Multi-source Meta Transfer for Low Resource Multiple-Choice Question Answering

Daeung Kim

2020.09.18.

dukim@dongguk.edu

INDEX

1. **Prerequisites**

2. **Multi-source Meta Transfer**

3. **Experiments & Results**

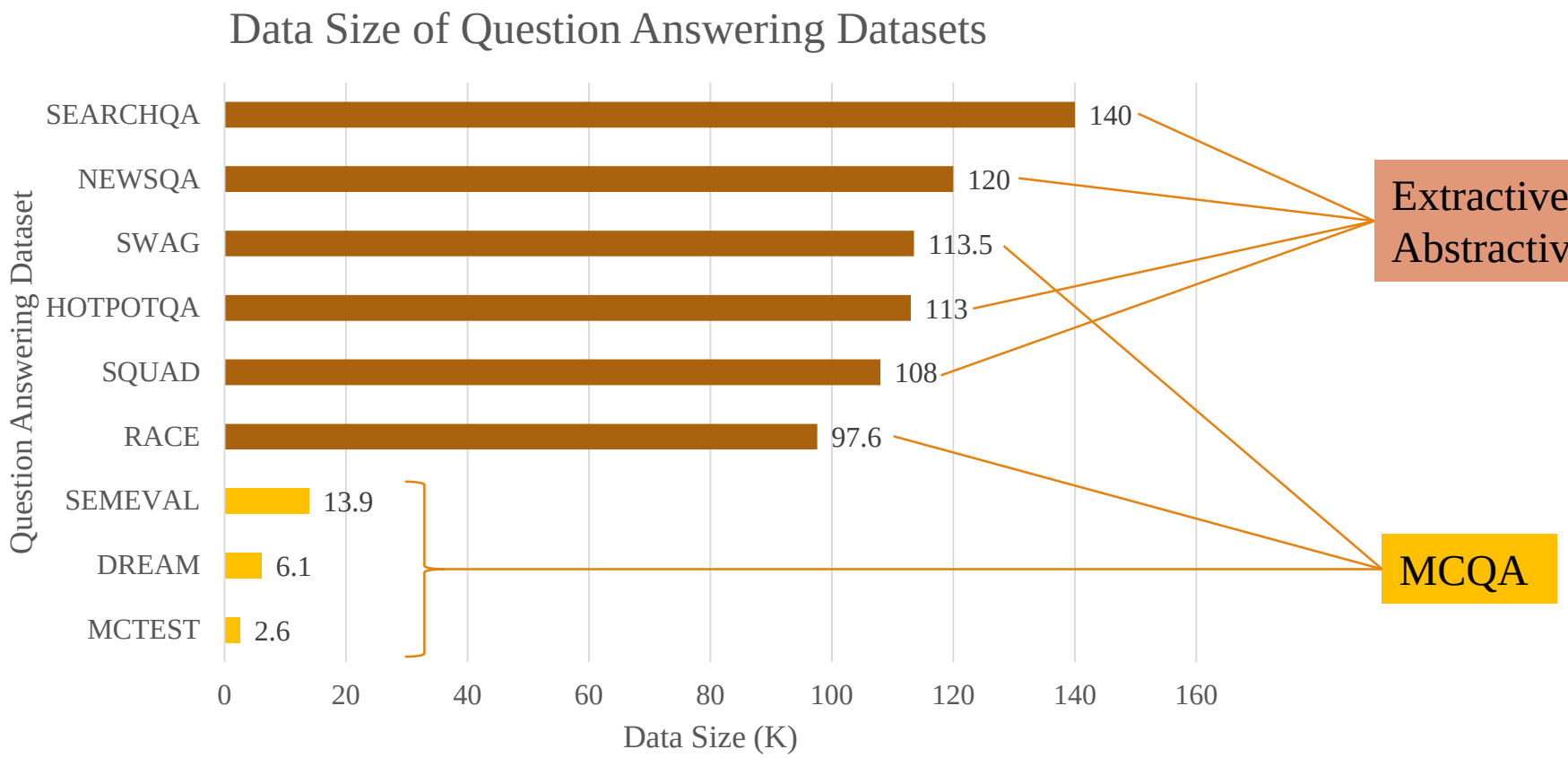
4. **Conclusions**

1. Prerequisites

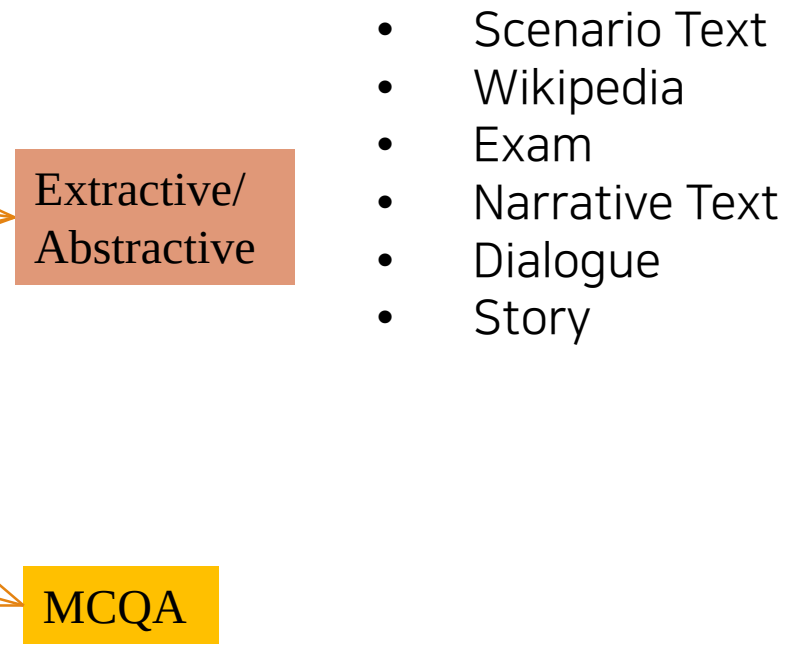
1. Prerequisites

Low resource & Domain discrepancy

Most existing MCQA datasets are small in size



Corpus from different domains



https://slideslive.com/38929127/multisource-meta-transfer-for-low-resource-multiplechoice-question-answering

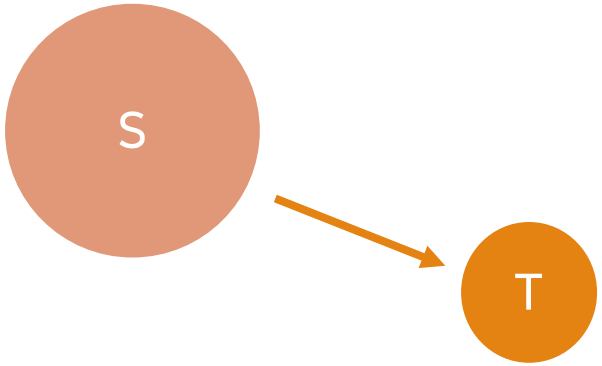
1. Prerequisites

How does meta learning work?

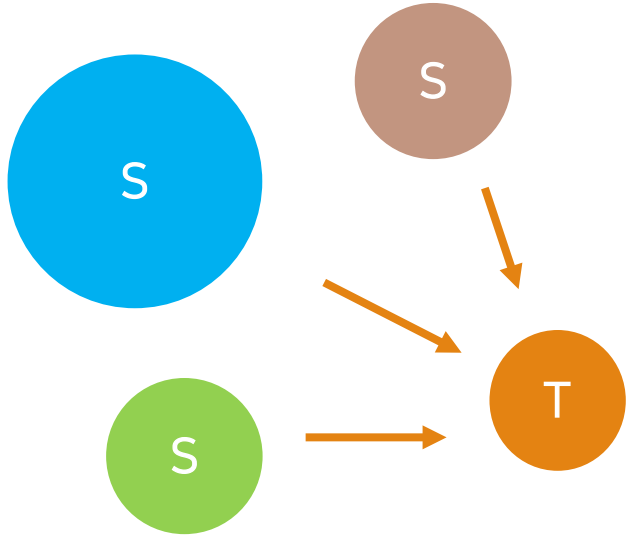
- Problem
  - Low resource setting
  - Domain discrepancy

- Existing Methods
  - Transfer learning
  - Multi-task learning

  Fine-tuning on the target domain



Transfer Learning



Multi-task learning

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
Jin, Di, et al. "Mmm: Multi-stage multi-task learning for multi-choice reading comprehension." *arXiv preprint arXiv:1910.00458* (2019).

1. Prerequisites

How does meta learning work?

- Meta-Learning
  - Model-Agnostic Meta Learning

$J: loss\ function$
$Support\ tasks: x_l \sim X\ Enquiry\ tasks: x_m \sim X$

$init\ w_m$ from $backbone\ model$		$model_l =: \boldsymbol{copy}(model_m)$
Fast Adaptation	$\mathrm{FF_L}$	$y_l = model_l(w_l, x_l)$
	$\mathrm{BP_L}$	$w_l =: w_l + \alpha \frac{\partial J_l}{\partial w_l}$
Meta-Learning	$\mathrm{FF_L}$	$y_m = model_l(w_l, x_m)$
	$\mathrm{BP_L}$	$w_m =: w_m + \alpha \frac{\partial J_m}{\partial w_m}$

