Shared Representation Learning with Self-Attention for Cross-domain Chinese Hedge Cue Recognition

Huiwei Zhou, Shixian Ning, Zhe Liu, Zhuang Liu and Chengkun Lang
School of Computer Science and Technology
Dalian University of Technology
Dalian, China
zhouhuiwei@dlut.edu.cn; {ningshixian, njjnlz, zhuangliu1992, kunkun}@mail.dlut.edu.cn

Abstract— **Hedge is usually used to express uncertainty and possibility. Research on the Chinese hedge cue detection is of great significance to the Chinese factual information extraction. So far, existing approaches focus on learning a model for Chinese hedge detection based on domain-specific corpus. However, the difference between hedge cue distributions in various domains makes domain-specific models difficult to extend to other domains. To make full use of out-of-domain data and reduce the cost of manual data annotation, we propose a novel cross-domain hedge cue detection approach based on shared representations (domain-general representations). Our method uses stacked CNNs model with self-attention to learn the shared semantic representations across different domains through parameter sharing mechanism. Meanwhile, we introduce adversarial training to separate the private features of each domain from the shared representations by iteratively training the generator and the discriminator alternately. Experiments on Chinese Biomedical domain and Wikipedia domain show that our method gets a significant improvement on cross-domain Chinese hedge cue detection, compared with instance-based transfer learning and feature-based transfer learning methods.**

Keywords- Chinese hedge cue detection; cross-domain; shared representations; self-attention; adversarial training

I. INTRODUCTION

Hedge is first proposed by G. Lakoff, which refers to the words that make things fuzzier and is used to express an opinion of uncertainty [1]. The information controlled by the hedges is called hedge information. In 2010, the Conference on Computational Natural Language Learning (CoNLL) [2] provides a competitive shared task of English hedge cue detection for the Computational Linguistics community, which greatly promoted the research of English hedge cue detection.

Hedge cue is also widely used in Chinese. It is usually used in various domains such as biomedicine and Wikipedia, and its role in different fields is different. Take the following two examples to illustrate. In the example S1 of Wikipedia domain, "根据 (according to)" is used to relieve the responsibility for the proposition that "Barbara Blackburn 是目前世界上最快的打字员 (Barbara Blackburn is the fastest typist in the world)". Therefore, "根据 (according to)" is used as a hedge in the example S1 of Wikipedia domain. In the example S2 of biomedical domain, "根据 (according to)" is used as the basis for the proposition that "提出一种结合灰色建模理论和混沌理论关联维数的人脑状态识别新方法 (a novel method for human brain state recognition based on the combined of grey modeling theory and chaotic theory correlation dimension is proposed)". This proposition is very clear and there is no ambiguity. Therefore, "根据 (according to)" is not used as a hedge in the example S2 of biomedical domain.

S1: <ccue>***根据***</ccue>《吉尼斯世界记录大全》, Barbara Blackburn 是目前世界上最快的打字员. (***According to*** Guinness Book of World Records, Barbara Blackburn is the fastest typist in the world.)

S2: ***根据***脑电信号自身的特点, 提出一种结合灰色建模理论和混沌理论关联维数的人脑状态识别新方法. (***According to*** the characteristics of EEG signals, a novel method for human brain state recognition based on the combined of grey modeling theory and chaotic theory correlation dimension is proposed.)

Traditional machine learning methods assume that training dataset and test dataset have the same data distribution. However, due to the domain-specificity of Chinese hedge, it is difficult to apply the existing models pre-trained on a certain domain to other domains directly. At the same time, annotating a large amount of training corpus for each domain is time-consuming and laborious. Szarvas et al. [3] points out that the hedges in resource-rich domains (source domains) can contribute to the recognition of the hedges in resource-poor domain (target domain), thereby efficiently reducing the cost of data annotation in the target domain.

