PAPER

Adversarial Domain Adaptation Network for Semantic Role Classification

Haitong YANG[†a)], **Guangyou ZHOU**[†], **Tingting HE**[†], ***Nonmembers***, ***and*** **Maoxi LI**[††], ***Member***

SUMMARY In this paper, we study domain adaptation of semantic role classification. Most systems utilize the supervised method for semantic role classification. But, these methods often suffer severe performance drops on out-of-domain test data. The reason for the performance drops is that there are giant feature differences between source and target domain. This paper proposes a framework called Adversarial Domain Adaption Network (ADAN) to relieve domain adaption of semantic role classification. The idea behind our method is that the proposed framework can derive domain-invariant features via adversarial learning and narrow down the gap between source and target feature space. To evaluate our method, we conduct experiments on English portion in the CoNLL 2009 shared task. Experimental results show that our method can largely reduce the performance drop on out-of-domain test data.

key words: *argument classification, domain adaption, adversarial domain adaptation, supervised learning*

1. Introduction

Semantic Role Labeling (SRL) is an important fundamental task in Natural Language Processing (NLP) community and its goal is to assign a formal semantic structure for each predicate of a given sentence, like WHO did WHAT to WHOM, WHEN, WHERE, WHY, HOW. The simple and pure form of SRL representations has been proved being beneficial to many NLP tasks, such as question and answering [1], information extraction [2], [3] and machine translation [4]–[8].

Currently, most SRL systems take supervised methods to perform semantic role classification. But, the supervised methods often suffer severe performance drops on out-of-domain test data due to giant feature differences between source and target domain. This problem is usually called domain adaption. In this paper, we focus on unsupervised adaption in which besides annotated data from source domain, we also have some unlabeled data from target domain. Our goal is to learn discriminative and domain-invariant from labeled and unlabeled data so that the learnt model can be adapted to the target domain.

There have been some works about domain adaption in NLP community. One line is to use the pivot features to induce a projected feature space in which source and target domain are similar. Blitzer et al. [9] proposed a method called Structural Correspondence Learning (SCL), which works well in cross-domain sentiment classification. Following their work, some research aims to learn better domain-specific words [10], [11] such that the domain discrepancy could be reduced. Kim et al., [12] proposed a method of feature augmentation to make different domains look similar. Although these works reported promising results, they have one limit that the performance depends heavily on the heuristic selection of pivot features. Moreover, the pivot features may be sensitive to different applications. The other research line is to map source and target domain into a common feature space. Yang et al., [13] utilized Deep Belief Networks (DBN) to learn the common features across domains. And then the learnt features are fed into a role classifier like Maximum Entropy.

Our method falls into the second line. But, different from Yang et al., [13], we try to perform deep feature learning and role classification in the same framework together. Our idea is driven by the theory [14] that a good feature representation for domain adaption is the one for which an algorithm cannot learn to identify the domain of the input observation. Specifically, in this paper, we propose a framework called Adversarial Domain Adaptation Network (ADAN) to address domain adaptation problem of SRL. The ADAN framework consists of two core components: (i) role classifier that predicts the label of a given sample; (ii) domain discriminator that predicts whether a sample is from source domain or target. The two components have different objectives. The optimization objective of the role classifier is to minimize the classification errors on the training set which can make the learnt features be discriminative for the final decision while the optimization objective of the domain discriminator is to maximize the domain classification errors which can encourage the domain-invariant features to emerge. In the implementation (see Fig. 1), we first use a Bi-LSTM layer to encoder the input sequence and then feed the hidden states into two full connected layers: the role classifier and the domain discriminator. The model is trained by optimizing the two components jointly. It is noted that the training set contains data from source domain and target domain and there are only unlabeled data in the target domain. If the input sample is from target domain, we train the model only by optimizing the domain discriminator. We carried out experiments on the out-domain data of the CoNLL 2009 shared task. The experimental results show that compared with the existing systems, our method can largely reduce

Manuscript received March 28, 2019.
Manuscript revised July 12, 2019.
Manuscript publicized September 2, 2019.

[†]The authors are with School of Computing, China Central Normal University, Wuhan, China.
[††]The author is with the School of Computer Information Engineering, Jiangxi Normal University, Nanchang, 330022, China.
a) E-mail: htyang@mail.ccnu.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2019EDP7087