Algorithm 1 Model-Agnostic Meta-Learning

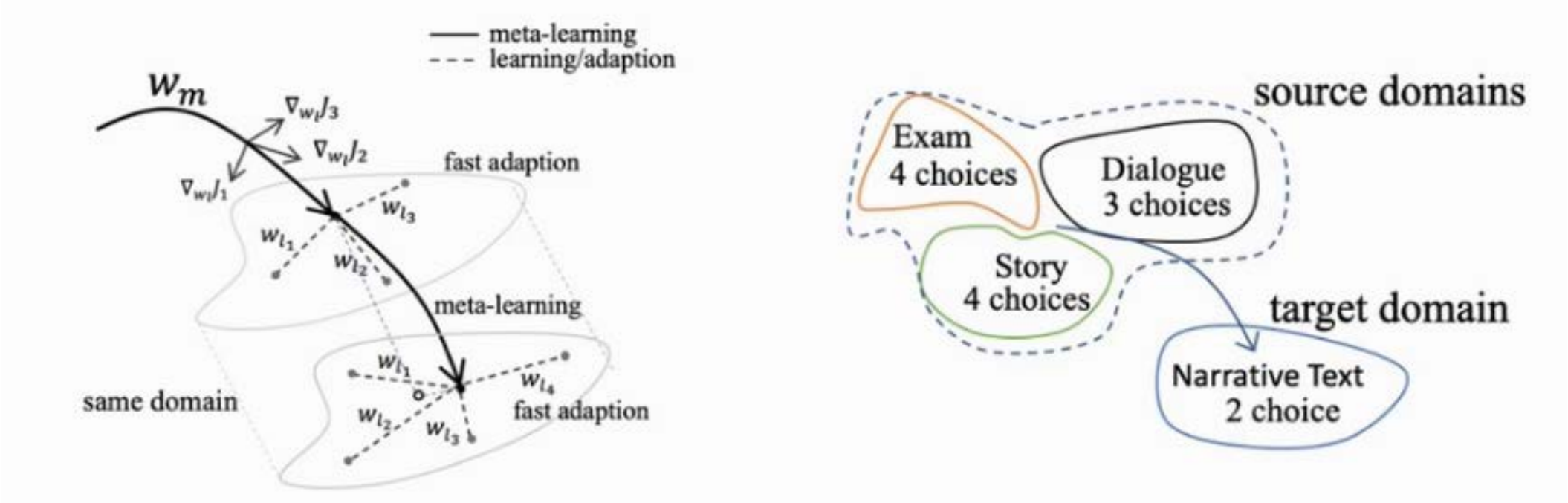
Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
6: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
7: **end for**
8: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
9: **end while**

Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." arXiv preprint arXiv:1703.03400 (2017).

1. Prerequisites

How does meta learning work?

