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

Anthony Rios¹, Ramakanth Kavuluru¹,², Zhiyong Lu¹
¹Department of Computer Sciences, ²Division of Biomedical Informatics, Department of Internal Medicine, University of Kentucky, Lexington, KY
²National 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-the-art results in extracting protein-protein interactions (PPIs) from text. While 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.

Method Overview

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.

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.

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.

\[ \mathcal{L}_{\text{CE}} = -\frac{1}{|S|} \sum_{i=1}^{|S|} (y_i \log(C(s_i)) + (1-y_i) \log(1-C(s_i))) \]

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.

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.

\[ \mathcal{L}_{\text{D}} = -\mathbb{E}_{s \sim D}(\log(D(M_t(s)))) - \mathbb{E}_{t \sim T}(-\log(1-D(M_t(t)))) \]

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.

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.

\[ \mathcal{L}_{\text{D}} = -\mathbb{E}_{s \sim D}(\log(D(M_t(s)))) - \mathbb{E}_{t \sim T}(-\log(1-D(M_t(t)))) \]

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.

\[ \mathcal{L}_{\text{D}} = -\mathbb{E}_{s \sim D}(\log(D(M_t(s)))) - \mathbb{E}_{t \sim T}(-\log(1-D(M_t(t)))) \]

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.

Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Sentences</th>
<th># Positive</th>
<th># Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIMed (A)</td>
<td>1955</td>
<td>1000</td>
<td>4834</td>
</tr>
<tr>
<td>BioInfer (B)</td>
<td>1100</td>
<td>2534</td>
<td>7132</td>
</tr>
<tr>
<td>DDI (D)</td>
<td>4579</td>
<td>4999</td>
<td>28509</td>
</tr>
</tbody>
</table>

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 vs. 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 vs. 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 datasets 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 adversarial learning to remove the bias we believe we can better take advantage of distantly supervised datasets.

References