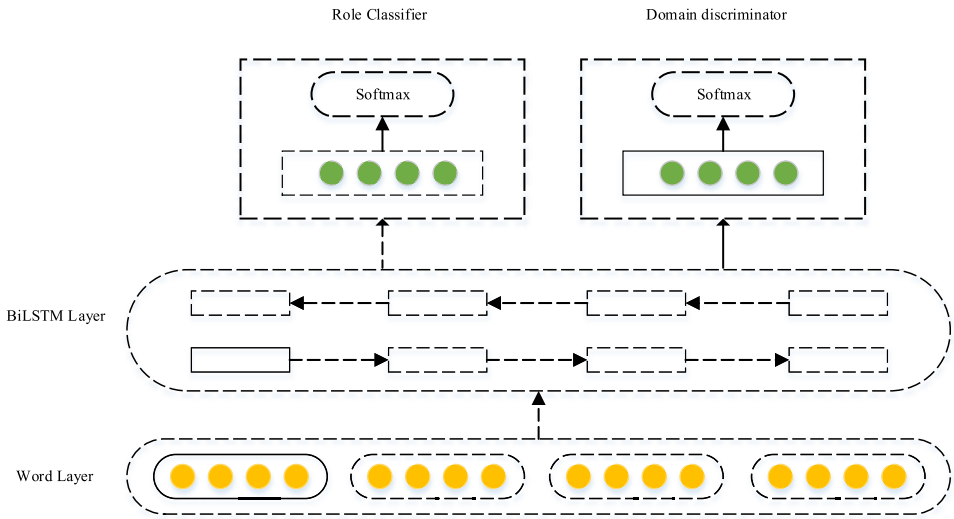

Fig. 1 The overview of our approach.

the performance drop on out-of-domain test data.

The remainder of this paper is organized as follows. Section 2 introduces the related works. The proposed method is presented in Sect. 3. The experiments and results are presented in Sect. 4. Finally, the conclusion is shown in Sect. 5.

2. Related Work

2.1 Semantic Role Labeling

Semantic Role Labeling (SRL) is an important basic task in NLP community. Gildea and Jurafsky [15] proposed the first SRL system which casts SRL as the classification task of machine learning and defines many manual features for the classifier. After Gildea and Jurafsky [15], many attentions are payed on feature engineering [16]–[20], efficient inference [21], [22].

Recent years have witnessed the great success of deep neural network in Computer Vision (CV) and Natural Language Processing (NLP). A series of neural model have been designed for SRL. Collobert and Weston [23] employed a Multi-Layer Perceptron (MLP) to fastly parse a sentence. Based on Convolutional Neural Network (CNN), Collobert et al. [24] proposed a multi-tasks architecture that can perform Part-Of-Speech Tagging, Chunking, Named Entity Recognition and Semantic Role Labeling jointly. Zhou and Xu [25] introduced deep bi-directional recurrent network as an end-to-end system for SRL that performed better that the previous state-of-the-art system. Similarly, Marcheggiani et al. [26] also proposed a syntax-agnostic model for dependency SRL and obtained favorable results. These deep model do not use syntax information. Despite the success of syntax-agnostic models, many works [15]–[22] think syntax information is useful for SRL and there have been several works which focus on leveraging the advantages of syntax. Roth and Lapata [27] embed dependency path as syntactic information into a model and exhibited a notable success. Li et al. [28] extended existing models and proposed a unified framework to investigate more effective and more diverse ways of incorporating syntax into sequential neural networks. Different from the above works, our work is constructed based on a Bi-LSTM neural model and an adversarial domain classifier is incorporated into the model to enhance the domain-invariant features to emerge.

2.2 Domain Adaption

Here, we review the works related to domain adaption in NLP. Blitzer et al. [9] proposed a model called structural correspondence learning (SCL) and they use the pivot features to induce a projected feature space in which source and target domain are similar. Following the work, Pan et al. [10] proposed a graph-based model to exploit the relations between the pivot features and the non-pivot ones. Although these works report promising results, they have one limit that the performance depend heavily on the heuristic selection of pivot features. Moreover, the pivot features are sensitive to different applications.

In the community of SRL, there have been some related works. Huang and Yates [29] leveraged latent-variable language models to learn kinds of features that are useful for SRL in out-of-domain test set. Do et al. [30] tried to replace important words from semantic frames in the training set by words in the vocabulary of the testing domain to create new semantic frames which are closer to the testing set, which can improve the generalization power of the SRL classier but the perform of their method rely on many linguistic resources such as WordNet. Based on distributed word representations, Hartmann et al. [31] developed a straightforward approach to frame identification of FrameNet-style SRL and thanks to the generalization power of distributed word representations, their method achieved good results on out-of-domain test. Yang et al., [13] utilized Deep Belief Networks

(DBN) to learn the common features across domains. But, their method use over 1M one-hot features as the input of DBN which cause DBN to be hard to train. Different from the above works, this paper focuses on address the domain adaption of SRL through adversarial learning.

2.3 Generative Adversarial Networks

Generative Adversarial Networks (GAN) was originally introduced in the community of Computer Vision. Goodfellow et al. [32] firstly proposed the idea of GAN in which a generator and a discriminator are constructed and the generator can produce the noise data while the discriminator should classify whether the data is real or noisy. They reported promising results in the task of image classification and proved that adversarial networks can narrow two different distributions. Inspired by the idea of Goodfellow et al. [32], several variants of GAN such as DCGAN [33], Cycle-GAN [34] has been explored to improve the performance. Ganin and Lempitsky [35] utilized a simple gradient reversal layer to learn discriminative and domain-invariant features for image classification. Tzeng et al. [36] proposed a generalized framework for adversarial adaptation in which discriminative modeling, untied weight sharing, and a GAN loss are combined together.

