Adversarial Discriminative Domain Adaptation for Extracting Protein-Protein Interactions from Text

Anthony Rios[1], Ramakanth Kavuluru[1,2], Zhiyong Lu[3]

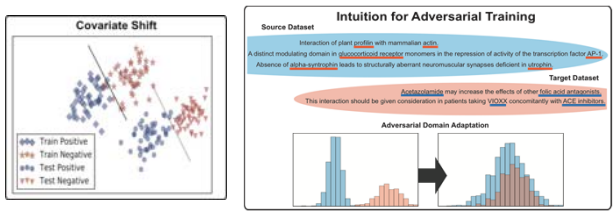
[1]Department of Computer Science; [2]Division of Biomedical Informatics, Department of Internal Medicine; University of Kentucky, Lexington, KY

[3]National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, Maryland

Introduction

Relation extraction is the process of extracting structured information from unstructured text. Recently, neural networks (NNs) have produced state-of-art results in extracting protein-protein interactions (PPIs) from text. While multiple corpora have been created to extract PPIs from text, most methods have shown poor cross-corpora generalization. In other words, models trained on one dataset perform poorly on other datasets for the same task. In the case of PPI, the F1 has been shown to vary by as much as 30% between different datasets. In this work, we utilize adversarial discriminative domain adaptation (ADDA) to improve the generalization between the source and target corpora. Specifically, we introduce a method of unsupervised domain adaptation, where we assume we have no labeled data in the target dataset.

Motivation



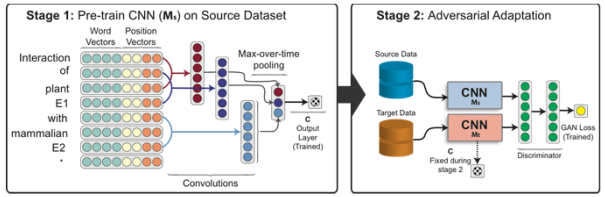
There are many external factors that can cause relation extraction datasets to become biased. One such issue is sample selection bias. What happens is the data distribution at test time will not match the training distribution. This phenomenon is also referred to as covariate shift. A hypothetical example distribution projected in two dimensions is illustrated above. The blue diamonds and circles represent positive train and test instances respectively. Likewise, the red stars and triangles represent negative train and test instances. We can see that the test distribution does not match the training distribution. Because of the biased sample, it is unreasonable to expect our models to generalize well on the test set. In order to overcome bias issues we make use of adversarial training. Specifically, we assume we have a small labeled dataset (source data) and a large unlabeled dataset (target dataset). Intuitively, our model forces the internal feature representation of instances in the target data to look like the source data instances. This idea is represented in the intuition figure above.

Datasets

	# Sentences	# Positive	# Negative
AIMed (**A**)	1955	1000	4834
BioInfer (**B**)	1100	2534	7132
DDI (**D**)	4579	4999	28509

In this work, we experiment with three datasets. Two protein-protein interaction datasets (AIMed and BioInfer) and a drug-drug interaction (DDI) dataset. Intuitively, both PPI and DDI interactions are have similar contexts in text. However, the slight differences in both their syntactic and semantic contexts result in poor cross-corpora generalization. This provides a good testbed to understand how our method works in when faced with a large amount of bias in the source dataset.

Method Overview



Method

Our method is trained in two stages. First, as shown in the Method Overview Figure, we train a standard CNN based relation classification model on the source data. It is important to note that the target data is not used at this stage. To train the model we use a traditional binary cross-entropy loss function.

$$\mathbb{E}_{(\mathbf{s},y)\sim\mathbf{S}}\big[-y\,log(C(\mathbf{s}))-(1-y)\,log(1-C(\mathbf{s}))\big]$$

Next, in the second stage we ignore the classification loss - meaning the final output layer from stage 1 is not updated - and train a new output layer (discriminator) that tries to predict whether each example is from the source or target dataset. Also, we make two copies of the CNN parameters and update the target CNN parameters to confuse the discriminator in stage 2. Similar to the classification loss, the adversarial loss is also a binary cross-entropy function that is minimized to predict whether a given instance is from the source or target dataset. The loss function is defined as

$$-\mathbb{E}_{\mathbf{s}\sim\mathbf{S}}\left[log(D(M_s(\mathbf{s})))\right]-\mathbb{E}_{\mathbf{t}\sim\mathbf{T}}\left[log(1-D(M_t(\mathbf{t})))\right]$$

where D is a 3 layer neural network with two 512 node hidden layers and a single sigmoid output node. It is important to note only the parameters of D are updated with this loss function even though the discriminator takes as input the max-pooled features from the CNN.

