An Iterative Multi-Source Mutual Knowledge Transfer Framework for Machine Reading Comprehension

Xin Liu[1*], **Kai Liu**[2*], **Xiang Li**[3*], **Jinsong Su**[1†], **Yubin Ge**[4], **Bin Wang**[3] and **Jiebo Luo**[5]
[1]Xiamen University, Xiamen, China
[2]Baidu Inc., Beijing, China
[3]Xiaomi AI Lab, Xiaomi Inc., Beijing, China
[4]University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
[5]Department of Computer Science, University of Rochester, Rochester NY 14627, USA
liuxin@stu.xmu.edu.cn, liukai20@baidu.com, lixiang21@xiaomi.com, jssu@xmu.edu.cn

Abstract

The lack of sufficient training data in many domains, poses a major challenge to the construction of domain-specific machine reading comprehension (MRC) models with satisfying performance. In this paper, we propose a novel iterative multi-source mutual knowledge transfer framework for MRC. As an extension of the conventional knowledge transfer with one-to-one correspondence, our framework focuses on the many-to-many mutual transfer, which involves synchronous executions of multiple many-to-one transfers in an iterative manner. Specifically, to update a target-domain MRC model, we first consider other domain-specific MRC models as individual teachers, and employ knowledge distillation to train a multi-domain MRC model, which is differentially required to fit the training data and match the outputs of these individual models according to their domain-level similarities to the target domain. After being initialized by the multi-domain MRC model, the target-domain MRC model is fine-tuned to match both its training data and the output of its previous best model simultaneously via knowledge distillation. Compared with previous approaches, our framework can continuously enhance all domain-specific MRC models by enabling each model to iteratively and differentially absorb the domain-shared knowledge from others. Experimental results and in-depth analyses on several benchmark datasets demonstrate the effectiveness of our framework. We release our code at https://github.com/DeepLearnXMU/IMM

1 Introduction

As one of the core abilities of artificial intelligence, Machine Reading Comprehension (MRC) attempts to enable machines to answer questions after reading a passage. Due to its valuable industry applications such as search engines, it has always been one of the research focuses in natural language processing [Seo *et al.*, 2017; Hu *et al.*, 2018; Yu *et al.*, 2018; Devlin *et al.*, 2019; Liu *et al.*, 2020]. As the basis for MRC studies, many datasets such as SQuAD [Rajpurkar *et al.*, 2016] and NewsQA [Trischler *et al.*, 2017] have been developed in recent years. However, some domain-specific datasets are limited, and thus they are unable to individually train a domain-specific MRC model with satisfying performance. Therefore, how to overcome the shortage of domain-specific training data has become one of the important research directions in MRC.

To this end, one direction is to mix multiple domains of data and train a unified MRC model [Xu *et al.*, 2019] and another line of work introduces silver data by automatically generating questions [Duan *et al.*, 2017; Song *et al.*, 2018] . However, such approaches ignore the differences between these domains and introduce noises, resulting in the underutilization of these data. Recently, researchers have explored several transfer learning methods for MRC. Min *et al.* [2017] and Chung *et al.* [2018] first introduced fine-tuning into MRC. Furthermore, Talmor and Berant [2019] investigated the effect of domain-level similarities on fine-tuning. Despite their successes, fine-tuning still has some drawbacks. First, it is a knowledge transfer method with one-to-one correspondence, which also cannot effectively utilize multiple source-domain data with different impacts. Second, the significant difference between the source-domain and the target-domain often makes the one-pass transfer procedure of fine-tuning fail to fully exploit domain-shared knowledge. It can be said how to fully exploit the training data in different domains to construct domain-specific MRC models still remains unresolved.

In this work, we propose a novel iterative multi-source mutual knowledge transfer framework for MRC. The intuition behind our method includes two aspects. First, each domain-specific MRC model can be enhanced by the training data and models of other domains, whose effects depend on their domain-level similarities to the target domain. Second, a process of iterative mutual reinforcement can achieve better knowledge transfer, where domain-shared knowledge among training data in different domains can be fully exploited to