For the case of very little training data in the target domain (only 200 training instances), this paper proposes a novel method based on shared representations for cross-domain Chinese hedge recognition. A large number of training data in the source domain is used to assist hedge cue detection in the target domain. First, we use stacked convolutional neural networks with self-attention mechanism to learn the shared representations from source and target data iteratively alternately. On one hand, stacking multiple convolutional layers could enhance the learning ability of advanced abstract features. On the other hand, self-attention mechanism [16] could directly capture the relationships between two arbitrary tokens in a sequence regardless of their distance. Then, adversarial training [4] is introduced to separate the private features of each domain from the shared features for purer shared semantic representations.

Our method can make effective use of information in both the source domain and target domain, and achieve better performance than traditional transfer learning methods in the cross-domain Chinese hedge cue recognition.

978-1-7281-1175-9/18/$31.00 ©2018 IEEE

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 30,2021 at 01:47:58 UTC from IEEE Xplore. Restrictions apply.

II. Related Work

Early researchers used a dictionary-based string matching method for hedge recognition. This method has achieved a high recall, but a very low precision. This may be because most of the hedges in the dictionary do not express the vague meaning in the context.

With the development of the hedge corpus, researchers begin to detect hedges based on machine learning methods. They transformed the problem of hedge detection into sequence labeling problem or classification problem. The former detects the hedges by labeling the position of the hedges in the sentence, such as beginning (B), middle (I) and end (E). The latter uses the hedge dictionary to extract candidate hedges and then classifies them. Ji et al. [5] proposed a variant online learning algorithm, named Passive Aggressive approach for Chinese hedge sentences classification and obtains a 70.53% *F*1-score in the Chinese news corpus. Chen et al. [6] recognized Chinese hedges based on sequence labeling methods and obtained a 43.20% *F*1-score on the "Journal of Computer" corpus. Zou et al. [7] constructed a feature-based sequence labeling model and obtained *F*1-scores of 73.27%, 70.29% and 68.57% in the three domains of scientific literature, stock market and product review, respectively.

The training data and test data of the above methods are all come from the same domain, that is, the training data has the same distribution as the test data. However, the use of hedges has the nature of domain-specific. Szarvas et al. [3] used transfer learning for cross-domain English hedges detection. They have experimentally proved that transfer learning can improve the performance of hedge detection system when the distribution of training data is inconsistent with test data. Existing methods of transfer learning can be divided into two kinds: feature-transfer learning [8], and instance-transfer learning [9]. Daumé III [8] proposed a simple but effective feature-transfer learning algorithm called FruDA, which introduced the common features between source domain and target domain to achieve the transformation of knowledge from the source domain to the target domain. Dai et al. [9] proposed an instance-transfer learning algorithm called TrAdaBoost, which adjusted the weights of the training instances in both the source domain and the target domain iteratively, so as to select training data in the source domain whose data distribution is similar to the target domain.

In recent years, with the development of deep learning, neural networks have been widely used for shared semantic representation learning among domains, and have achieved good results. Yang et al. [10] learned the shared character representations, which were obtained by using the similarity of spelling between two languages, and the private word representations of each language for the cross-language sequence labeling task, simultaneously. However, the shared representation learning method proposed by Yang et al. [10] cannot be used between languages without common characters. To overcome this problem, Kim et al. [11] adopted a shared Bidirectional Long Short-Term Memory (BLSTM) module to learn the shared representations among languages, and multiple language-specific private BLSTM modules to learn the private representations of each language. In addition, in order to make the shared representations language-independent, they introduced adversarial training in the shared BLSTM module to obtain a purer shared representation without private features. Chen et al. [12] proposed a shared-private model called **Sh-Pri** in this paper. They extracted shared features and private features from multiple Chinese word segmentation corpus, and similarly introduced adversarial training for shared representation learning. In the end, the segmentation performance of the **Sh-Pri** model on multiple corpus has been greatly improved.

Up to now, the shared representation learning has not been explored in cross-domain Chinese hedge recognition. Inspired by the **Sh-Pri** model proposed by Chen et al. [12], we propose a Shared-Adversarial model with Self-Attention (**Sh-Adv-SA**) for cross-domain hedge cue recognition in this paper. The proposed approach uses the stacked convolutional neural networks with self-attention to learn the shared semantic representations and introduces a domain discriminator to form adversary training with shared representation learning.