Inspired by the works in CV, adversarial training has also been explored in some typical NLP tasks. Wang et al. [37] used adversarial learning for microblog sentiment classification. Then, Li et al. [38] proposed to use adversarial training for open-domain dialogue generation.

3. Methodology

In this section, we describe the proposed method in detail. The main idea is that we use adversarial learning to learn domain-invariant features from both source domain and target domain. It is noted that all data of target domain are unlabeled, thus our model belongs to unsupervised adaption methods which does not use any labeled data of target domain.

3.1 Model Overview

The overview of our model is shown in Fig. 1. As shown in Fig. 1, our model consists of four main modules: (1) the word layer that transforms the word sequence into vector representations via a lookup table; (2) the Bi-LSTM layer that transforms the vector representations into a latent vector with a fixed length; (3) the role classifier that predicts the label of the input argument; (4) the domain discriminator that predicts whether the data comes from the source domain or the target. We assume that we have a set of labeled training samples $\{s_1, \ldots, s_N\}$ from the source domain and a set of unlabeled data $\{s_{N+1}, \ldots, s_M\}$ from the target domain. In the next, we will illustrate the four modules in detail.

3.2 Word Layer

The word layer is mapping discrete language symbols into distributed real vectors. Usually, an embedding matrix is built to store all embedding vectors. In the task of semantic role labeling, argument classification is related to the predicate directly, thus the predicate-related features are also introduced. Following Li et al. [27], in this paper, for a word e_i in a sentence s, we construct the concatenation of the following features: a randomly initialized word embedding x_k^r, a pretrained word embedding x_k^p, a randomly initialized lemma embedding x_k^l, a randomly initialized POS tag embedding x_k^{pos}, and a predicate-specific feature x_k^f, which is a binary flag indicating whether the current word is the given predicate. The whole distributed real vectors for the word x_k in a sentence s is $[x_k^r, x_k^p, x_k^l, x_k^{pos}, x_k^f]$.

3.3 Bi-LSTM Layer

We use a Bi-LSTM layer to transform the input text into a vector with a fixed length. Many neural models have been investigated to model texts such as recurrent neural networks [24], convolutional neural networks [23]. Here we adopt recurrent neural network with long short-term memory (LSTM) due to their superior performance in addressing long-term dependencies [23], [24].

Here, we give a brief formulation about LSTM. Let us use $X = (x_1, x_2, \ldots, x_N)$ to denote an input sequence where $x_k \in R$, $1 \le k \le N$. At each position k, there is a set of internal vectors, including an input gate i_k, a forget gate f_k, an output gate o_k and a memory cell c_k. All these vectors are used together to generate a d-dimensional hidden state $\overrightarrow{h_k}$ as follows:

$$
\begin{aligned}
i_k &= \sigma(W^i x_k + V^i h_{k-1} + b^i) \\
f_k &= \sigma(W^f x_k + V^f h_{k-1} + b^f) \\
o_k &= \sigma(W^o x_k + V^o h_{k-1} + b^o) \\
c_k &= f_k \odot c_{k-1} i_k + tanh \odot (W^c x_k + V^c h_{k-1} + b^c) \\
\overrightarrow{h_k} &= o_k \odot tanh(c_k)
\end{aligned}
$$

where σ is the sigmoid function, $\odot$ is the element-wise multiplication of two vectors, and all $W^* \in R^{d \times l}$, $V^* \in R^{d \times d}$, $b^* \in R^d$ are weight matrices to be learned.

The above LSTM processes the input sequence in the forward direction, and we can get a d-dimensional hidden state $\overrightarrow{h_k}$. Similarly, we can also process the input sequence in the backward direction and we can get a d-dimensional hidden state $\overleftarrow{h_k}$. By concatenating the two states, we get a contextual representation $h_k = [\overrightarrow{h_k}, \overleftarrow{h_k}]$, which will be taken by the next layer of our framework.

Formally, the BiLSTM layer can be formulated as the following equation,

$$h_k = \boldsymbol{lstm}(x_k; \theta_l)$$

Where θ_l is the parameters of the Bi-LSTM layer.

3.4 Role Classifier

We use a softmax layer to make the final role prediction. The role classifier takes the output vector h_k of Bi-LSTM layer as the input and its output is denoted as $\hat{y}_k$. The role classifier f can be formulated as follows,

$$\hat{y}_k = f(h_k; \theta_y)$$

Where θ_y is the parameters of the role classifier f. The parameters of the role classifier f can be trained to minimize the cross-entropy of the predicted role $\hat{y}_k^j$ and gold role y_k^j. The objective function of f is

$$\text{Loss}_{rc}(h_k; \theta_y) = -\sum_{k=1}^{N}\sum_{j=1}^{l_k} y_k^j \log(\hat{y}_k^j)$$

where l_k denotes the length of the sentence and y_k^j is the ground-true label at the position k of the sentence with the length l_k.