*These authors contributed equally to this work.
†Corresponding author

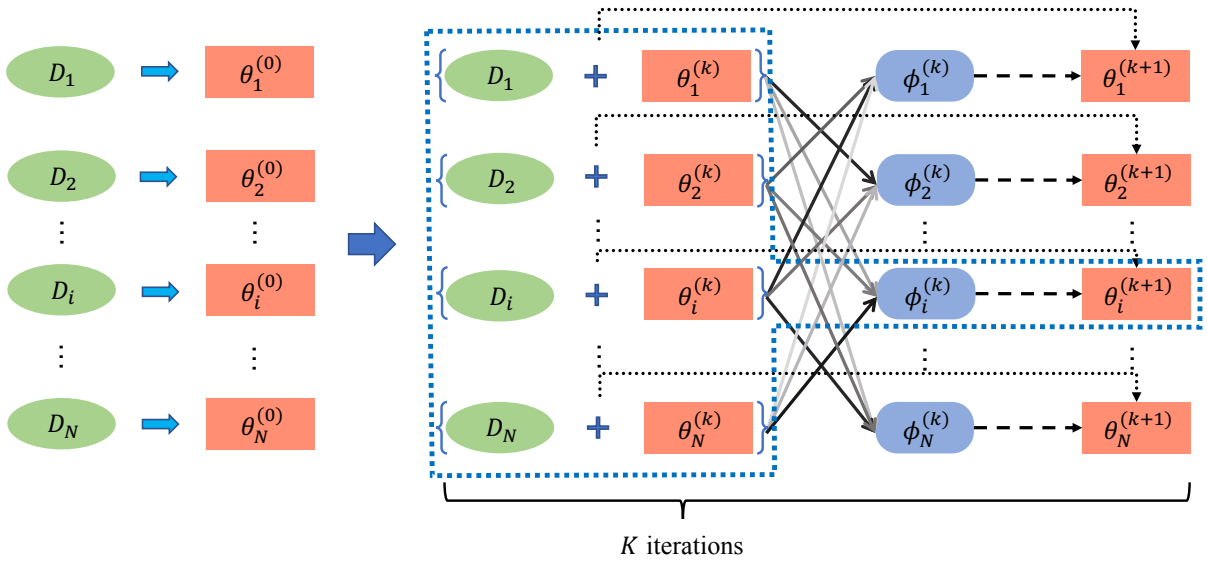

Figure 1: Overview of our framework for MRC. D_i is the i-the domain training data, $\theta_i^{(k)}$ and $\phi_i^{(k)}$ represent the i-th domain-specific model and its corresponding multi-domain model at the k-th iteration, respectively. Solid arrows show the distillation based knowledge transfer from $\{D_j, \theta_j^{(k)}\}_{j \neq i}$ to $\phi_i^{(k)}$, where color intensities of solid arrows indicate different domain-level similarities of multiple training data to the i-th domain data. The dotted arrow pointing from $\phi_i^{(k)}$ to $\theta_i^{(k+1)}$ indicates initializing $\theta_i^{(k+1)}$ using $\phi_i^{(k)}$. The dotted arrow departing from $\{D_i, \theta_i^{(k)}\}$ to $\theta_i^{(k+1)}$ illustrates the distillation based knowledge transfer from $\{D_i, \theta_i^{(k)}\}$ to $\theta_i^{(k+1)}$. Please note that each domain-specific model has its own multi-domain model.

benefit all domain-specific MRC models.

Figure 1 shows our framework for MRC. Significantly different from previous approaches, the execution process of our framework involves synchronous executions of multiple many-to-one knowledge transfers in an iterative manner. More specifically, as shown in the blue dashed part of Figure 1, we conduct a many-to-one transfer involving two steps to update a target-domain MRC model: **First**, we apply *knowledge distillation* [Hinton *et al.*, 2015; Tan *et al.*, 2019; Clark *et al.*, 2019] to establish a multi-domain MRC model, which is trained to simultaneously fit the training data and match the outputs of other domain-specific MRC models. Particularly, we introduce domain-level similarities to control the impacts from the other domains on the construction of this multi-domain MRC model. **Second**, after the initialization with the multi-domain MRC model, we employ knowledge distillation to fine-tune the target-domain MRC model, supervised by both its training data and the outputs of its previous best model.

Compared with previous methods [Min *et al.*, 2017; Chung *et al.*, 2018; Talmor and Berant, 2019], in this work, we make the following contributions: (1) We extend the existing knowledge transfer with one-to-one correspondence into a new setting of many-to-many mutual knowledge transfer for MRC; (2) We exploit multi-source mutual knowledge transfer for MRC, based on knowledge distillation, to particularly leverage the similarity between each domain and the target domain for better knowledge transfer; and (3) We propose to perform multi-source mutual knowledge transfer in an iterative manner, so that domain-shared knowledge can be fully exploited to benefit all domain-specific MRC models.