III. Methods

The architecture of our **Sh-Adv-SA** model is shown in the Fig. 1. It mainly consists of three parts: a shared representation generator G, a domain discriminator D and the hedge classifiers for the source domain and the target domain respectively.

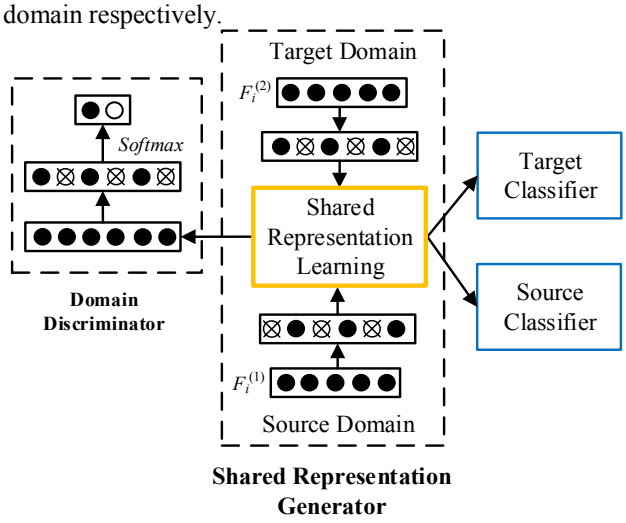

Figure 1. The architecture of the **Sh-Adv-SA** model.

Both of the the **Sh-pri** model [12] and our **Sh-Adv-SA** model use the idea of shared representation learning and adversarial training. The difference between them is that the **Sh-pri** model considers the commonalities and differences of data between different domains, that is, it learns the shared semantic representations while also learns the private semantic representation of each domain. However, we hypothesize that the training data in the target domain are extremely rare (only 200 training instances), this make the private semantic representation of the target domain cannot be fully learned. Therefore, the shared semantic representations may be affected by the private semantic representations, which reduce the performance of hedges recognition in the target domain. The **Sh-Adv-SA** model does not consider the private semantic representations of various domains, but directly introduces the domain discriminator to form an adversarial

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 30,2021 at 01:47:58 UTC from IEEE Xplore. Restrictions apply.

training with the shared representation generator for the purpose of obtaining a purer shared representation without private domain features.

Next, we will use three parts to introduce the Shared-Adversarial model with Self-Attention (**Sh-Adv-SA**) in detail. The first is the data preprocessing, followed by the shared representation learning, and finally the adversarial training.

A. Data Preprocessing

Given a data set $\{\mathbf{x}_i^{(m)}, \mathbf{y}_i^{(m)}\}_{i=1}^{N_m}$, where $\mathbf{x}_i^{(m)}$ is the input for the m^{th} domain, $\mathbf{y}_i^{(m)}$ is the corresponding hedge labels of $\mathbf{x}_i^{(m)}$, N_m is the number of training data in the m^{th} domain. We adopt the method of Zhou et al. [13] to preprocess the data. First of all, we use the Forward Maximum Matching algorithm to obtain candidate hedges based on the existing hedge dictionary. Then some relevant features of candidate hedges are extracted and used as input to the model. The relevant features include contextual features $\mathbf{x}_i^{(m)} = (w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}) \in \mathbf{R}^{d_1 \times n}$, POS features $\mathbf{pos}_i^{(m)} = (p_{i-2}, p_{i-1}, p_i, p_{i+1}, p_{i+2}) \in \mathbf{R}^{d_2 \times n}$ and co-occurrence features $cooccurrence_i^{(m)} \in \mathbf{R}^{d_3}$, where the context window size is 2, n is the length of input instance, d_1, d_2 and d_3 are the dimensions of the word vector w_i, the POS vector p_i, and the co-occurrence vector, respectively. The co-occurrence features refer to whether there are two or more candidate hedges in the same sentence, such as "不是…就是… (either... or...)". If there are two or more candidate hedges in a sentence, then $cooccurrence_i^{(m)}=Y$, otherwise $cooccurrence_i^{(m)}=N$. We get the input instances $\{F_i^{(m)}, y_i^{(m)}\}_{i=1}^{N_m}$ by concatenating the above features by location $\mathbf{F}_i^{(m)} = (\mathbf{x}_i^{(m)}; \mathbf{pos}_i^{(m)}; cooccurence_i^{(m)} \times n) \in \mathbf{R}^{d \times n}$, where d is the dimension of input vector.

