Multi-source Meta Transfer for Low Resource MCQA

Ming Yan\textsuperscript{1}, Hao Zhang\textsuperscript{1,2}, Di Jin\textsuperscript{3}, Joey Tianyi Zhou\textsuperscript{1}

\textsuperscript{1} IHPC, A*STAR, Singapore
\textsuperscript{2} CSCE, NTU, Singapore
\textsuperscript{3} CSAIL, MIT, USA
Background

Low resource MCQA with data size under 100K

Corpus from different domains

Snippets
Newswire
Scenario Text
Wikipedia
Wikipedia Snippets
Newswire

Extractive/Abstractive
Multi-hop
MCQA

Data Size (K)

SEARCHQA
NEWSQA
SWAG
HOTPOTQA
SQUAD
RACE

Background

MCQA

Corpus from different domains

Extractive/Abstractive
Multi-hop
MCQA

Data Size (K)

SEARCHQA
NEWSQA
SWAG
HOTPOTQA
SQUAD
RACE

Corpus from different domains

Extractive/Abstractive
Multi-hop
MCQA

Data Size (K)

SEARCHQA
NEWSQA
SWAG
HOTPOTQA
SQUAD
RACE

Corpus from different domains

Extractive/Abstractive
Multi-hop
MCQA

Data Size (K)

SEARCHQA
NEWSQA
SWAG
HOTPOTQA
SQUAD
RACE

Corpus from different domains

Extractive/Abstractive
Multi-hop
MCQA

Data Size (K)

SEARCHQA
NEWSQA
SWAG
HOTPOTQA
SQUAD
RACE

Corpus from different domains

Extractive/Abstractive
Multi-hop
MCQA

Data Size (K)

SEARCHQA
NEWSQA
SWAG
HOTPOTQA
SQUAD
RACE

Corpus from different domains

Extractive/Abstractive
Multi-hop
MCQA

Data Size (K)

SEARCHQA
NEWSQA
SWAG
HOTPOTQA
SQUAD
RACE

Corpus from different domains

Extractive/Abstractive
Multi-hop
MCQA

Data Size (K)

SEARCHQA
NEWSQA
SWAG
HOTPOTQA
SQUAD
RACE

Corpus from different domains

Extractive/Abstractive
Multi-hop
MCQA

Data Size (K)

SEARCHQA
NEWSQA
SWAG
HOTPOTQA
SQUAD
RACE

Corpus from different domains

Extractive/Abstractive
Multi-hop
MCQA
How does meta learning work?

- Low resource setting
- Domains discrepancy

Transfer learning, multi-task learning
Fine-tuning on the target domain

init \( w_m \) from backbone model \( J : \text{cost function} \)

Support tasks: \( x_l \sim X \) Enquiry tasks: \( x_m \sim X \)

\[
\text{model}_l =: \text{copy}(\text{model}_m)
\]

\[
y_l = \text{model}_l(w_l, x_l)
\]

\[
w_l =: w_l + \alpha \frac{\partial J_l}{\partial w_l}
\]

\[
y_m = \text{model}_l(w_l, x_m)
\]

\[
w_m =: w_m + \alpha \frac{\partial J_m}{\partial w_m}
\]

Source 1  Source 2  Source 3  Target

How does meta learning work?

*init* $w_m$ from pretrained model

Support $T$: $x_l \sim X$  \hspace{1cm} Enquiry $T$: $x_m \sim X$  \hspace{1cm} $J$: cost function

Learn a model that can generalize over the task distribution.

Multi-source Meta Transfer

- Learn knowledge from multiple sources
- Reduce discrepancy between sources and target.
Multi-source Meta Transfer

Multi-source Meta Learning (MML)
Learn knowledge from multiple sources.
Learn a representation near to the target.

Multi-source Transfer Learning (MTL)
Finetune meta-model to the target source.
How MMT samples the task?

