dataset at a time. We train a *separate* BiLSTM-CRF model for each domain and task; no parameters are shared across these settings. We will also include the best published result for each dataset in our comparisons.

4.3 Model Variations

To demonstrate the flexibility of our model we evaluate four settings: 1) multi-task multi-domain, where we train across all four datasets, 2) multi-domain, 3) multi-task, and 4) a data mismatch setting, where we have data from two different tasks in two different domains.

Multi-task multi-domain We begin with the fully joint model: a multi-task multi-domain setting that trains a model for both news and social and for word segmentation and NER. All four datasets jointly train a single BiLSTM, followed by a domain transformation layer where each domain is projected using domain specific parameters and fed into a task-

[2] Zhang et al. (2013a) report 2,038 messages, but the dataset we obtained from the authors has only 2,000 messages.

Methods \ Datasets	News						Social					
	Segmentation			NER			Segmentation			NER		
	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
Separate	95.5	94.6	95.0	88.6	85.7	87.1	86.2	85.7	86.0	53.4	32.5	40.4
BiLSTM+Separate CRF	95.2	93.7	94.4	86.6	84.4	85.5	88.6	87.0	87.8	58.6	38.7	46.6
Domain Mask	95.3	94.2	94.7	87.1	85.2	86.1	89.2	88.8	89.0	60.8	**44.9**	**51.7**
Linear Projection	95.0	94.0	94.5	86.5	84.2	85.3	**89.3**	**89.0**	**89.1**	**67.2**	41.2	51.1
Best Published	**96.6**	**96.4**	**96.5**[1]	**91.6**	**88.7**	**90.2**[2]	87.5	87.5	87.5[3]	63.3	39.2	48.4[4]

Table 1: Test results for our model trained jointly on all four datasets. The first two rows are baselines (§4.2), followed by our proposed methods. For reference, we include the best published result on each dataset ([1]Chen et al. (2015); [2]Zhang et al. (2013b); [3]Zhang et al. (2013a); [4]Peng and Dredze (2016))

Task \ Domain	News			Social		
	Prec	Recall	F1	Prec	Recall	F1
Segmentation: Separate	**95.5**	**94.6**	**95.0**	86.2	85.7	86.0
Domain Mask	95.2	94.2	94.7	88.4	87.3	87.8
Linear Projection	95.3	94.2	94.7	**88.7**	**87.8**	**88.3**
NER: Separate	**88.6**	**85.7**	**87.1**	53.4	32.5	40.4
Domain Mask	86.7	84.6	85.6	**63.1**	39.7	**48.7**
Linear Projection	85.7	82.5	84.0	51.6	**41.8**	46.1

Table 2: Results for multi-domain learning.

Domain \ Task	Segmentation			NER		
	Prec	Recall	F1	Prec	Recall	F1
News: Separate	**95.5**	**94.6**	**95.0**	**88.6**	**85.7**	**87.1**
BiLSTM+Separate CRF	95.3	94.3	94.8	85.6	83.9	84.7
Social: Separate	86.2	85.7	86.0	53.4	32.5	40.4
BiLSTM+Separate CRF	**86.3**	**86.2**	**86.2**	**56.6**	**33.0**	**41.7**

Table 3: Results for multi-task learning

	Prec	Recall	F1	Prec	Recall	F1
	News Segmentation			Social NER		
Separate	**95.5**	**94.6**	**95.0**	53.4	32.5	40.4
Mismatch	94.8	93.8	94.3	**59.1**	**33.5**	**42.8**
	News NER			Social Segmentation		
Separate	**88.6**	85.7	**87.1**	86.2	**85.7**	**86.0**
Mismatch	87.1	**86.1**	86.6	**86.3**	84.6	85.4

Table 4: Results on mismatch setting.

specific CRF decoder. We experiment with two different methods for domain transformation: 1) domain masks, and 2) linear projections. We also compare to a model with a separate CRF for each task and domain (BiLSTM+Separate CRF) (Yang et al., 2016).

Multi-domain Next, we consider a multi-domain setting in which we build one model per task, but the model is shared across the news and social domains. In this setting, we use a shared BiLSTM for the two domains, followed by a domain transformation layer and a shared CRF decoder.

Multi-task The multi-task setting learns a shared BiLSTM for both tasks within a domain, and then trains a separate CRF decoder for each task on the resulting representation. Without domain projections or shared CRF parameters, this is the model of (Yang et al., 2016).

Data mismatch Finally, we consider a setting where we have mismatched training data. We train a model on a pair of datasets, where they reflect both different domains and different tasks. Following our approach, we train a BiLSTM on these two datasets, and use a separate CRF decoder for each task.