3.5 Adversarial Training

If we only use the cross-entropy loss of the role classifier to train the model, our method can be seen as a standard supervised model which performs poorly on the out-of-domain evaluation due to the divergence between source domain data distribution and target domain data distribution. Therefore, we introduce adversarial learning to learn a domain-invariant features space.

Recent years have witnessed great successes of adversarial networks. Its idea is to learn a generative distribution $p_{G(x)}$ that matches the real data distribution $p_{data(x)}$ Specifically, GAN learns a generative network G and discriminative model D, in which G generates samples from the generator distribution $p_{G(x)}$ and D learns to determine whether a sample is from $p_{G(x)}$ or $p_{data(x)}$. Adversarial networks can be optimized by playing a max-min game in which discriminative model D is trained to classify the real samples correctly and fails to classify the samples generated by the generator distribution $p_{G(x)}$.

The main reason for the domain adaption problem is the giant difference between source domain data distribution and target domain data distribution. The problem can be tackled if we can construct a new vector space where the divergence between the two distributions are very small. Some works [28], [29] suggest that the adversarial loss can measure the H-divergence between two distributions. Thus, this paper incorporates adversarial learning to narrow down the divergence between the source distributions and the target.

We first introduce the domain discriminator in adversarial learning. The domain discriminator takes the output h_k of Bi-LSTM layer as the input, and predicts whether the input sequence is from source domain or target domain. Here, we also use a softmax layer to build the domain discriminator f_d. The output of the domain discriminator is denoted as $\hat{d}_k$. The domain discriminator f_d can be formulated as the following equation,

$$\hat{d}_k = f_d(h_k; \theta_d)$$

where θ_d is the parameters of the domain discriminator f_d. The loss $loss_{adv}$ of the domain discriminator is defined as the cross-entropy of the predicted domain label $\hat{d}_k$ and the ground-true domain label d_k.

$$\text{Loss}_{adv}(h_k; \theta_d) = -\sum_{k=1}^{M}\sum_{j=1}^{l_k} d_k^j \log(\hat{d}_k^j)$$

where d_k^j is the ground-true domain label at the position k of the sentence.

Formally, we consider the joint loss function in below,

$$\text{Loss}(\theta_f, \theta_y, \theta_d) = \text{Loss}_{rc}(\theta_f, \theta_y) - \lambda \text{Loss}_{adv}(\theta_f, \theta_d) \quad (1)$$

Following the setup of the adversarial training [28], we design a min-max game to optimize the whole network. The train process consists of the following two parts.

$$\begin{aligned}(\hat{\theta}_f, \hat{\theta}_y) &= \arg\min \text{Loss}(\theta_f, \theta_y, \hat{\theta}_d) \\ (\hat{\theta}_d) &= \arg\max \text{Loss}(\hat{\theta}_f, \hat{\theta}_y, \theta_d)\end{aligned} \quad (2)$$

The saddle point $(\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d)$ is obtained by training the two parts alternately and each part is optimized according to the different parameters. In the *max* step, the parameters $\hat{\theta}_d$ of the domain discriminator seek to minimize the domain classification errors (since it enters with the minus sign). In the *min* step, the parameters θ_f and θ_y are updated jointly. The parameters $\hat{\theta}_y$ of the role classifier seek to minimize the role classification loss while the optimization of the parameters $\hat{\theta}_f$ has two goals. One goal is to minimize the role classification loss, which make the learnt features are discriminative for role classification and the other is to maximize the domain classification loss which enhances the domain-invariant features to emerge up. The parameter λ is the hyper-parameter which makes a trade-off between the *min* and *max* game.

Algorithm 1 illustrates the pseudo-code of adversarial training. In the beginning, we initialize the model parameters $\theta_f, \theta_y, \theta_d$ and the batch size is set to S. During the training, for each batch we first make a forward propagation and compute the loss to update θ_f, θ_y using SGD; then, compute the loss again to update θ_d using SGD.

Algorithm 1: Adversarial Training

Input: labeled training samples $\{x_1, \ldots, x_N\}$ from source domain; unlabeled training samples $\{x_{N+1}, \ldots, x_M\}$ from the target domain
Output: network parameters $(\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d)$
Initialize the parameters $\theta_f, \theta_y, \theta_d$ and batch size is set to S
While stop condition is false **do**
    for s from 1 to S **do**
        compute $loss(\theta_f, \theta_y, \theta_d)$, for $x \in batch_s$,
        update θ_f, θ_y
        compute $loss(\theta_f, \theta_y, \theta_d)$, for $x \in batch_s$,
        update θ_d
    End for
End While

4. Experiments

4.1 Experiment Setup

To evaluate the proposed method, we conduct experiments on a benchmark: the English portion of CoNLL 2009 shared tasks. In the dataset, the training data comes from WSJ corpus while the out-of-domain test data comes from Brown corpus. Our method belongs to the unsupervised adaption method which needs a large set of unlabeled data. Here, we use unlabeled data consisting of the following sections of the Brown corpus: K, L, M, N, P. The proposed method is expected to learn domain-invariant features from both labeled and unlabeled data. The labeled training data from WSJ contains 39279 sentences while the unlabeled data from Brown contains 16407 sentences.