- Learn a model that can generalize over the task distribution

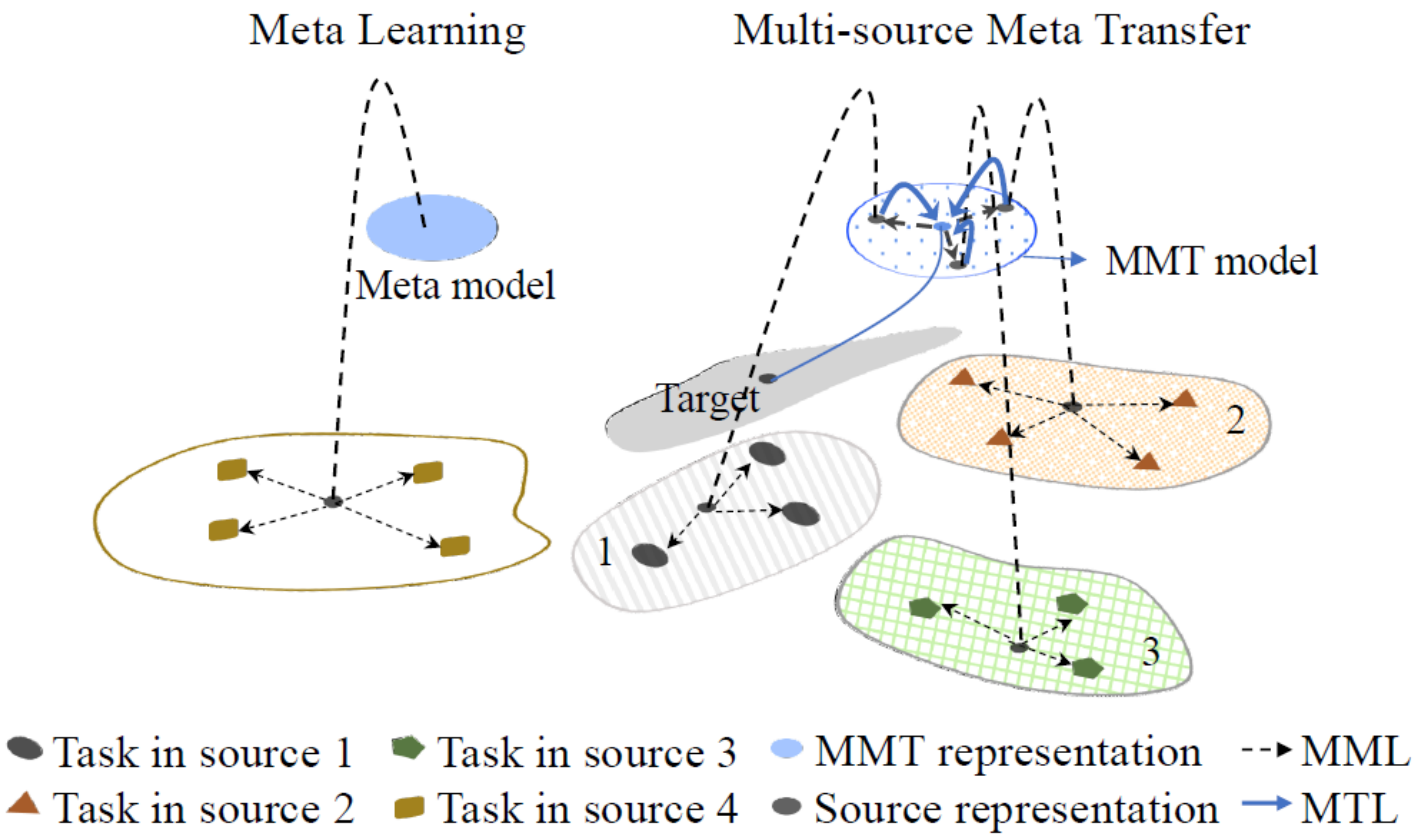


2. Multi-source Meta Transfer

2. Multi-source Meta Transfer(MMT)

Meta-Learning vs. Multi-source Meta Transfer(MMT)

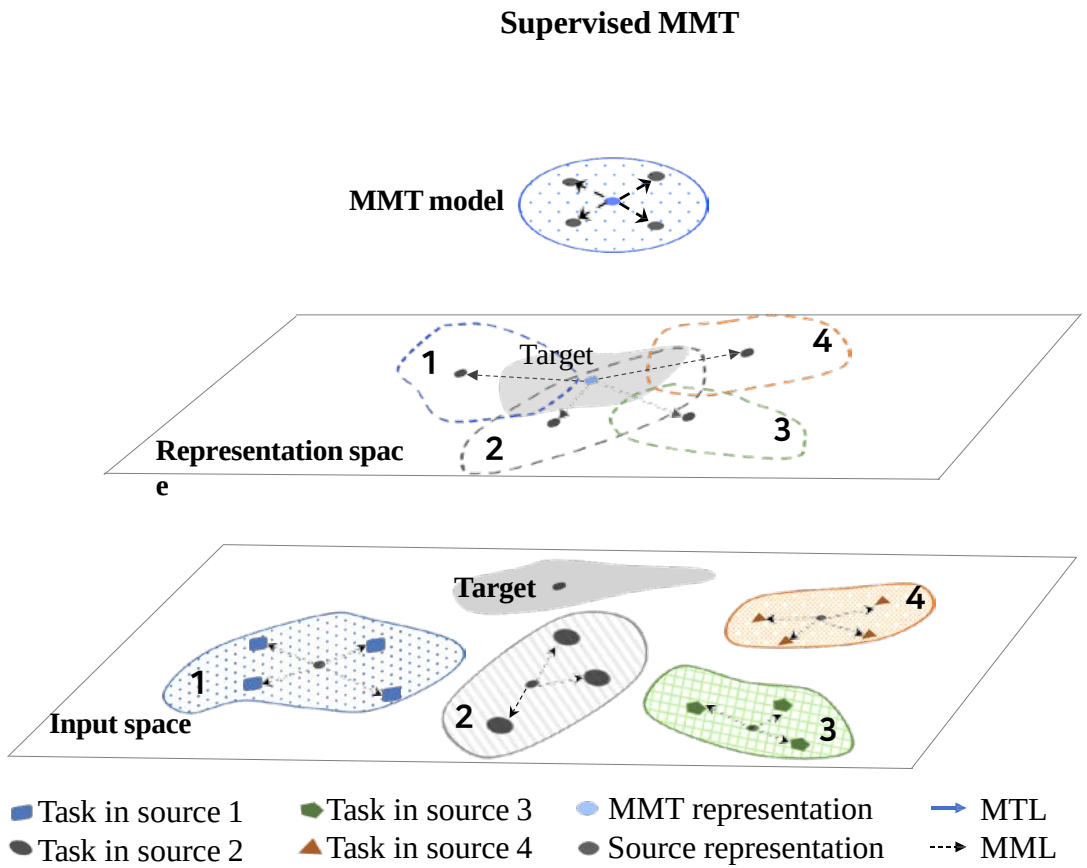
- MMT learns knowledge from multiple source
- Reduce discrepancy between sources and target