5 Results

Multi-task multi-domain Table 1 presents results for the multi-task multi-domain setting, where we train the full model on all four datasets. Looking first to the BiLSTM+Separate CRF, whether it improves over the baseline is a reflection of the amount of training data. For both news tasks, separate models do better than the BiLSTM+Separate CRF, most likely because we have large amounts of training data and little can be gained from adding new tasks and domains. This is typical of multi-domain learning where large amounts of training data yield better models than can be achieved through learning from multiple domains (Dredze et al., 2010). In contrast, the BiLSTM+Separate CRF improves over the baseline on social, showing particularly large improvements for NER. This may be due to the social training sets are small, so social benefits from having the additional supervision from the news data.

The patterns of improvement continue when we consider our projection methods. For news, neither a domain mask or a linear projection beat the separate baseline (same as BiLSTM+Separate CRF), but projections improve over BiLSTM+Separate CRF, showing the benefit to the projections and sharing of CRF parameters. When there is sufficient data, training across tasks and domains is the wrong strategy, our model mitigates the loss in accuracy in this setting. In contrast, we see large improvements on both social tasks, which have much less training data. Both of our methods improve by a wide margin over the baseline and BiLSTM+Separate CRF.

Finally, we compare these improved results to the best published work on each dataset. For the news datasets, our framework comes close to the best published results but does not match it. For segmentation, our lack of task specific preprocessing is likely the source of the difference[3]. For NER, it seems a BiLSTM model may not be as good as the previous feature rich CRF approach which used resources like gazetteers. Perhaps improvements on how we learn embeddings for a multi-domain setting would close this gap. For the social datasets, we set a new state-of-the-art result with our methods.[4] We hypothesize that we could get even better results for the NER task following the strategy of Peng and Dredze (2016) who augmented an LSTM based model with hand engineered features. We leave this extension to future work.

Multi-domain Focusing only on the multi-domain setting, Table 2 compares our projection models to the baseline for each task separately, when trained over both domains. Not surprisingly, since we have less data (only two domains for each task) we do worse than when training on all four data sets. However, we still see considerable improvements in soical domain; our model still does well even with a single task.

Multi-task Similarly, in the multi-task setting (Table 3) we again see improvements in social domain. However, the improvements are less than in the multi-domain setting, suggesting that its better to have multiple datasets for the same task than multiple tasks from the same domain.

Mismatch Finally, we consider the setting where we have two datasets from different domains and tasks (Table 4). We still see some improvements on the social NER, but not for social segmentation. One hypothesis could be more data for up-stream tasks helped down-stream tasks, but not vice-versa.

6 Conclusion

We have presented a deep architecture that combines a BiLSTM and a CRF to learn a joint representations for a multi-task multi-domain sequence learning setting. We use a single CRF per task, across domains, to share the learned model parameters. To ensure all domains have a consistent representation to use as feature for the CRF, we use domain-specific projections of the hidden states from the BiLSTM to obtain representations for the CRF. These projections provide the BiLSTM the flexibility to learn domain specific representations based on distributional shifts in the input data. Our model yields new state-of-the-art results for Chinese word segmentation and NER for social media.

With this model framework in mind, there are a number of interesting research directions. First, our method is general and can be applied to other sequence tagging tasks, a larger number of tasks and domains, other languages and other structure prediction problems in NLP. We plan to pursue these in future work. Second, we assume a uniform embedding representation for all domains and tasks. Yet we know that task-specific fine tuning and learning of embeddings can yield improvements (Faruqui et al., 2015; Yu and Dredze, 2014). Our model would benefit from a means of learning cross domain word embeddings. Third, domain specific projections are clearly beneficial to multi-domain learning. We've explored two such projections, but there are many more sophisticated ones that we can consider. Finally, our work draws on two traditions in multi-domain learning: parameter sharing (our shared CRFs) and representation learning (the BiLSTM). We plan to explore how other methods from this literature can be applied to a deep architecture.

[3]When we added segmentation specific preprocessing, our BiLSTM achieved comparable results to the best published result (which is an LSTM model). However, since this preprocessing is specific to word segmentation, and not NER, we omit it in our work.

[4]The segmentation results are not a fair comparison since the best published work did not use in-domain training data, and we use a different data split.

References

[Ando and Zhang2005] Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*.

[Ben-David et al.2010] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*.

[Bengio et al.1994] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*.

[Bengio2008] Yoshua Bengio. 2008. Neural net language models. *Scholarpedia*.

[Blitzer et al.2006] John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*.

[Blitzer et al.2007] John Blitzer, Mark Dredze, Fernando Pereira, et al. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of Association for Computational Linguistics (ACL)*.

[Caruana1997] Rich Caruana. 1997. Multitask learning. *Machine learning*.

[Chen et al.2012] Minmin Chen, Zhixiang Xu, Fei Sha, and Kilian Q Weinberger. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*.