Algorithm 1: The procedure of MMT

**Input:** Task distribution over source $p^s(\tau)$, data distribution over target $P^t(\tau)$, backbone model $f(\theta)$, learning rates in MMT $\alpha, \beta, \lambda$

**Output:** Optimized parameters $\theta$

Initial the value of $\theta$

**While not done do**

**for all source $S$ do**

Sample batch of tasks $\tau_i^s \sim p^s(\tau)$

for all $\tau_i^s$ do

Evaluate $\nabla_{\theta} L_{\tau_i^s}(f(\theta))$ with respect to k examples

Compute gradient for fast adaption:

$$\theta' = \theta - \alpha \nabla_{\theta} L_{\tau_i^s}(f(\theta))$$

end

Meta model update:

$$\theta = \theta - \beta \nabla_{\theta} \sum_{\tau_i^s \sim p^s(\tau)} L_{\tau_i^s}(f(\theta'))$$

Get batch of data $\tau_i^t \sim p^t(\tau)$

**for all $\tau_i^t$ do**

Evaluate $\nabla_{\theta} L_{\tau_i^t}(f(\theta))$ with respect to k examples

Gradient for target fine-tuning:

$$\theta = \theta - \beta \nabla_{\theta} L_{\tau_i^t}(f(\theta))$$

end

end

Get all batches of data $\tau_i^t \sim p^t(\tau)$

**for all $\tau_i^t$ do**

Evaluate with respect to batch size;

Gradient for meta transfer learning:

$$\theta = \theta - \beta \nabla_{\theta} L_{\tau_i^t}(f(\theta))$$

end
Multi-source Meta Transfer

Algorithm 1: The procedure of MMT

Input: Task distribution over source \( p^s(\tau) \), data distribution over target \( p^t(\tau) \), backbone model \( f(\theta) \), learning rates in MMT \( \alpha, \beta, \lambda \)

Output: Optimized parameters \( \theta \)

Initial the value of \( \theta \)

While not done do
  for all source \( S \) do
    Sample batch of tasks \( \tau^s_i \sim p^s(\tau) \)
    for all \( \tau^s_i \) do
      Evaluate \( \nabla_{\theta} L_{\tau^s_i}(f(\theta)) \) with respect to \( k \) examples
      Compute gradient for fast adaption:
      \[
      \theta' := \theta - \alpha \nabla_{\theta} L_{\tau^s_i}(f(\theta))
      \]
    end
    Meta model update:
    \[
    \theta := \theta - \beta \nabla_{\theta} \sum_{\tau^s_i \sim p^s(\tau)} L_{\tau^s_i}(f(\theta'))
    \]
    Get batch of data \( \tau^t_i \sim p^t(\tau) \)
    for all \( \tau^t_i \) do
      Evaluate \( \nabla_{\theta} L_{\tau^t_i}(f(\theta')) \) with respect to \( k \) examples
      Gradient for target fine-tuning:
      \[
      \theta := \theta - \beta \nabla_{\theta} L_{\tau^t_i}(f(\theta))
      \]
    end
  end
  Get all batches of data \( \tau^t_i \sim p^t(\tau) \)
  for all \( \tau^t_i \) do
    Evaluate with respect to batch size;
    Gradient for meta transfer learning:
    \[
    \theta := \theta - \beta \nabla_{\theta} L_{\tau^t_i}(f(\theta))
    \]
  end
end

MTL

S1

S2

Target

S3

S4

Target

Transfer meta model to the target

MMT is agnostic to backbone models

Support task and Query task sampled from the same distribution

Updated the learner (\( \theta' \)) on support task

Updated the meta model (\( \theta \)) on query task

Updated the meta model (\( \theta \)) on target data

MTL

MML
## Results

### Performance of Supervised MMT

<table>
<thead>
<tr>
<th>Methods</th>
<th>DREAM</th>
<th>MCTEST</th>
<th>SemEval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dev</td>
<td>Test</td>
<td>Dev</td>
</tr>
<tr>
<td>CoMatching (Wang et al., 2018)</td>
<td>45.6</td>
<td>45.5</td>
<td>-</td>
</tr>
<tr>
<td>HFL (Chen et al., 2018)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>QACNNN (Chung et al., 2018)</td>
<td>-</td>
<td>-</td>
<td>72.66</td>
</tr>
<tr>
<td>IMC (Yu et al., 2019)</td>
<td>-</td>
<td>76.59</td>
<td>-</td>
</tr>
<tr>
<td>XLNet (Yang et al., 2019)</td>
<td>-</td>
<td>72.0</td>
<td>-</td>
</tr>
<tr>
<td>GPT+Strategies (2×) (Sun et al., 2019b)</td>
<td>-</td>
<td>-</td>
<td>81.9</td>
</tr>
<tr>
<td>BERT-Base</td>
<td>60.05</td>
<td>61.58</td>
<td>70.0</td>
</tr>
<tr>
<td>RoBERTa†</td>
<td>82.16</td>
<td>84.37</td>
<td>88.37</td>
</tr>
<tr>
<td>MMT (BERT-Base)</td>
<td>68.38</td>
<td>68.89</td>
<td>81.56</td>
</tr>
<tr>
<td>MMT (RoBERTa)†</td>
<td>83.87</td>
<td>85.55</td>
<td>88.66</td>
</tr>
</tbody>
</table>

