Adversarial Domain Adaptation for Machine Reading Comprehension

Motivation & Contribution

- **Motivation**
  - Recent success in MRC relies on large-scale annotated in-domain data (e.g., SQuAD)
  - Directly adapting models from source domain to low-resource target domain performs poorly due to domain shift

- **Contribution**
  - Unsupervised Domain Adaptation by generating pseudo data on target domain and learning domain-invariant representations through adversarial learning

T-SNE plot of encoded feature representations

Without domain adaptation

With domain adaptation

AdaMRC Framework

- **Question Generator (QG):** using passage and answer (extracted by NER) as input for generating pseudo questions in the target domain
- **Encoder & Decoder:** source domain and target domain share the same encoder & decoder (i.e., MRC Module)
- **Discriminator:** an MLP as domain classifier. A gradient reverse layer is used for gradient backpropagation

Training Algorithm

**Algorithm 1** AdaMRC training procedure.

1. **Input:** source domain labeled data \( S = \{p^s, q^s, a^s\} \), target domain unlabeled data \( T = \{p^t\} \)
2. Train the MRC model \( \theta^S = (\theta^p, \theta^q) \) on source domain \( S \)
3. Train the QG model \( \theta^{QG} \) on source domain \( S \)
4. Generate \( T_{gen} = \{p^t, q^t, a^t\} \) using the QG model
5. Initialize \( \theta = (\theta^p, \theta^q, \theta^t) \) with \( \theta^p \)
6. for epoch \( < 1 \) to \( \theta^{epochs} \) do
7. Optimize \( \theta \) on \( S \cup T_{gen} \). Each minibatch is composed with \( k_s \) samples from \( S \) and \( k_t \) samples from \( T_{gen} \)
8. **end for**
9. **Output:** Model with the best performance on the target development set \( \theta^* \)

Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>EM/F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuAD + NewsQA</td>
<td>36.68/52.79</td>
</tr>
<tr>
<td>AdaMRC + SQuAD</td>
<td>38.46/54.20</td>
</tr>
<tr>
<td>AdaMRC + SQuAD + GT questions</td>
<td>39.37/54.63</td>
</tr>
<tr>
<td>NewsQA + SQuAD</td>
<td>56.83/68.62</td>
</tr>
<tr>
<td>AdaMRC + SQuAD</td>
<td>58.20/69.75</td>
</tr>
<tr>
<td>AdaMRC + SQuAD + GT questions</td>
<td>58.82/70.14</td>
</tr>
<tr>
<td>MS MARCO (v1) + SQuAD</td>
<td>13.06/25.80</td>
</tr>
<tr>
<td>AdaMRC + MS MARCO (v1) + SQuAD</td>
<td>14.09/26.09</td>
</tr>
<tr>
<td>AdaMRC + MS MARCO (v1) + SQuAD + GT questions</td>
<td>15.59/26.40</td>
</tr>
<tr>
<td>SQuAD</td>
<td>27.06/40.07</td>
</tr>
<tr>
<td>AdaMRC + SQuAD</td>
<td>27.92/41.47</td>
</tr>
</tbody>
</table>

Dataset Domain

- SQuAD (v1.1) Wiki
- NewsQA News
- MS MARCO (v1) Web

**Method**

- **Main results are based on Stochastic Answer Network (SAN)**
- AdaMRC consistently improves performance over baselines
- Direct data augmentation and fine-tuning (SynNet) hurts performance
- Question generation is effective (margin with "AdaMRC with GT questions" is relatively small)
- Generalizable to other datasets and other MRC models with consistent performance gain

**With pre-trained language models**

<table>
<thead>
<tr>
<th>Method</th>
<th>EM/F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaMRC</td>
<td>38.46/54.20</td>
</tr>
<tr>
<td>AdaMRC + SQuAD</td>
<td>40.96/56.25</td>
</tr>
<tr>
<td>AdaMRC + SQuAD + GT questions</td>
<td>42.00/58.71</td>
</tr>
<tr>
<td>AdaMRC + BERT-base</td>
<td>42.59/59.25</td>
</tr>
</tbody>
</table>

**Can be extended to semi-supervised setting**

- **Ratio (%Labeled data)**
  - 0%: 36.68/52.79
  - 5%: 47.61/62.69
  - 10%: 48.66/63.32
  - 20%: 50.75/64.80
  - 50%: 53.24/67.30
  - 100%: 56.48/69.14