[Chen et al.2015] Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. Long short-term memory neural networks for chinese word segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Collobert and Weston2008] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning (ICML)*.

[Collobert et al.2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*.

[Dahl et al.2012] George E Dahl, Dong Yu, Li Deng, and Alex Acero. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*.

[Daumé III2007] Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of Association for Computational Linguistics (ACL)*.

[Daumé III2009] Hal Daumé III. 2009. Bayesian multitask learning with latent hierarchies. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*.

[Deng et al.2010] Li Deng, Michael L Seltzer, Dong Yu, Alex Acero, Abdel-Rahman Mohamed, and Geoffrey E Hinton. 2010. Binary coding of speech spectrograms using a deep auto-encoder. In *Interspeech*.

[Dredze and Crammer2008] Mark Dredze and Koby Crammer. 2008. Online methods for multi-domain learning and adaptation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

[Dredze et al.2010] Mark Dredze, Alex Kulesza, and Koby Crammer. 2010. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*.

[Dyer et al.2015] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of Association for Computational Linguistics (ACL)*.

[Emerson2005] Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*.

[Faruqui et al.2015] Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015. Retrofitting word vectors to semantic lexicons. In *North America Chapter of Association for Computational Linguistics (NAACL)*.

[Finkel and Manning2009] Jenny Rose Finkel and Christopher D Manning. 2009. Hierarchical bayesian domain adaptation. In *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*.

[Giles2001] Rich Caruana Steve Lawrence Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems (NIPS)*.

[Glorot et al.2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*.

[Gong et al.2012] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*.

[Graves et al.2013] Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*.

[Joshi et al.2012] Mahesh Joshi, William W Cohen, Mark Dredze, and Carolyn P Rosé. 2012. Multi-domain learning: when do domains matter? In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*.

[Kumar et al.2010] Abhishek Kumar, Avishek Saha, and Hal Daume. 2010. Co-regularization based semi-supervised domain adaptation. In *Advances in neural information processing systems*.

[Lafferty et al.2001] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*.

[Lample et al.2016] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *North America Chapter of Association for Computational Linguistics (NAACL)*.

[Levow2006] Gina-Anne Levow. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, Sydney, Australia. Association for Computational Linguistics.

[Li and Zong2008] Shoushan Li and Chengqing Zong. 2008. Multi-domain sentiment classification. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies (ACL)*.

[Liu et al.2016] Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. 2016. Implicit discourse relation classification via multi-task neural networks. In *Proceedings of the 30th AAAI Conference on Artifi- cial Intelligence (AAAI)*.

[Ma and Hovy2016] Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of Association for Computational Linguistics (ACL)*.

[Mansour et al.2009] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory (COLT)*.

[Mikolov and Zweig2012] Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *SLT*.

[Mikolov et al.2011] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Honza Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*.

[Mohamed et al.2012] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton. 2012. Acoustic modeling using deep belief networks. *Audio, Speech, and Language Processing, IEEE Transactions on*.

[Pan et al.2011] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. 2011. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Transactions on*.

[Pascanu et al.2013] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of The 30th International Conference on Machine Learning*.

[Pei et al.2014] Wenzhe Pei, Tao Ge, and Chang Baobao. 2014. Maxmargin tensor neural network for chinese word segmentation. In *Proceedings of Association for Computational Linguistics (ACL)*.

[Peng and Dredze2015] Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisboa, Portugal, September.

[Peng and Dredze2016] Nanyun Peng and Mark Dredze. 2016. Improving named entity recognition for chinese social media with word segmentation representation learning. In *Association for Computational Linguistics (ACL)*.

[Yang and Eisenstein2015] Yi Yang and Jacob Eisenstein. 2015. Unsupervised multi-domain adaptation with feature embeddings. In *Proceedings of North America Chapter of Association for Computational Linguistics (NAACL)*.

[Yang and Hospedales2015] Yongxin Yang and Timothy M Hospedales. 2015. A unified perspective on multi-domain and multi-task learning. *Proceedings of the 2015 International Conference on Learning Representations (ICLR)*.

[Yang et al.2016] Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *CoRR*, abs/1603.06270.

[Yu and Dredze2014] Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of Association for Computational Linguistics (ACL)*.

[Zhang et al.2013a] Longkai Zhang, Li Li, Zhengyan He, Houfeng Wang, and Ni Sun. 2013a. Improving chinese word segmentation on micro-blog using rich punctuations. In *Proceedings of Association for Computational Linguistics (ACL)*.

[Zhang et al.2013b] Longkai Zhang, Houfeng Wang, Xu Sun, and Mairgup Mansur. 2013b. Exploring representations from unlabeled data with co-training for chinese word segmentation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

[Zheng et al.2013] Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for chinese word segmentation and pos tagging. In *Empirical Methods in Natural Language Processing (EMNLP)*.