We conduct experiments on several commonly-used datasets. Experimental results and in-depth analyses show that our framework achieves significantly better performance than the dominant knowledge transfer approaches for MRC.

2 Related Work

Transfer learning for MRC. To address the lack of large-scale domain-specific training data in MRC, previous studies use upstream datasets to enhance the performance of MRC models, including word embeddings [Pennington *et al.*, 2014] and language models [Devlin *et al.*, 2019]. Meanwhile, more attempts have been made to explore transfer learning approaches for MRC, where fine-tuning is the most common method achieving satisfying results on several datasets [Min *et al.*, 2017]. Particularly, Golub *et al.* [2017] introduced a two-stage synthesis network to produce the target-domain synthetic dataset, which can be used to fine-tune the MRC model. Chung *et al.* [2018] demonstrated that transfer learning is helpful even in unsupervised scenarios. Xu *et al.* [2019] explored a multi-task learning framework to exploit different domains of datasets for MRC, where two re-weighting strategies are investigated. Talmor and Berant [2019] performed a thorough empirical investigation of generalization and transfer over different domains of MRC datasets.

Unlike these work, our framework iteratively transfers the knowledge among multiple MRC datasets, where each domain-specific MRC model and its training data can be iteratively and differentially exploited to enhance other models. To the best of our knowledge, such an approach has not been fully explored before in MRC.

Knowledge distillation based NLP. Recently, knowledge distillation has been successfully applied into many tasks, such as model compression [Kim and Rush, 2016] and knowledge transfer [Zeng *et al.*, 2019; Tan *et al.*, 2019].

Similar to our first many-to-one knowledge distillation, Tan *et al.* [2019] presented a distillation-based approach to boost the performance of multilingual machine translation. Furlanello *et al.* [2018] proposed BAN that is a simple retraining procedure. During this procedure, after the teacher model coverges, they initialize a new student model and train it with the dual goals of predicting the correct labels and matching the output distribution of the teacher model. On this basis, Clark *et al.* [2019] extended BAN to BAM, which utilizes multiple task datasets simultaneously and thus can be considered as the multi-task version of BAN.

The most related work to ours is [Clark *et al.*, 2019]. However, our work significantly differs from this work in the following important aspects: (1) During the first step of knowledge distillation, Clark *et al.* [2019] applied knowledge distillation to transfer the knowledge of all single-task models to a unified multi-task model. By contrast, we distills all non-target single-domain models into a multi-domain model, which leverages domain-level similarities and thus is a domain-specific one, resulting in the effectiveness of the iterative execution of our framework. (2) During the second step of knowledge distillation, our teacher model is the current domain-specific model rather than the multi-task model adopted by [Clark *et al.*, 2019]; (3) We focus on MRC rather than natural language inference in [Clark *et al.*, 2019]. Note that experimental results reported in Table 1 verify the superiority of our framework over [Clark *et al.*, 2019].

3 BERTQA

In this section, we briefly introduce BERTQA [Devlin *et al.*, 2019] that is chosen as our basic MRC model. Figure 2 shows the architecture of BERTQA. Given a passage $p = p_1, p_2 ... p_{|p|}$ and a question $q = q_1, q_2 ... q_{|q|}$, we first pack them into a single sequence s=[<CLS>, q, <SEP>, p, <SEP>], where <CLS> is the token used for classification, <SEP> is the token separating q and p, and the sequence length $|s| = |p| + |q| + 3$. For each token s_t, we construct its input representation as $h_t^{(0)} = e_t^{tok} + e_t^{pos} + e_t^{seg}$, where e_t^{tok}, e_t^{pos}, and e_t^{seg} are the token, position, and segment embeddings for s_t, respectively. Please note that all tokens of p share a same segment embedding, and all tokens of q use a same segment embedding.