2. Multi-source Meta Transfer

Multi-source Meta Transfer

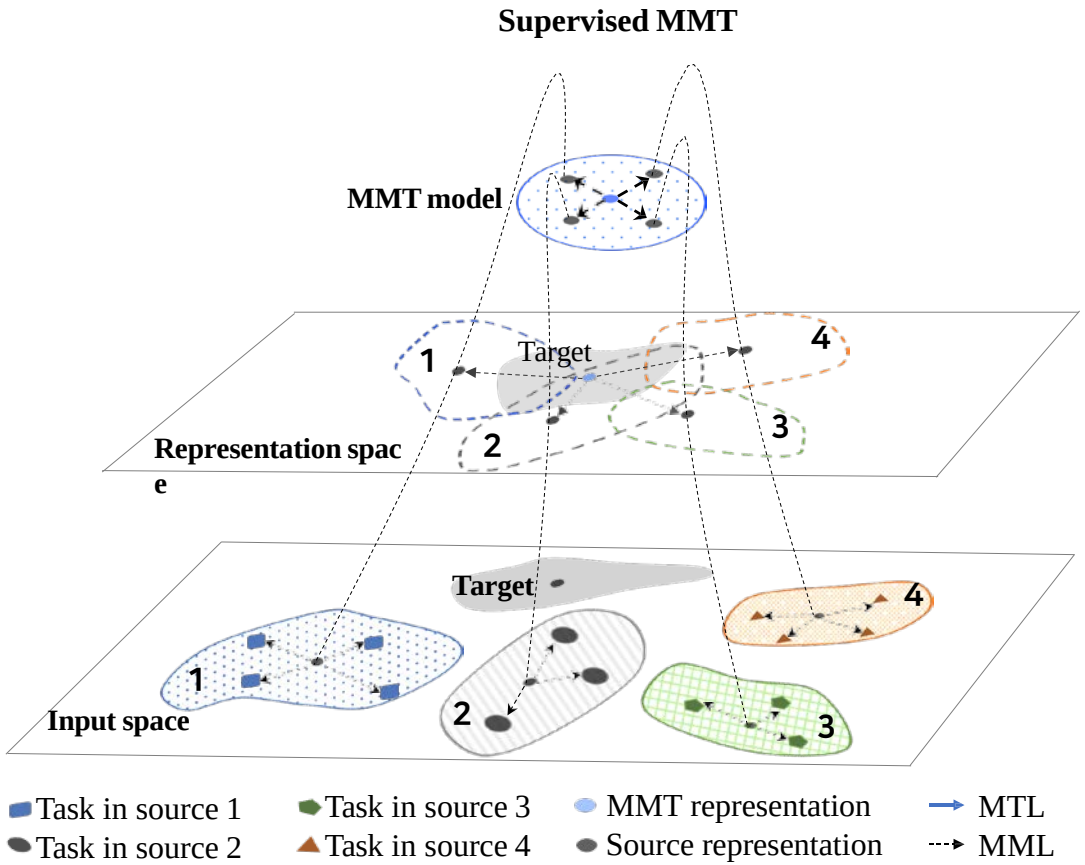
1. Multi-source Meta Learning(MML)
2. Meta-Transfer Learning(MTL)



2. Multi-source Meta Transfer

1. Multi-source Meta Learning(MML)

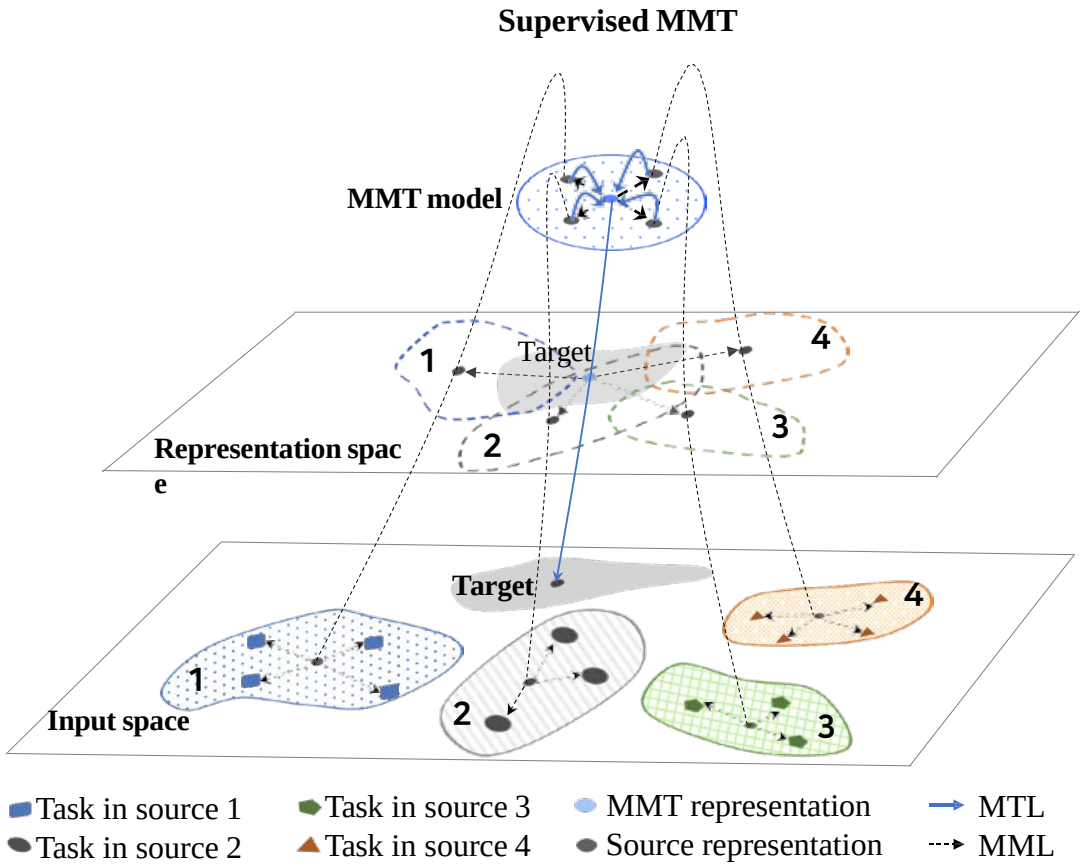
- Learn knowledge from multiple sources
- Learn a representation near to the target