B. Shared representation learning

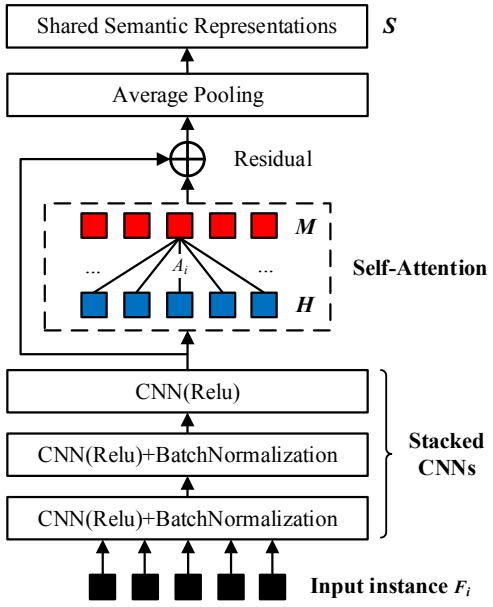

Figure 2. The architecture of the shared representation generator.

We use the shared representation generator to learn the shared representations for the hedge classification and the domain discrimination. Fig. 2 illustrates the architecture of the shared representation generator. The input instance first passes through three layers of stacked CNNs for feature extraction, with a BatchNormalization layer [17] between each layer, then follows the self-attention layer and finally joins the residual connection to obtain shared representations.

1) Stacked CNNs: Stacked CNNs expands the receptive field by stacking layers to enhance the learning ability of advanced abstract features. It consist of multiple convolutional layers with Relu activation and BatchNormalization layers. In convolutional layer, we slide a filter $\mathbf{W}_h \in \mathbf{R}^{d \times k}$ of size k over the output of the previous layer $\mathbf{E} \in \mathbf{R}^{d \times n}$. The convolutional operation with can be expressed as $h_i = f(\mathbf{W}_h \mathbf{E}_{i:i+k-1} + b_h)$, where b_h is a bias term and f is a Relu nonlinear activation. As the word window slides, a feature map can be represented as follows: $\mathbf{h} = (h_1, h_2, ..., h_n) \in \mathbf{R}^n$. In this paper, we use q filters to obtain multiple feature maps $\mathbf{H}_l = (\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_q) \in \mathbf{R}^{q \times n}$, where l indicates the current number of convolutional layers. Then BatchNormalization layer is used to normalize the activations of the previous layer at each batch, i.e. applies a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1.

2) self-attention: To model dependencies between two arbitrary tokens in a sequence, the output vector sequence $\mathbf{H}_{last} \in \mathbf{R}^{q \times n}$ obtained from the last layer of stacked CNNs are fed to the self-attention layer. Self-attention calculates response at a position as a weighted sum of the features at all positions. The attention weights can be calculated by a softmax function as follows:

$$\mathbf{A} = softmax(\mathbf{W}_a \mathbf{H}_{last}^T) \in \mathbf{R}^{n \times q} \tag{1}$$

where $\mathbf{W}_a \in \mathbf{R}^{n \times n}$ is the weighted matrix. The softmax is performed along the second dimension of its input. Then the resulting matrix of the self-attention layer is computed with:

$$\mathbf{M} = \mathbf{A}^T \otimes \mathbf{H}_{last} \in \mathbf{R}^{q \times n} \tag{2}$$

where $\otimes$ represents the element-wise product operation.