### Performance of Unsupervised MMT

<table>
<thead>
<tr>
<th>Method</th>
<th>Sup.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bert-Base</td>
<td>Yes</td>
<td>67.98</td>
</tr>
<tr>
<td>QACNN (Chung et al., 2018)</td>
<td>Yes</td>
<td>72.66</td>
</tr>
<tr>
<td>IMC (Yu et al., 2019)</td>
<td>Yes</td>
<td>76.59</td>
</tr>
<tr>
<td>MemN2N (Chung et al., 2018)</td>
<td>No</td>
<td>53.39</td>
</tr>
<tr>
<td>QACNN (Chung et al., 2018)</td>
<td>No</td>
<td>63.10</td>
</tr>
<tr>
<td>TL(S)</td>
<td>No</td>
<td>50.02</td>
</tr>
<tr>
<td>TL(R)</td>
<td>No</td>
<td>77.02</td>
</tr>
<tr>
<td>TL(R-S)</td>
<td>No</td>
<td>62.97</td>
</tr>
<tr>
<td>TL(S-R)</td>
<td>No</td>
<td>77.38</td>
</tr>
<tr>
<td>TL(R+S)</td>
<td>No</td>
<td>79.17</td>
</tr>
<tr>
<td>Unsupervised MMT(S+R)</td>
<td>No</td>
<td><strong>81.55</strong></td>
</tr>
</tbody>
</table>

### MCTEST Performance of Unsupervised MMT

<table>
<thead>
<tr>
<th>Method</th>
<th>Dream</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-Base</td>
<td></td>
<td>60.05</td>
<td>61.58</td>
</tr>
<tr>
<td>+MML(M)</td>
<td></td>
<td>49.85</td>
<td>52.87</td>
</tr>
<tr>
<td>+MML(R)</td>
<td></td>
<td>49.56</td>
<td>51.69</td>
</tr>
<tr>
<td>+MML(M∪R)</td>
<td></td>
<td>29.60</td>
<td>29.20</td>
</tr>
<tr>
<td>+TL(M)</td>
<td></td>
<td>60.31</td>
<td>60.14</td>
</tr>
<tr>
<td>+TL(R)</td>
<td></td>
<td>68.72</td>
<td>67.72</td>
</tr>
<tr>
<td>+TL(R-M)</td>
<td></td>
<td>68.97</td>
<td>67.38</td>
</tr>
<tr>
<td>+TL(M+R)</td>
<td></td>
<td>68.61</td>
<td>68.15</td>
</tr>
<tr>
<td>+MML(M)</td>
<td></td>
<td>67.99</td>
<td>68.54</td>
</tr>
<tr>
<td>+MML(R)</td>
<td></td>
<td>68.04</td>
<td>68.69</td>
</tr>
<tr>
<td>+MML(M∪R)</td>
<td></td>
<td>61.72</td>
<td>60.12</td>
</tr>
<tr>
<td>MMT(M+R)</td>
<td></td>
<td><strong>68.38</strong></td>
<td><strong>68.89</strong></td>
</tr>
</tbody>
</table>

### MMT Ablation Study
How to select sources?

### Transferability Matrix

<table>
<thead>
<tr>
<th>Sources</th>
<th>Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREAM</td>
<td>0, 6.6, 0.31, 0.16, -0.69</td>
</tr>
<tr>
<td>RACE</td>
<td>-0.58, 0, 2.1, 0.35, 0.03</td>
</tr>
<tr>
<td>MCTEST</td>
<td>2.9, 14, 0, 1.8, 1.4</td>
</tr>
<tr>
<td>SemEval</td>
<td>0.14, 1.1, 1.1, 0, -0.51</td>
</tr>
<tr>
<td>SWAG</td>
<td>-0.55, -1.9, -0.75, -0.57, 0</td>
</tr>
</tbody>
</table>

### T-SNE Visualization of BERT Feature

Test on SemEval 2018

![T-SNE Visualization of BERT Feature](image)
Takeaways

- MMT extends to meta learning to multi-source on MCQA task
- MMT provided an algorithm both for supervised and unsupervised meta training
- MMT give a guideline to source selection