Such input representations are then fed into a BERT encoder with successive L Transformer [Vaswani *et al.*, 2017] layers, each of which is composed of two sub-layers with the layer normalization mechanism: a multi-head self-attention and a position-wise fully connected feed-forward neural network. Finally, on the top of the hidden states $h_t^{(L)}$ at the L-th layer, we use a linear layer with softmax functions to produce the probability of each token s_t to be the beginning or ending position of the answer span:

$$P_t^1 = \frac{\exp(\mathbf{W}_{s1}^\top h_t^{(L)})}{\sum_{t'=1}^{|s|} \exp(\mathbf{W}_{s1}^\top h_{t'}^{(L)})}, \quad (1)$$

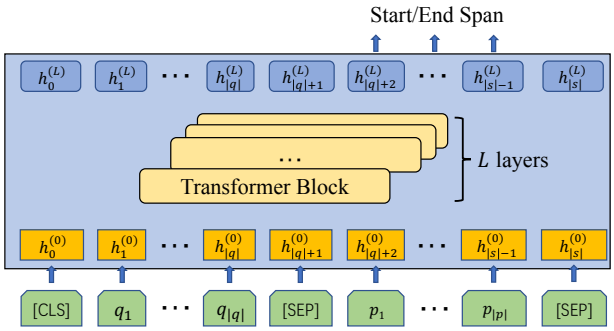

Figure 2: The BERTQA model.

$$P_t^2 = \frac{\exp(\mathbf{W}_{s2}^\top h_t^{(L)})}{\sum_{t'=1}^{|s|} \exp(\mathbf{W}_{s2}^\top h_{t'}^{(L)})}, \quad (2)$$

where $\mathbf{W}_{s*}$ are related parameters.

To train BERTQA, we define the training objective based on the log-likelihood of the true beginning and ending positions:

$$J = -\frac{1}{|D|} \sum_{qu \in D} (\log P_b^1(\theta) + \log P_e^2(\theta)), \quad (3)$$

where the **qu**adruple $qu = (p, q, b, e)$ denotes an example of the training data D, with b and e as the answer beginning and ending position, respectively, and θ is the set of model parameters. At inference time, the span (b,e) where $a \leq b$ with the maximum $P_b^1 \cdot P_e^2$ is chosen as the predicted answer.

4 Our Framework

In this section, we describe our framework in detail. As displayed in Figure 1, it is a multi-source multi-pass knowledge transfer process. To facilitate the understanding of our framework, we also summarize its training procedure in Algorithm 1. We first train all initial domain-specific MRC models $\{\theta_i^{(0)}\}_{i=1}^N$ using their own training data $\{D_i\}_{i=1}^N$, respectively (**Line** 1). Then, we calculate the domain-level similarities between training corpora, denoted by a matrix S=$\{s_{i,j}\}$, $1 \leq i, j \leq N$ (**Line** 2), which can be subsequently exploited to distinguish the impacts of other domains on the construction of each multi-domain MRC model. Afterwards, we conduct K iterations of many-to-many knowledge transfer (**Lines** 3-13). Concretely, at the k-th iteration, we synchronously perform multiple many-to-one knowledge transfers from other domains to each target-domain model $\theta_i^{(k+1)}$, $1 \leq j \leq N$ (**Lines** 4-12). Such a transfer mainly consists of two substeps: (1) We construct a multi-domain MRC model $\phi_i^{(k)}$ using $\{D_j\}_{j \neq i}$ and $\{\theta_j^{(k)}\}_{j \neq i}$ (**Lines** 5-6), and (2) We use $\phi_i^{(k)}$, D_i and $\theta_i^{(k)}$ to establish $\theta_i^{(k+1)}$ (**Lines** 7-11). Through this multi-pass procedure, different domain-specific MRC models can constantly absorb domain-shared knowledge to reinforce each other.

Clearly, the calculation of domain-level similarities S and these aforementioned two substeps are the most important steps in our framework. We describe them below.

Calculations of Domain-level Similarities S. To this end, for the training data D_i of each domain, we first collect its all questions to form a large question document Q_i, and then use the TF-IDF representation to represent the domain Q_i. Finally, we directly define the domain-level similarity $s_{i,j}$ between i-th and j-th domains as the cosine similarity based on their TF-IDF representations: $s_{i,j}$ = cosine(TF-IDF(Q_i),TF-IDF(Q_j)). Then, we normalize this similarity as $s_{i,j} = \frac{s_{i,j}}{\sum_{j' \neq i} s_{i,j'}}$.

Please note that we only use the question words to calculate the domain-level similarities. This is because the passages of different domains may be similar. For example, SQuAD and HotpotQA are both constructed from Wikipedia. By contrast, the question words are able to better reflect the domain-level differences.