3) residual connection: In addition, we explore a residual connenction to add the $\mathbf{H}_{last}$ back to the output of the self-attention layer $\mathbf{M}$, followed by an average pooling operation to obtain the final shared semantic representations S:

$$S = AveragePooling(\mathbf{M} + \mathbf{H}_{last}) \in \mathbf{R}^q \tag{3}$$

After that, softmax is used to predict the hedge label of the current input instance $\hat{\mathbf{y}}_i^{(m)} = softmax(\mathbf{W}_1^T S + b_1)$.

The hedge classifier is trained by optimizing the cross-entropy loss as follows,

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 30,2021 at 01:47:58 UTC from IEEE Xplore. Restrictions apply.

$$J_{hedge} = -\sum_{i=1}^{N_m} \sum_{j=1}^{C} \left(\mathbf{y}_{i,j}^{(m)} \log \left(\hat{\mathbf{y}}_{i,j}^{(m)} \right) \right) \quad (4)$$

where C is the number of categories of hedge, $\hat{\mathbf{y}}_{i,j}$ is the predicted probability of the j^{th} category of the i^{th} input instance. $m = 1$ refers to source domain, and $m = 2$ refers to target domain.

C. Adversarial training

In order to reduce the private semantic representations mixed in the shared representations, a domain discriminator is introduced to form an adversary with shared representation learning. Inspired by Goodfellow et al. [4], our adversarial training consists of two parts, named shared representation generator *G* and domain discriminator *D* respectively. The purpose of the generator *G* is to learn the shared data distribution between the source domain and the target domain as much as possible; and the domain discriminator *D* is to discriminate which domain (source or target) the input instance belongs to.

Specifically, **Sh-Adv-SA** model use the shared representation generator $G\left(\mathbf{F}_i^{(m)}, \theta_s\right)$, which is mentioned in section III. B, to get the shared semantic representations S. θ_s is the shared parameters need to be trained. Then, it will be fed to the domain discriminator to map the shared representation into a probability distribution, estimating what kinds of domains the input instance comes from.

$$D\left(h^s, \theta_d\right) = softmax\left(\mathbf{W}_2^T S + b_2\right) \quad (5)$$

where $\mathbf{W}_2 \in \mathbf{R}^{q \times 2}$ is a learnable parameter and b_2 is a bias.

For the input instances of different domain, adversarial loss J_{adv} is introduced to train the generator *G* and the domain discriminator *D*. In order to get purer shared representations, the two are adversarial to each other and alternately optimized to separate the private features from the shared representations. The adversarial loss J_{adv} is trained in an alternating fashion as shown below,

$$J_{adv} = \min_{\theta_s} \left(\lambda \max_{\theta_d} \left(\sum_{m=1}^{2} \sum_{i=1}^{N_m} \mathbf{t}_i^{(m)} \log \left[D\left(G\left(\mathbf{F}_i^{(m)}, \theta_s \right), \theta_d \right) \right] \right) \right) \quad (6)$$

where $\mathbf{t}_i^{(m)}$ is the correct domain label (source domain or target domain) of the input instance $\mathbf{F}_i^{(m)}$, and λ is a weight which is used to control the proportion of the discriminator and the shared module in the adversarial loss. Here, the basic idea in the min-max optimization is that, the shared semantic representations learned by the generator need to mislead the domain discriminator. At the same time, the domain discriminator needs to predict the domain to which the data belongs as accurately as possible.

After the training phase, the shared representation generator and domain discriminator reach a point at which both cannot improve and the discriminator is unable to differentiate between the two domains.

IV. EXPERIMENTS

A. Data

We evaluate our method on the Chinese Hedge Corpus[1] (CHC) constructed by Zhou et al. [14], which contains Wikipedia, and results, abstracts, discussions of the biomedical literature, totally 24,414 sentences and approximately 1 million words. In the Wikipedia, 33.78% sentences contain hedges. In the biomedical literature, 27.8% sentences in result, 25.28% sentences in abstract, and 47.69% sentences in discussion contain hedges. The number of hedges in each part are 1958, 1622, 2759 and 4674 respectively. In order to test the performance of cross-domain Chinese hedge cue detection in both Wikipedia and biomedical domains, we have set up a total of six groups of experiments, as shown in Tab. I.