2. Multi-source Meta Transfer

2. Meta-Transfer Learning(MTL)

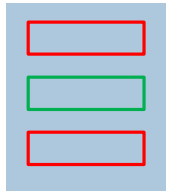
- Finetune meta-model to the target source



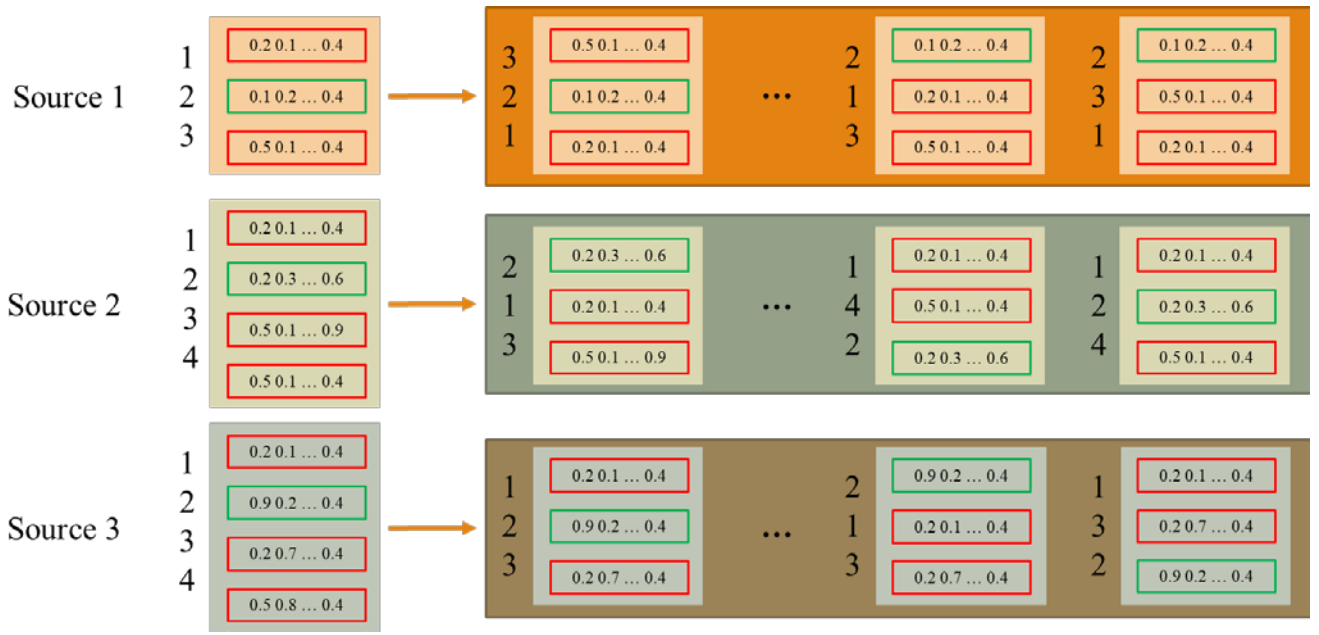
2. Multi-source Meta Transfer

How MMT samples the task?

- Samples the number of choices equal to the number of choices in the target task.
- The correct answer choice must be included, and the number of choices that the source task has must be equal to or greater than the number of choices of the target task.