Substep 1: Constructing Multi-domain MRC Model $\phi_i^{(k)}$. Inspired by [Tan *et al.*, 2019; Clark *et al.*, 2019], we employ knowledge distillation to construct multi-domain MRC model $\phi_i^{(k)}$, where we train $\phi_i^{(k)}$ to fit the training data $\{D_j\}_{j \neq i}$ and match the outputs of individual domain-specific models $\{\theta_j\}_{j \neq i}$ simultaneously according to their domain-level similarities to D_i.

Specifically, for the i-th domain, we mix training corpora $\{D_j\}_{j \neq i}$ from other domains to form a mixed training corpus M_i, where we conduct up-sampling on $\{D_j\}_{j \neq i}$ to make all domains have the same amount of training instances. For a sampled training example $qu = (p, q, b, e)$ from D_j, two kinds of errors are involved:

1. **Negative likelihood** $L_1(qu; \phi_i^{(k)})$: it is used to enable $\phi_i^{(k)}$ to fit training data D_j. Formally, it can be formulated as

$$L_1(qu; \phi_i^{(k)}) = -\log P_b^1(\phi_i^{(k)}) - \log P_e^2(\phi_i^{(k)}), \quad (4)$$

where $P_b^1(\phi_i^{(k)})$ and $P_e^2(\phi_i^{(k)})$ are defined in Equation 1 and 2, respectively.

2. **Prediction divergence** $L_2(qu; \theta_j^{(k)}; \phi_i^{(k)})$: it aims to keep the outputs of ϕ_i^k consistent with those of domain-specific MRC model $\theta_j^{(k)}$, and is formally defined as the mean squared error between outputs of $\theta_j^{(k)}$ and $\phi_i^{(k)}$:

$$\begin{aligned} L_2(qu; \theta_j^{(k)}; \phi_i^{(k)}) = &(P_b^1(\phi_i^{(k)}) - P_b^1(\theta_j^{(k)}))^2 \\ &+ (P_e^2(\phi_i^{(k)}) - P_e^1(\theta_j^{(k)}))^2. \end{aligned} \quad (5)$$

Thus, the joint error for the training instance qu is

$$\begin{aligned} L_{\text{md}}(qu; \theta_j^{(k)}; \phi_i^{(k)}) = &\lambda_{\text{md}} \cdot L_1(qu; \phi_i^{(k)}) \\ &+ (1 - \lambda_{\text{md}}) \cdot L_2(qu; \theta_j^{(k)}; \phi_i^{(k)}). \end{aligned} \quad (6)$$

Here we follow Clark et al. [2019] to linearly increase λ_{md} from 0 to 1 throughout training. In this way, $\phi_i^{(k)}$ will absorb the guidance information from its teachers models $\{\theta_j^{(k)}\}_{j \neq i}$ as much as possible during the early stage of training, and then rely more on the training data so it can learn to surpass its teachers.

Finally, the objective function over M_i becomes

$$J_{\text{md}} = \frac{1}{|M_i|} \sum_{qu \in M_i} s_{i,j} \cdot L_{\text{md}}(qu; \theta_j^{(k)}; \phi_i^{(k)}). \quad (7)$$

Algorithm 1 Iterative Multi-source Mutual Knowledge Transfer Framework for MRC

Input: Training corpora $\{D_i\}_{i=1}^N$, validation sets $\{D_i^v\}_{i=1}^N$, the maximal iteration number K
Output: Domain-specific MRC models $\{\theta_i^{(K)}\}_{i=1}^N$

$\theta_i^{(0)} \leftarrow$TrainModel$(D_i), i = 1, 2, ..., N$
Compute the similarity matrix S=$\{s_{i,j}\}, i, j = 1, 2, ..., N$, where $s_{i,j}$ indicates the domain-level similarity between D_i and D_j // **Step 1**
for $k = 0, 1, 2, ..., K-1$ **do**
  for $i = 1, 2, ..., N$ **do**
    Initialize the multi-domain MRC model $\phi_i^{(k)}$
    $\phi_i^{(k)} \leftarrow$ TrainMultiDomainModel$(\{D_j\}_{j \neq i}, \{\theta_j^{(k)}\}_{j \neq i}, S)$ //**Step 2**
    Initialize $\theta_i^{(k+1)}$ with $\phi_i^{(k)}$
    $\theta_i^{(k+1)} \leftarrow$ FineTuneModel$(D_i, \theta_i^{(k)})$ // **Step 3**
    if EvalModel$(\theta_i^{(k+1)}, D_i^v) <$ EvalModel$(\theta_i^{(k)}, D_i^v)$ **then**
      $\theta_i^{(k+1)} \leftarrow \theta_i^{(k)}$
    end if
  end for
end for

Where M_i is the previously-mentioned mixed training corpus. Note that by introducing $s_{i,j}$, we can effectively differentiate the impacts of different domains on $\phi_i^{(k)}$.