In addition, we use a total of 117M corpus to train word vectors with Word2Vec toolkit[2], including 6.19M biomedical literature abstracts downloaded from Wanfang Data[3], 106M Chinese Wikipedia corpus and 4.16M CHC corpus used in our experiments.

B. Implementation Details

To reduce chance, we performed a 5-fold cross validation on each group of experiments. That is, we divide the target data into five equal parts, taking 200 instances of each of them as training data and the remaining four as test data. We use *F*1-score to evaluate the performance of the model. The formula for calculating the *F*1-score is as $F1 = \frac{2PR}{P+R}$, where P indicates precision and R indicates recall.

The POS features and co-occurrence features are initialized randomly and optimize through training phase. The dimensions of word vectors, POS features, and co-occurrence features are 100, 50, and 10, respectively. 160 filters with window size $k = 5$ are used in CNN. RMSprop [18] optimizer is used for both generator and discriminator to update the parameters. It is worth noting that for the classifier in the target domain, we use a larger learning rate to compensate for the slow fitting problem caused by the small amount of data in the target domain. The weight of adversarial loss λ is set to 0.05. Mini-batch size is 64 and epoch size is 15.

In order to alleviate the over-fitting problem, we apply the dropout on the final input layer. We fix dropout rate at 0.5 throughout the experiments. In addition, batch normalization method is also used in our model to promote the convergence speed and prevent overfitting to some extent. Tanh activation is used as the last layer of the generator output, while ReLU activation is used in other layers to help the gradients flow easier through the architecture and speed up the training.

C. Baseline methods

In order to explore the effects of shared representations on cross-domain Chinese hedge detection, we compared the following four baseline methods: Support Vector Machine based on Linear Kernel Function (SVM), single-

[1] https://github.com/DUT-NLP/CHScope

[2] Available at https://code.google.com/p/word2vec/.

[3] http://www.wanfangdata.com.cn/

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 30,2021 at 01:47:58 UTC from IEEE Xplore. Restrictions apply.

layer BLSTM without shared representation learning module (BLSTM_NS), Feature-based transfer learning method FruDA [8] and Instance-based transfer learning method TrAdaBoost [9]. The three data formats used by the baseline methods are as follows:

1) Target Only (TO): use only 200 instances in the target domain to train the model.

2) Source Only (SO): use all instances in the source domain to train the model.

3) Target + Source (T+S): use the 200 instances in the target domain and all instances in the source domain to train the model simultaneously.

During testing, all models predict on the test data of the target domain. In addition, the FruDA method and TrAdaBoost method only use T+S data for training, and their parameter settings are all default. The average *F*1-score of the four baseline methods in the six groups of experiments are shown in Tab. II.

TABLE I. 6 GROUPS OF EXPERIMENTAL DATA SETTINGS

Group	Source	Target	Group	Source	Target
1	Wikipedia	Biomedical (discussions)	4	Biomedical (discussions)	Wikipedia
2	Wikipedia	Biomedical (abstracts)	5	Biomedical (abstracts)	Wikipedia
3	Wikipedia	Biomedical (results)	6	Biomedical (results)	Wikipedia

TABLE II. THE *F*1-SCORE OF THE BASELINE METHODS ON CROSS-DOMAIN CHINESE HEDGE DETECTION

Method	Format	Group						Average
		1	2	3	4	5	6	
SVM	TO	38.29	37.55	42.86	32.41	32.41	32.41	35.99
	SO	64.26	55.54	60.01	74.51	63.58	61.78	63.28
	T+S	67.01	64.22	63.93	75.04	66.51	63.82	66.76
BLSTM_NS	TO	31.02	43.94	40.04	53.90	57.89	54.55	46.89
	SO	63.00	71.30	65.10	75.84	79.10	74.57	71.46
	T+S	67.59	73.05	68.63	76.89	79.13	76.74	**73.67**
FruDA	T+S	67.26	65.04	68.59	72.28	64.33	69.00	67.75
TrAdaBoost	T+S	72.77	65.18	64.16	74.24	64.44	65.63	67.74