In the experiments, we use the pretrained GloVe embedding vectors[†] to initialize the word vectors and the dimension of word embedding as 100. All other vectors are randomly initialized, the dimension of lemma embeddings is 100, and the dimension of POS tag embedding is 64. The BiLSTM layers contains 256-dimensional hidden units. The dropout rate is set to 0.1, the batch size is 32, the learning rate is 0.001 and the max epoch is 30. The hyper-parameter λ is set to 1.

4.2 Metrics and Comparison Systems

We use the official tool[††] to score the outputs of different systems. The metrics about SRL in the tool are Precision (P), Recall (R) and labeled F_1 score (F_1). In this paper, we compare the proposed method with the below systems,

- **SourceLabeler**. The system is a standard supervised model which only takes classifications errors as the optimization objective.
- **Zhao09**. The system [17] reaches the best results on the out-of-domain test of the CoNLL 2009 shared task.
- **Yang15**. The system [13] utilizes deep belief network to learn a latent feature representation (LFR) for different domains in SRL.
- **Liu15**. We implement a multi-task learning framework [39] in which the hidden layers are shared by two domains while keeping two domain-specific output layers. Training the framework need labeled data of the target domain. Here, we utilize *SourceLabeler* to classify unlabeled data of Brown.
- **Kim16**. The system [12] proposes an easy feature augmentation method for domain adaption. We implement their method based on *SourceLabeler* system.
- **Roth16**. The system [27] jointly learns embeddings for dependency paths and feature combinations in a neural sequence-to-sequence model.
- **Marcheggiani17**. The system [26] proposes a version of graph convolutional networks for semantic role labeling.

[†]https://nlp.stanford.edu/projects/glove/
[††]https://ufal.mff.cuni.cz/conll2009-st/eval09.pl

Table 1 Comparison results.

Corpus	Method	P(%)	R(%)	F_1(%)
WSJ	SourceLabeler	89.1	86.8	88.0
	Zhao09	-	-	85.4
	Liu15	87.6	86.5	87.0
	Yang15	-	-	85.8
	Kim16	87.9	86.4	87.1
	Roth16	88.1	85.3	86.7
	Marcheggiani17	89.1	86.8	88.0
	Ours	88.7	86.7	87.7
Brown	SourceLabeler	78.3	75.4	76.8
	Zhao09	-	-	73.3
	Liu15	79.4	76.3	77.8
	Yang15	-	-	78.7
	Kim16	79.1	76.0	77.5
	Roth16	76.9	73.8	75.3
	Marcheggiani17	78.5	75.9	77.2
	Ours	80.2	77.1	78.6

4.3 Results and Discussions

Table 1 shows the results of all comparison systems on WSJ and Brown test set. From the table, we have the following observations.

First, the F_1 scores of all systems on Brown test set drop severely about 10 points compared with on WSJ test set, which experimentally confirms the domain adaption of semantic role classification. Our method achieves 78.6 F_1 score on Brown test set, 1.8 points improvement over *SourceLabeler* system, which suggests that the features learned by adversarial learning are beneficial to classifying roles of Brown corpus. It is also noted that our method performs poorer slightly on WSJ test set compared with *SourceLabeler*. We think the reason is that our method fails to learn some exclusive and discriminative features of source domain.

Second, we compare our method with *Kim16*. *Kim16* employs an easy feature augmentation method for domain adaption and achieves 77.5 F_1 score on Brown test set, 0.7 points higher than *SourceLabeler* which shows that the feature augmentation manner helps relieve domain adaption but the improvement of their method is limited.

Third, we compare our method with *Yang15*. *Yang15* utilizes deep belief network to learn a latent feature representation (LFR) for different domains and they report good results. But, one limit of their method is that they take one-hot features as the input of the network. There are more than 1,000,000 original features in the system, and thus their network is very huge. Training such a huge network is very costly and inefficient while our method can be trained efficiently due to its compact architecture. Therefore, although their method achieves comparable results with ours, our method is be superior to theirs in efficiency. Besides, the features and the classifier in their model are learned independently while our method can perform deep feature learning and role classification in the same framework together.