Substep 2: Constructing Domain-specific MRC Model $\theta_i^{(k+1)}$. During this process, we first initialize $\theta_i^{(k+1)}$ with $\phi_i^{(k)}$, and then also apply knowledge distillation [Tan *et al.*, 2019; Clark *et al.*, 2019] to update $\theta_i^{(k+1)}$ using its previous model $\theta_i^{(k)}$ and training corpus D_i.

Similarly, the objective function of each training example qu from D_i involves two kinds of errors:

1. **Negative likelihood** $L_3(qu; \theta_i^{(k+1)})$: it requires $\theta_i^{(k+1)}$ to fit the reference of D_i:

$$L_3(qu; \theta_i^{(k+1)}) = -\log P_b^1(\theta_i^{(k+1)}) - \log P_e^2(\theta_i^{(k+1)}). \quad (8)$$

2. **Prediction divergence** $L_4(qu; \theta_i^{(k+1)})$: it makes the outputs of $\theta_i^{(k+1)}$ consistent with those of $\theta_i^{(k)}$:

$$\begin{aligned} L_4(qu; \theta_i^{(k)}; \theta_i^{(k+1)}) = &(P_b^1(\theta_i^{(k+1)}) - P_b^1(\theta_i^{(k)}))^2 \\ &+ (P_e^2(\theta_i^{(k+1)}) - P_e^1(\theta_i^{(k)}))^2. \end{aligned} \quad (9)$$

Finally, the objective function over D_i is defined as

$$J_{\text{ds}} = \frac{1}{|D_i|} \sum_{qu \in D_i} L_{\text{ds}}(qu; \theta_i^{(k)}; \theta_i^{(k+1)}), \quad (10)$$

where

$$\begin{aligned} L_{\text{ds}}(qu; \theta_i^{(k)}; \theta_i^{(k+1)}) = &\lambda_{\text{ds}} \cdot L_3(qu; \theta_i^{(k+1)}) \\ &+ (1 - \lambda_{\text{ds}}) \cdot L_4(qu; \theta_i^{(k)}; \theta_i^{(k+1)}), \end{aligned} \quad (11)$$

where the hyper-parameter λ_{ds} is adjusted in a similar way as λ_{md}. In particular, we compare the performance of $\theta_i^{(k+1)}$ and $\theta_i^{(k)}$ to retain the current best model (**Lines** 9-11).

As described before, we apply the above procedure to generate all domain-specific models $\{\theta_i^{(k+1)}\}_{i=1}^N$ in a parallel manner, which can effectively reduce the training time of our framework. Furthermore, we repeat this many-to-many knowledge transfer process, until the maximal number of iteration K is reached or all domain-specific models converge.

5 Experiments

5.1 Setup

In our experiments, we use the following datasets:

- SQuAD [Rajpurkar *et al.*, 2016]. It is one of the most popular MRC datasets, which consists of more than 100K instances. Each example is a pair of context paragraph from Wikipedia and a question created by a human, and the answer is a span in the context. Particularly, we use SQuAD v1.0 in this work.
- NewsQA [Trischler *et al.*, 2017]. It contains about 120K question-answer pairs, all of which are related to CNN articles.
- TriviaQA [Joshi *et al.*, 2017]. This dataset includes 95K question-answer pairs from Trivia Database, each of which is paired with a document collected by completing a web search of the question.
- HotpotQA [Yang *et al.*, 2018]. Crowdsourcing workers were shown pairs of related Wikipedia paragraphs and asked to author questions that require multi-hop reasoning over the paragraphs. There are two versions of HotpotQA, and we use the second version, where 10 paragraphs retrieved by an information retrieval (IR) system are given.
- NQ [Kwiatkowski *et al.*, 2019]. It is a question answering dataset where the numbers of its training, validation and test examples are 307K, 7.8K and 7.8K, respectively. Each example is comprised of a google.com query and a corresponding Wikipedia page.