It can be seen from the experimental results in Tab. II that: (1) The result of *F*1-score performed on the TO data is the worst, the SO data is greatly improved, and the T+S data is the best. It explains that using the data from source domain to assist shared representation learning can achieve the similar data distribution between the two domains, providing feasibility for data transfer between biomedical and Wikipedia domains. (2) The average *F*1-score of FruDA method and TrAdaBoost method in T+S data are lower than that of BLSTM_NO method. This is because the former could only capture the shallow semantic information, while the latter could make effective use of the deep semantic dependent information, thus improves the performance of cross-domain hedges detection.

D. Shared representation methods

We compare the *F*1-score between the **Sh-Pri** model and our **Sh-Adv-SA** model in the six groups of cross-domain Chinese hedge detection experiments. For the **Sh-Adv-SA** model, we consider the following three variants: 1) **w/o Stacked CNNs & w/ BLSTM**: This variant is to explore the effect of different feature extraction models on the shared representation learning. We simply replace stacked CNNs with BLSTM model. 2) **w/o discriminator**: This variant is to explore the effect of the discriminator on the shared representation learning. We get it through removing the discriminator module from both model. 3) **w/o self-attention**: This variant is to investigate the effect of self-attention mechanism. We get it through removing the self-attention from the shared representation generator.

Tab. III compares the experimental results of the **Sh-Pri** model, the **Sh-Adv-SA** model and the three variants of **Sh-Adv-SA** on the CHC corpus. From Tab. III, we can get the following conclusions,

1) In general, the results of shared representation methods are better than that of baseline methods, indicating the effectiveness of the shared representations-based methods in cross-domain Chinese hedges detection.

2) For **w/o Stacked CNNs & w/ BLSTM**, we can see that using stacked CNNs as a feature extraction module is better than BLSTM. Since the length of input instance is not very long, stacked CNNs can get advanced abstract features by stacking multiple convolutional layers, which contains enough n-gram information.

3) For **w/o discriminator**, when we remove the discriminator module, their performance has dropped slightly both. In the case of scarce training data in the target domain, it is difficult to learn to obtain the private semantic representations of the target domain.

4) For **w/o self-attention**, we can see that the performance is slightly reduced when we remove the self-attention layer from the shared representation generator. Self-attention mechanism conducts direct connections between two arbitrary tokens to model global dependencies of input instance. At the same time, self-attention can learn the internal structure of input instance, thus obtaining better shared representations.

5) Finally, comparing the **Sh-Pri** model with the **Sh-Adv-SA** model, we can find that the introduction of

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 30,2021 at 01:47:58 UTC from IEEE Xplore. Restrictions apply.

private semantic representation learning in the source domain during training will affect the performance of the hedge detection in target domain. On the contrary, extracting shared features from different domains can better realize the recognition of cross-domain hedges. However, the improvement brought by the adversarial training is not obvious. The main reason may be that the shared representation module tries to maintain the shared features through the adversarial training, however, too little training data in the target domain has resulted in private features that cannot be completely stripped from the shared representation. Ultimately, this leads to the model being unable to learn a purer shared representation.

TABLE III. THE *F*1-SCORE OF THE SHARED REPRESENTATION METHODS ON CROSS-DOMAIN CHINESE HEDGE DETECTION

Method	Format	Group						Average
		1	2	3	4	5	6	
Sh-pri	T+S	69.63	73.75	70.81	77.16	79.79	77.42	74.76
Sh-Adv-SA	T+S	**73.86**	**76.83**	**74.10**	**78.30**	**79.94**	78.94	**77.00**
w/o Stacked CNNs & w/ BLSTM	T+S	72.8	76.30	73.50	77.71	79.51	78.83	76.44
w/o discriminator	T+S	73.03	76.61	73.53	77.83	79.73	**79.03**	76.63
w/o self-attention	T+S	72.88	76.28	73.85	78.26	79.58	78.86	76.62

V. CONCLUSION

In this paper, we propose a novel cross-domain hedge cue detection approach based on the shared representations. Under the condition that the training data in the target domain are rarely scarce with only 200 instances, we use adversarial training strategy to learn purer shared semantic representations between source domain and target domain. At the same time, we incorporate a self-attention mechanism into the shared representation learning to directly capture the dependencies between two arbitrary tokens in a sequence. In the experiments of biomedicine and Wikipedia domains, our method has achieved state-of-the-art performance in cross-domain Chinese hedge detection.