Fourth, we compare our method with *Liu15*. *Liu15* implements a multi-task learning framework in which the hid-

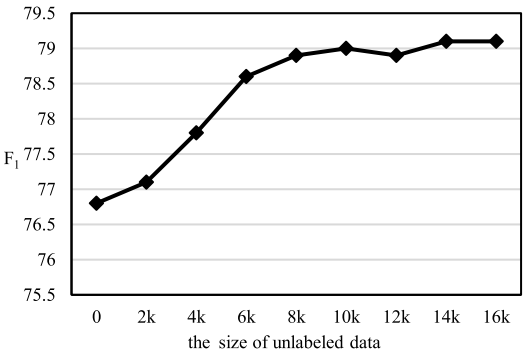

Fig. 2 The performance curve of our method as more unlabeled data being added.

den layers are shared by the two domains and the hidden layers are expected to learn generalized features for SRL. However, the performance of their method heavily depends on the quality of automatically labeled data of target domain while our method only needs unlabeled data of target domain.

Last, we compare our method with *Zhao09*, *Roth16* and *Marcheggiani17*. These systems explore syntax information for semantic role classification. Many previous works [14]–[17] show that syntax information is important for role classification but syntax information between different domains have giant difference which causes poor performance on Brown test set. Our method achieves large improvement over these methods.

4.4 Effects of Unlabeled Data from Brown Corpus

The proposed method in this paper is expected to learn domain-invariant features from both source domain and target domain, thus the performance is influenced by the unlabeled data of Brown corpus. Here, we investigate the effects of unlabeled data on the performance. Figure 2 shows the curve of the performance on Brown test set as more unlabeled data being added. From the figure, we can see that the performance of our method can improve as more unlabeled data being added. This suggests that unlabeled data from Brown corpus is crucial to our method and our method can learn discriminative and domain-invariant features. After 10k sentences are added, the curve reaches a peak and although even more data being added, there is not significant improvement.

4.5 Effects of the Length of the Sentences

Previous work shows that the performance of SRL is influenced by the length of the input sentence. Thus, we furtherly investigate the effects of the sentence length. Here, we compare our method with *SourceLabeler* system. We divide all sentences of Brown test set into six groups [1-5], [6-10], [11-20], [21-30], [31-50] and [51-]. The statistics about different groups are shown in Table 2. Most sentences fall in the groups [6-10], [11-20] and [21-30].

Figure 3 shows the performance comparisons of the two methods. From the figure, we can see a shared trend of the two methods is that the performance drops as the length of the sentences and the group [1-5] reaches the best performance while the group [51-] reaches the poorest performance. Our method achieves better results than *SourceLabeler* over all groups. An interesting point is that the performance gap of the two methods on the group [1-5] is smaller than other groups, which indicates that our method can perform better on long sentences than *SourceLabeler*.

Table 2 The statistics about different groups of Brown test set.

Groups	[1-5]	[6-10]	[11-20]	[21-30]	[31-50]	[51-]
Sentence number	42	106	145	83	46	6
Argument number	59	427	1293	1211	921	207

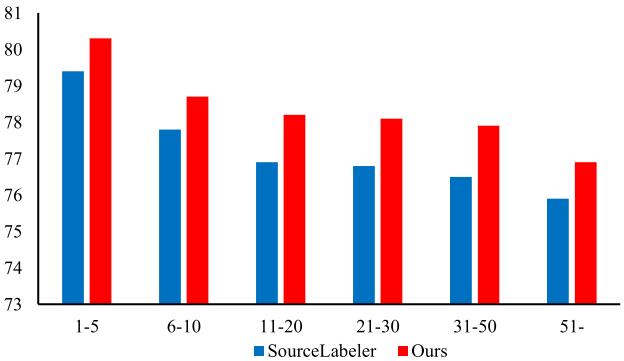

Fig. 3 The comparisons on different groups of Brown test set.

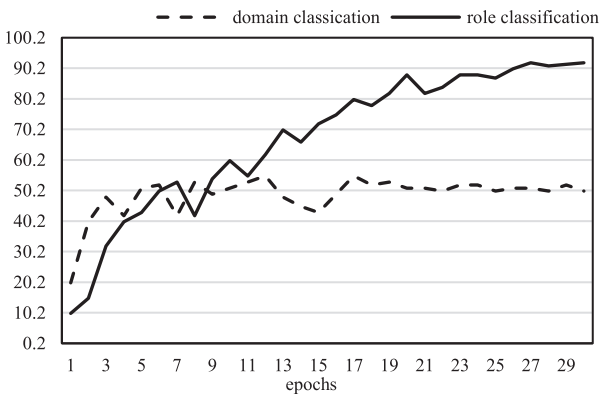

Fig. 4 The performance curve of the role classifier and the domain discriminator as training progresses.