We adopt the F1 score as our evaluation metric. We sample 300 examples from the training data of each domains as our validation sets. Finally, we evaluate the performance of different models on the official validation sets.

To enhance the performance of our MRC model, we initialize the parameters of the BERT encoding layer using the pre-trained model officially released by Google*, and randomly initialize other trainable parameters. These models were pre-trained on the concatenation of BooksCorpus (800M words) and Wikipedia (2,500M words) via joint modeling of tasks of masked language model and next sentence prediction. We use the uncased base model, which is case insensitive and contains 12 Transformer encoding blocks, each with 12 self-attention heads and 768 hidden units. Moreover, we use the Adam optimizer [Kingma and Ba, 2015] with a learning rate of 1.5×10^{-5} and a batch size of 12.

*https://github.com/google-research/bert

Model	SAD	NQA	HQA	NQ	TQA	Ave.
Single	88.30	65.25	76.17	77.00	68.42	75.03
Mix	87.88	67.19	75.69	76.08	72.60	75.89
FT	**89.52**	66.87	77.83	77.92	72.74	76.98
MS-FT	88.83	65.97	76.34	77.28	70.14	75.71
MFT	88.56	67.65	76.35	77.70	73.46	76.74
MS-MFT	87.69	67.47	76.50	77.67	71.20	76.11
BAM	89.50	68.00	77.45	77.79	72.83	77.11
IMM	89.44	**68.78**	**78.52**	**79.23**	**73.71**	**77.94**

Table 1: Experimental results on various test sets. SAD = SQuAD, NQA = NewsQA, HQA = HotpotQA, and TriviaQA = TQA Ave. = average score.

5.2 Baseline Models

We refer to our framework as **IMM** and compare it with the following baseline models:

- **Single**. The BERTQA model trained on a single domain training corpus.
- **Mix**. The BERTQA model trained on the mix of all training corpora.
- **Fine-tuning (FT)**. We first train an MRC model on the mix of non-target domain training corpora until convergence, and then update its parameters on the target-domain training corpus.
- **Fine-tuning with the most similar training data (MS-FT)**. A variant of FT, where we only exploit a source-domain data that is the most similar to the target-domain data when we using fine-tuning.
- **Mixed fine-tuning (MFT)** [Chu *et al.*, 2017]. Unlike FT, we resume training the MRC model on a mix of all training data.
- **Mixed fine-tuning with the most similar training data (MS-MFT)**. A variant of MFT, where only the source-domain data most similar to the target-domain data is leveraged via mixed fine-tuning.
- **Born-Again Multi-task Network (BAM)**. It is the multi-task version of Born-Again Network (BAN) [Clark *et al.*, 2019]. Using the same many-to-one knowledge distillation procedure as BAM, we first construct a unified multi-domain model using all equally considered domain-specific models and training data, then further fine-train the domain-specific models on individual datasets.

Effect of the Maximal Iteration Number K

From Figure 1, we observe that the maximal iteration number K is an important hyper-parameter, which directly determines the amount of the transferred knowledge into domain-specific MRC models. To investigate the effect of K, we report the average performance of domain-specific MRC models using different Ks on the validation sets. Note that we only keep the best models at each iteration.

Figure 3 illustrates the experimental results using different Ks. At first, the performance of almost all domain-specific

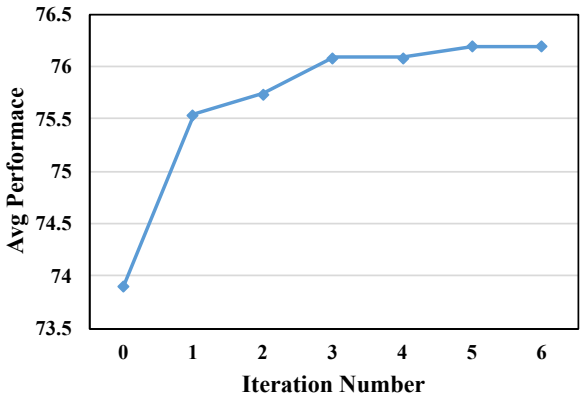

Figure 3: Effect of the iteration number (K) on the validation sets.