For the future work, we will explore how to combine deep learning with active learning to further improve the performance of cross-domain Chinese hedge cue recognition under a small amount of manually annotated data.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (No. 61772109, No. 61272375) and the Ministry of education of Humanities and Social Science project (No. 17YJA740076).

REFERENCES

[1] G. Lakoff, "Hedges: a study in meaning criteria and the logic of fuzzy concepts," Journal of Philosophical Logic, vol. 2, no. 4, pp. 458-508, 1973.

[2] R. Farkas, V. Vincze, G. Móra, J. Csirik, G. Szarvas, "The CoNLL-2010 shared task: learning to detect hedges and their scope in natural language text," in Proceedings of the Fourteenth Conference on Computational Natural Language Learning. ACL, 2010, pp. 1-12.

[3] G. Szarvas, V. Vincze, R. Farkas, G. Móra, I. Gurevych, "Cross-genre and cross-domain detection of semantic uncertainty," Computational Linguistics, vol. 38, no. 2, pp. 335-367, 2012.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, Y. Bengio. "Generative adversarial nets," in Advances in neural information processing systems. 2014, pp. 2672-2680.

[5] F. Ji, X. Qiu, X. Huang. "Exploring uncertainty sentences in Chinese," in Proceedings of the 16th China Conference on Information Retreval. CCL, 2010, pp. 594-601.

[6] Z. Chen, B. Zou, Q. Zhu, P. Li, "Chinese negation and speculation detection with conditional random fields," in Natural Language Processing and Chinese Computing. Springer Berlin Heidelberg, 2013, pp. 30-40.

[7] B. Zou, Q. Zhu, G. Zhou, "Negation and Speculation Identification in Chinese Language," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. ACL, 2015, vol. 1, pp. 656-665.

[8] H. Daumé III. "Frustratingly easy domain adaptation," in Proceedings of the 45rd Annual Meeting of the Association for Computational Linguistics. ACL, 2007, pp. 256-263.

[9] W. Dai, Q. Yang, G. Xue, "Boosting for transfer learning," in International Conference on Machine Learning. ACM, 2007, pp.193-200.

[10] Z. Yang, R. Salakhutdinov, W W. Cohen, "Transfer learning for sequence tagging with hierarchical recurrent networks," arXiv preprint arXiv:1703.06345, 2017.

[11] J. K. Kim, Y. B. Kim, R. Sarikaya, E. Fosler-Lussier, "Cross-Lingual Transfer Learning for POS Tagging without Cross-Lingual Resources," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. ACL, 2017, pp. 2822-2828.

[12] X. Chen, Z. Shi, X. Qiu, X. Huang, "Adversarial multi-criteria learning for chinese word segmentation," arXiv preprint arXiv:1704.07556.

[13] H. Zhou, X. Li, D. Huang, Z. Li, Y. Yang, "Exploiting multi-features to detect hedges and their scope in biomedical texts," in Proceedings of the Fourteenth Conference on Computational Natural Language Learning---Shared Task. ACL, 2010, pp. 106-113.

[14] H. Zhou, H. Yang, J. Zhang, S. Kang, D. Huang, "The research and construction of Chinese hedge corpus," Journal of Chinese Information Processing, vol. 29, no. 6, pp. 83-89, 2015.

[15] H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, "Self-Attention Generative Adversarial Networks," arXiv:1805.08318, 2018.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," arXiv:1706.03762, 2017.

[17] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[18] T. Tieleman, G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural networks for machine learning, vol. 4, no. 2, pp. 26-31, 2012.

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 30,2021 at 01:47:58 UTC from IEEE Xplore. Restrictions apply.