4.6 Discussions about Adversarial Learning

In this paper, we use adversarial learning to drive the domain-invariant features to emerge up. Here, we further investigate how adversarial learning works in the training process. Figure 4 shows the performance curve of the role classifier and the domain discriminator on the training set as training progresses. The vertical axis means F_1 score for the domain classifier and accuracy for the domain discriminator. As can be seen from the figure, in the first 5 epochs, the accuracy of domain classification improves slowly. Then,

it fluctuates around random chance level (50%) with a big wave. After 20 epochs, the accuracy of domain classification is stable, which indicates that the learnt features can confuse the domain discriminator and adversarial learning successfully make domain-invariant features to emerge up. Different from the performance curve of the domain discriminator, the F_1 score of the domain classifier improves fluctuantly in the whole training process and converges after 30 epochs. In summary, the two curves prove that the learnt features by adversarial learning are domain-invariant and discriminative for SRL.

5. Conclusions

Current SRL systems face severe domain adaption problem, which limits the system's application on other domains. The reason is the giant feature difference between source and target domain. To relieve the problem, this paper proposes an adversarial domain adaption network for SRL. The core of our idea is to learn domain-invariant features via adversarial learning. We conduct experiments on CoNLL2009 share tasks and experimental results evaluate the effectiveness of our method. The domain-invariant features learned by adversarial learning can narrow down the discrepancy between different domains and relieve the domain adaption problem of SRL greatly.

Acknowledgments

This paper was financially supported by National Natural Science Foundation of China (No. 61702209, 61573163, 61662031) and self-determined research funds of CCNU from the colleges' basic research and operation of MOE (No. 20205170149).

References

[1] S. Narayanan and S. Harabagiu, "Question Answering Based on Semantic Structures," Proc. 20th International Conference on Computational Linguistics, Switzerland, Article no.693, Aug. 2004.
[2] J. Christensen, Mausam, S. Soderland, and O. Etzioni, "Semantic role labeling for open information extraction," Proc. NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading, Los Angeles, California, pp.52–60, June 2010.
[3] M. Surdeanu, S. Harabagiu, J. Williams, and P. Aarseth, "Using Predicate-argument Structures for Information Extraction," Proc. 41th Annual Meeting on Association for Computational Linguistics, Sapporo, Japan, pp.8–15, July 2003.
[4] D. Liu and D. Gildea, "Semantic role features for machine translation," Proc. 23th International Conference on Computational Linguistics, Beijing, China, pp.716–724, Aug. 2010.
[5] D. Wu and P. Fung, "Can semantic role labeling improve SMT?," Proc. 13th Annual Conference of European Association for Machine Translation, Barcelona, pp.218–225, May 2009.
[6] D. Xiong, M. Zhang, and H. Li, "Modeling the translation of predicate-argument structure for SMT," Proc. 50th Annual Meeting of the Association for Computational Linguistics, Jeju, Republic of Korea, pp.902–911, July 2012.
[7] F. Zhai, J. Zhang, Y. Zhou, and C. Zong, "Machine translation by modeling predicate argument structure transformation," Proc. 23th International Conference on Computational Linguistics, Mumbai, pp.3019–3036, Dec. 2012.
[8] F. Zhai, J. Zhang, Y. Zhou, and C. Zong, "Handling ambiguities of bilingual predicate argument structures for statistical machine translation," Proc. 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, pp.1127–1136, Aug. 2013.
[9] J. Blitzer, R. McDonald, and F. Pereira, "Domain Adaptation with Sturctural Correspondance Learning," Proc. 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, pp.120–128, July 2006.
[10] S.J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," Proc. 19th International World Wide Web Conference, Raleigh, North Carolina, USA, pp.751–760, April 2010.
[11] J. Yu and J. Jiang, "Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification," Proc. 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, pp.236–246, Nov. 2016.
[12] Y. Kim, K. Stratos, and R. Sarikaya, "Frustratingly easy neural domain adaptation," Proc. 26th International Conference on Computational Linguistics, Osaka, Japan, pp.387–396, Dec. 2016.
[13] H. Yang, T. Zhuang, and C. Zong, "Domain Adaptation for Syntactic and Semantic Dependency Parsing Using Deep Belief Networks," Transactions of the Association for Computational Linguistics, vol.3, pp.271–282, May 2015.
[14] B. David, J. Blitzer, C. Koby, and F. Pereira, "Analysis of representations for domain adaptation," Proc. 19th International Conference on Neural Information Processing Systems, Canada, pp.137–144, Dec. 2006.
[15] D. Gildea and D. Jurafsky, "Automatic labeling for semantic roles," Comput. Linguist., vol.28, no.3, pp.245–288, 2002.
[16] S. Pradhan, K. Hacioglu, V. Krugler, W. Ward, J.H. Martin, and D. Jurafsky, "Support Vector Learning for Semantic Argument Classification," Mach. Learn., vol.60, no.1-3, pp.11–39, Sept. 2005.
[17] H. Zhao and C. Kit, "Parsing Syntactic and Semantic Dependencies with Two Single-Stage Maximum Entropy Models," Proc. Twelfth Conference on Computational Natural Language Learning, Manchester, England, pp.203–207, Aug. 2008.
[18] H. Zhao, W. Chen, J. Kazama, K. Uchimoto, and K. Torisawa, "Multilingual Dependency Learning: Exploiting Rich Features for Tagging Syntactic and Semantic Dependencies," Proc. Thirteenth Conference on Computational Natural Language Learning, Boulder, Colorado, pp.61–66, June 2009.
[19] J. Li, G. Zhou, H. Zhao, Q. Zhu, and P. Qian, "Improving nominal SRL in Chinese language with verbal SRL information and automatic predicate recognition," Proc. 2009 Conference on Empirical methods in Natural Language Processing, Singapore, pp.1280–1288, Aug. 2009.
[20] N. Xue, "Labeling Chinese Predicates with Semantic Roles," Comput. Linguist, vol.34, no.2, pp.225–255, June 2008.
[21] V. Punyakanok, D. Roth, W.-T. Yih, and D. Zimak, "Semantic Role Labeling via Integer Linear Programming Inference," Proc. 20th International Conference on Computational Linguistics, Switzerland, Article no.1346, Aug. 2004.
[22] K. Toutanova, A. Haghighi, and C. Manning, "Joint Learning Improves Semantic Role Labeling," Proc. 43th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Ann Arbor, Michigan, pp.589–596, June 2005.
[23] R. Collobert and J. Weston, "Fast semantic extraction using a novel neural network architecture," Proc. 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, pp.560–567, June 2007.
[24] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," J. Mach. Learn. Res., vol.12, pp.2493–2537, 2011.
[25] J. Zhou and W. Xu, "End-to-end learning of semantic role labeling using recurrent neural networks," Proc. 53rd Annual Meeting