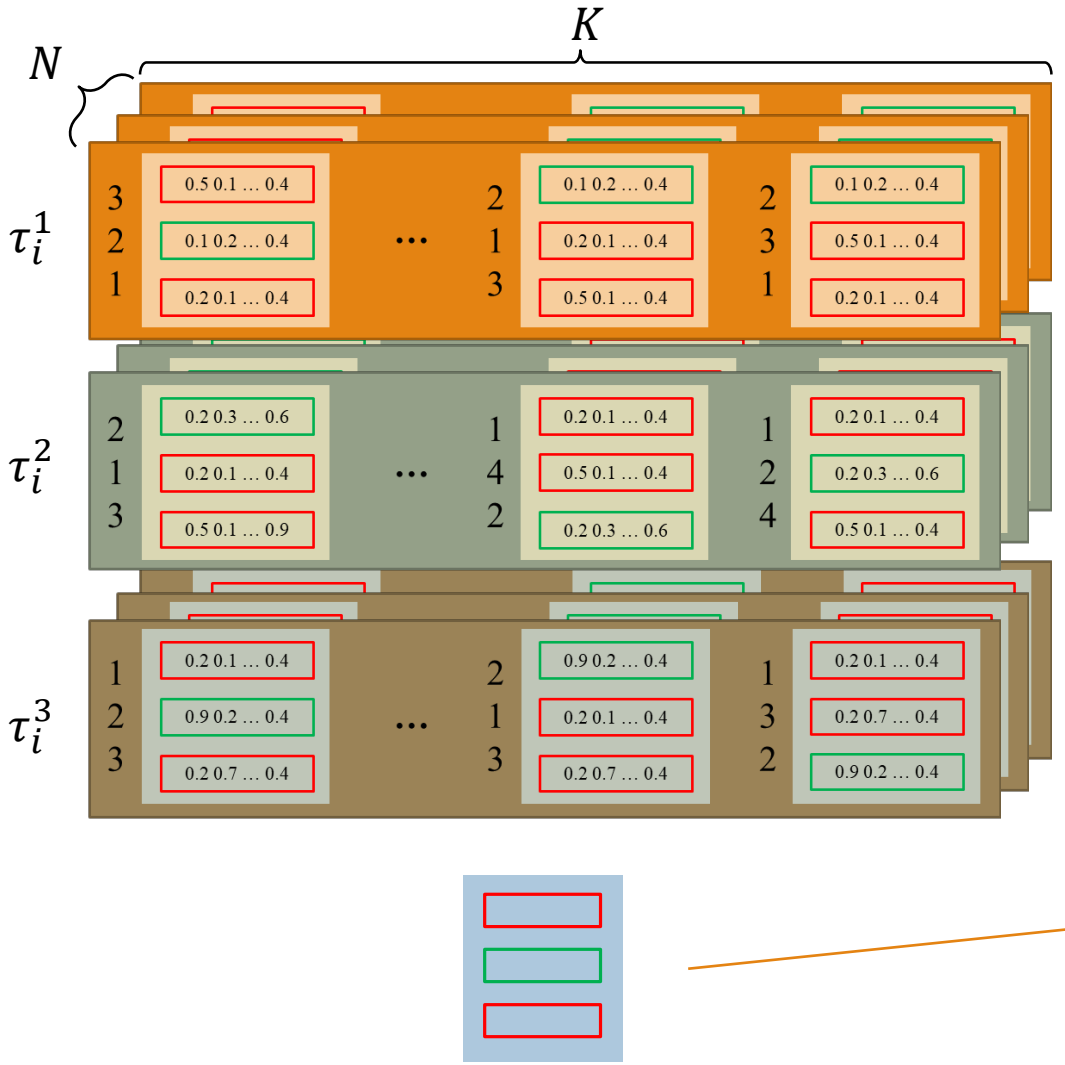


Target Task
(MCQA task that has 3 choices)



2. Multi-source Meta Transfer

How MMT samples the task?



Target 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 α, β, λ.

Output: Optimized parameters θ.

Initialize θ from backbone model;
while *not done* **do**
  for *all source* S **do**
    Sample batch of tasks $\tau_i^s \sim p^s(\tau)$;
    for *all* τ_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* τ_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
end
Get all batches of data $\tau_i^t \sim p^t(\tau)$;
for *all* τ_i^t **do**
  Evaluate $\nabla_\theta L_{\tau_i^t}(f(\theta))$ with respect to batch size;
  Gradient for meta transfer learning: $\theta = \theta - \lambda \nabla_\theta L_{\tau_i^t}(f(\theta))$;
end

2. Multi-source Meta Transfer

How MMT samples the task?

MML

- MMT is agnostic to backbone models
- Sequentially do below algorithm on source datasets
  - Sample Support task and Query task from the same distribution
  - Update the learner (θ') on support task
  - Update the meta model (θ') on query task
  - Update the meta model (θ') on target data

MTL

- Transfer meta model to the target

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 α, β, λ.

Output: Optimized parameters θ.