With discriminative adversarial training, we want to update the parameters of the CNN such that it produces features that are not discriminative for the discriminator. We accomplish this by flipping the target label compared to the discriminators loss.

$$-\mathbb{E}_{\mathbf{t}\sim\mathbf{T}}\left[log(D(M_t(\mathbf{t})))\right]$$

Unfortunately, because of the competition between the last two loss functions we can observe oscillation during learning rather than converging to an equilibrium. Even worse, the model could converge to a degenerate solution. To overcome this we combine the last loss function with a historical regularization term

$$-\mathbb{E}_{\mathbf{t}\sim\mathbf{T}}\left[log(D(M_t(\mathbf{t})))\right]+\|\theta_{M_t}-\frac{1}{V}\sum_{j}^{V}\theta^j_{M_t}\|^2_F$$

applied to the target CNN weights where $\theta^J_{M_t}$ represents the parameters of the target CNN after the j-th update. The training algorithm is formally defined in the next section.

Training Algorithm

Algorithm 1 Mini-batch stochastic gradient descent algorithm to train our adversarial domain-adaptation method

for max number of stage 1 training iterations **do**
  Sample minibatch of m source instances $\{\mathbf{s}^1,\mathbf{s}^2,\ldots,\mathbf{s}^m\}$ from $\mathbf{S}$
  Minimize the classification loss and update the parameters according to the gradient:

$$\nabla_{\theta_{M_s},\theta_C}\frac{1}{m}\sum_{i}^{m}y^i\,log(C(M_s(\mathbf{s}^i)))-(1-y^i)\,log(1-C(M_s(\mathbf{s}^i)))$$

end for
for max number of stage 2 training iterations **do**
  Sample minibatch of k source instances $\{\mathbf{s}^1,\mathbf{s}^2,\ldots,\mathbf{s}^k\}$ from $\mathbf{S}$
  Sample minibatch of k target instances $\{\mathbf{t}^1,\mathbf{t}^2,\ldots,\mathbf{t}^k\}$ from $\mathbf{T}$
  Update the discriminator parameters using the following gradient:

$$\nabla_{\theta_D}\frac{1}{k}\sum_{i}^{k}log(D(M_s(\mathbf{s}^i)))-log(1-D(M_t(\mathbf{t}^i)))$$

  Sample a new batch of k target instances $\{\mathbf{t}^1,\mathbf{t}^2,\ldots,\mathbf{t}^k\}$ from $\mathbf{T}$
  Update the base model parameters using the following gradient:

$$\nabla_{\theta_{M_t}}\frac{1}{k}\sum_{i}^{k}log(D(M_t(\mathbf{t}^i)))+\|\theta_{M_t}-\frac{1}{V}\sum_{j}^{V}\theta^j_{M_t}\|^2_F$$

end for

Results

	B ➔ A	A ➔ B	D ➔ A	D ➔ B
CNN	0.4522	0.3975	0.2793	0.2213
CNN RevGrad	0.4731	0.4255	0.3072	0.3611
Adv-CNN (Ours)	**0.4879**	**0.5413**	**0.4471**	**0.4853**

In this work, we are interested only in the relation detection problem. We compare our method with a CNN without domain adaptation and we compare against a recent variant of adversarial domain adaptation, RevGrad. We run 4 pairwise (source ➔ target) experiments and use the F1-score as our evaluation measure. We note that our method outperforms the other methods across all 4 experiments. Likewise, we see that when there is a large domain shift (D➔B) cross-corpora performance is very low, however after applying our method we make substantial improvements.

Conclusion

We have demonstrated the usefulness of adversarial domain adaptation for relation extraction. However, there is still room for improvement and we see two promising areas to expand this work: 1) In this work we are only dealing with a single source, however we may have access to multiple biased labeled data sources that we can use to improve our results. 2) Distant supervision (DS) has proven valuable when applied to relation extraction tasks. Unfortunately, DS datasets are biased because of faulty assumptions. If we can use use adversarial learning to remove the bias we believe we can better take advantage of distantly supervised datasets.

References

- Yifan Peng and Zhiyong Lu. Deep learning for extracting protein-protein interactions from biomedical literature. In BioNLP 2017 , pages 29–38, Vancouver, Canada,, August 2017. Association for Computational Linguistics.
- Eric Tzeng, Judy Homan, Trevor Darrell, and Kate Saenko. Adversarial discriminative domain adaptation. In Computer Vision and Pattern Recognition (CVPR) , 2017.