of the Association for Computational Linguistics, Beijing, China, pp.1127–1137, July 2015.
[26] D. Marcheggiani and I. Titov, "Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling," Proc. 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, pp.1506–1515, Sept. 2017.
[27] M. Roth and M. Lapata, "Neural Semantic Role Labeling with Dependency Path Embeddings" Proc. 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, pp.1192–1202, Aug. 2016.
[28] Z. Li, S. He, J. Cai, Z. Zhang, H. Zhao, G. Liu, L. Li, and L. Si, "A Unified Syntax-aware Framework for Semantic Role Labeling," Proc. 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp.2401–2411, Oct. 2018.
[29] F. Huang and A. Yates, "Open-domain semantic role labeling by modeling word spans," Proc. 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp.968–978, July 2010.
[30] Q.T.N. Do, S. Bethard, and M.-F. Moens, "Domain Adaptation in Semantic Role Labeling Using a Neural Language Model and Linguistic Resources," The IEEE/ACM Trans. Audio, Speech, Language Process., vol.23, no.11, pp.1812–1823, Nov. 2015.
[31] S. Hartmann, I. Kuznetsov, T. Martin, and I. Gurevych, "Out-of-domain FrameNet Semantic Role Labeling," Proc. 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, pp.471–482, April 2017.
[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Proc. Advances in Neural Information Processing Systems 27, Montreal Canada, pp.2672–2680, Dec. 2014.
[33] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," Proc. International Conference on Learning Representations, San Juan, Puerto Rico, pp.1–16, May 2016.
[34] J.-Y. Zhu, T. Park, P. Isola, and A.A. Efros, "Unpaired Image-to-Image ranslation using Cycle-Consistent Adversarial Networks," Proc. International Conference on Computer Vision, Venice, Italy, pp.2223–2232, Oct. 2017.
[35] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," Proc. 32rd International Conference on Machine Learning, Lille, France, pp.1180–1189, July 2015.
[36] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial Discriminative Domain Adaptation," Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, pp.2962–2971, July 2017.
[37] W. Wang, S. Feng, W. Gao, D. Wang, and Y. Zhang, "Personalized Microblog Sentiment Classification via Adversarial Cross-lingual Multi-task Learning," Proc. 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp.338–348, Oct. 2018.
[38] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial Learning for Neural Dialogue Generation," Proc. the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, pp.2157–2169, Sept. 2017.
[39] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-Y. Wang, "Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval," Proc. 2015 Annual Conference of the North American Chapter of the ACL, Denver, Colorado, pp.912–921, June 2015.

Haitong Yang received the Ph.D. degrees in Computer Science from University of Chinese Academy of Sciences in 2016. Since July 2016, he work in Central China Normal University. His research includes natural language process, deep learning, artificial intelligence.

Guangyou Zhou received his Ph.D. degree from National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (IACAS) in 2013. Currently, he worked as an full Professor at the School of Computer, Central China Normal University. His research interests include natural language processing and information retrieval.

Tingting He received her Ph.D. degree from Central China Normal University in 2003. Currently, he worked as a full Professor at the School of Computer, Central China Normal University. Her research interests include natural language processing and information retrieval.

Maoxi Li received the Ph.D. degree in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (2007–2011). He joined the School of Computer Information Engineering, Jiangxi Normal University as a lecturer in July 2011. His research interests are mainly in natural language processing and machine translation.