MML

Initialize θ from backbone model;
while *not done* **do**
  for *all source* S **do**
    Sample batch of tasks $\tau_i^s \sim p^s(\tau)$;
    for *all* τ_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* τ_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
end

MTL

Get all batches of data $\tau_i^t \sim p^t(\tau)$;
for *all* τ_i^t **do**
  Evaluate $\nabla_\theta L_{\tau_i^t}(f(\theta))$ with respect to batch size;
  Gradient for meta transfer learning: $\theta = \theta - \lambda \nabla_\theta L_{\tau_i^t}(f(\theta))$;
end

3. Experiments & Results

3. Experiments & Results

Experiments

- Settings
  - The backbone model : BERT & RoBERTa
  - The maximal sequence input length : 512 for BERT, 256 for RoBERTa
  - The model optimization : Adam, initial lr of fast adapt : 1e-3, the rest ones are set to 1e-5
- Dataset

Name	DREAM	RACE	MCTEST	SemEval	SWAG
Type	Dialogue	Exam	Story	Narrative Text	Scenario Text
Ages	15+	12-18	7+	-	-
Generator	Expert	Expert	Crowd.	Crowd.	AF./Crowd.
Level	High School/College	High/Middle School	Children	Unlimited	Unlimited
Choices	3	4	4	2	4
Samples	6,444	27,933	660	2,119	92,221
Questions	10,197	97,687	2,640	13,939	113,557

Table 1: Statistics of MCQA datasets, where "Crowd." denotes questions generated by crowd-sourcing, and "AF." denotes question generated by adversarial filtering.

3. Experiments & Results

Results

- MMT(RoBERTa) achieves the best performances overall benchmark datasets
- MMT is able to boost up performance over different pre-trained language models
- MMT is backbone-free, It can improve the performance with the advance of LMs

Methods	DREAM		MCTEST		SemEval	
	Dev	Test	Dev	Test	Dev	Test
CoMatching (Wang et al., 2018)	45.6	45.5	-	-	-	-
HFL (Chen et al., 2018)	-	-	-	-	86.46	84.13
QACNN (Chung et al., 2018)	-	-	-	72.66	-	-
IMC (Yu et al., 2019)	-	-	-	76.59	-	-
XLNet (Yang et al., 2019)	-	72.0	-	-	-	-
GPT+Strategies (2×) (Sun et al., 2019b)	-	-	-	81.9	-	89.5
BERT-Base	60.05	61.58	70.0	67.98	86.03	87.53
RoBERTa†	82.16	84.37	88.37	87.26	93.76	94.00
MMT (BERT-Base)	**68.38**	**68.89**	**81.56**	**82.02**	**88.52**	**88.85**
MMT (RoBERTa)†	**83.87**	**85.55**	**88.66**	**88.80**	**94.33**	**94.24**

Table 2: Comparison with state-of-the-art methods in MCQA datasets, where "†" denotes the maximal sequence length of RoBERTa-large is limited to 256.

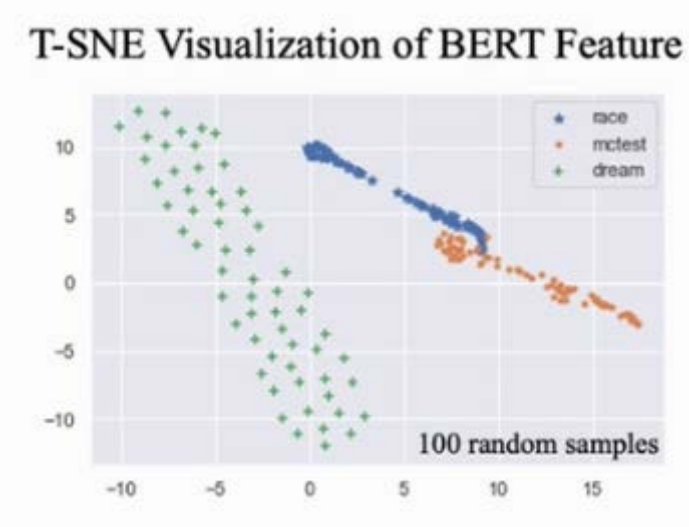
Method	Sup.	Test
Bert-Base	Yes	67.98
QACNN (Chung et al., 2018)	Yes	72.66
IMC (Yu et al., 2019)	Yes	76.59
MemN2N (Chung et al., 2018)	No	53.39
QACNN (Chung et al., 2018)	No	63.10
TL(S)	No	50.02
TL(R)	No	77.02
TL(R-S)	No	62.97
TL(S-R)	No	77.38
TL(R+S)	No	79.17
Unsupervised MMT(S+R)	No	**81.55**

Table 3: Unsupervised domain adaptation on MCTEST. "Sup." denotes supervised, "S" denotes SWAG, "R" denotes RACE, and "TL(∗)" denotes transfer learning from different datasets to MCTEST. For example, "TL(R-S)" denotes that Bert-Base is first fine-tuned on RACE, then on SWAG. Unsupervised MMT(S+R) denotes that the meta model is trained on the sources of SWAG and RACE.

3. Experiments & Results

How to select sources?

- Source selection is prerequisite step for MMT
- Dissimilar data sources will cause negative transfer when their distribution is far away from the target one.
- To handle this, the model learns from dissimilar sources to similar one



4. Conclusion

4. Conclusion

- MMT extends meta learning to multiple sources on MCQA task
- MMT provides an algorithm both for supervised and unsupervised meta training
- MMT gives a guideline to source selection

dongguk UNIVERSITY

Q & A