MRC models has been improved with the increment of the iteration number. However, when the iteration number is greater than 3, all models tend to converge. Therefore, we directly used K=3 in all subsequent experiments.

Overall Performance

Table 1 shows the experimental results. Overall, the conventional approaches including *FT*, *MS-FT*, *MFT*, *MS-MFT* and *BAM*, achieve better performance than both *Single* and *Mix*, echoing the reported results in previous studies [Min *et al.*, 2017; Chung *et al.*, 2018; Xu *et al.*, 2019]. Besides, under our framework, the performance of all domain-specific models is further significantly improved. Specifically, the average score of our framework is **77.94**, which is **+2.91**, **+2.05**, **+0.96**, **+2.23**, **+1.20**, **+1.83**, **+0.83** points higher than those of *Single*, *Mix*, *FT*, *MS-FT*, *MFT*, *MS-MFT* and *BAM*, respectively. In particular, in all datasets except SQuAD, our model achieves the best performance among all models. Even compared with BAM, our improvement is still remarkable. This result strongly demonstrates the effectiveness and generality of our framework.

Robustness in MRC

To inspect the impact of our framework on enhancing robustness, we follow Hu *et al.* [2018] to report the results on two adversarial SQuAD datasets, namely AddOneSent and AddSent.

From Table 2, we can observe that the improvement on adversarial data is much higher than the one on original squad dataset. This result is reasonable, because our framework can better absorb the knowledge of other domains, boosting its ability in dealing with the much more confusing answers in adversarial datasets.

5.3 Ablation Study

The main highlights of our framework consist of the usages of domain-level similarities of training corpora to the target-domain one, and two stages of distillation-based knowledge transfers. To evaluate their effects on our framework, we compared our framework with its following variants: (1) **-SIMI**. The variant of IMM without domain-level similarities, i.e., we directly set $s_{i,j}$ of Equation 7 to 1; (2) **-KD1**. The variant of IMM without the distillation-based knowledge transfer from $\{\theta_j^{(k)}\}_{j\neq i}$ to $\phi_i^{(k)}$, where λ_{md} of Equation 6 is directly set to 0; and (3) **-KD2**. The variant of IMM without the distillation-based knowledge transfer from $\phi_i^{(k)}$ to $\theta_i^{(k+1)}$, where λ_{ds} of Equation 11 is directly set to 0.

Model	SQuAD	AddSent	AddOneSent
Single	88.30	50.15	61.38
Mix	87.88	53.15	64.04
FT	**89.52**	53.05	63.91
MS-FT	88.83	50.88	62.78
MFT	88.89	53.29	64.23
MS-MFT	87.69	51.03	62.94
BAM	89.50	55.52	65.20
IMM	89.44	**56.49**	**66.20**

Table 2: Comparison of different approaches on two adversarial SQuAD datasets.

Model	Ave.
IMM	**77.94**
−SIMI	77.49
−KD1	77.68
−KD2	77.58
−KD1, KD2	77.34

Table 3: Ablation study of our framework on various test sets.

From the experimental results shown in Table 3, we can observe that our framework achieves better performance than all its variants. These results show that domain-level similarities, two stages of distillation-based knowledge transfers are indeed beneficial for MRC.

6 Conclusion

In this work, we have presented a novel iterative multi-source mutual knowledge transfer framework for MRC, which enables all domain-specific MRC models to constantly reinforce each other by iteratively and differentially absorbing the domain-shared knowledge from others. Experimental results and in-depth analyses on several MRC datasets strongly demonstrate the effectiveness of our framework.

In the future, we will focus on how to effectively leverage the unlabeled data of different domains under our framework. Besides, we plan to apply our framework to other tasks, such as machine translation [Zeng *et al.*, 2019]. Finally, inspired by studies of multi-domain NMT [Zeng *et al.*, 2018; Su *et al.*, 2019], we will explore a unified multi-domain model for MRC.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61672440), Beijing Advanced Innovation Center for Language Resources (No. TYR17002), and the Scientific Research Project of National Language Committee of China (No. YB135-49). We also thank the reviewers for their insightful comments.

References

[Chu *et al.*, 2017] Chenhui Chu, Raj Dabre, and Sadao Kurohashi. An empirical comparison of domain adaptation methods for neural machine translation. In *ACL*, 2017.

[Chung *et al.*, 2018] Yu-An Chung, Hung-yi Lee, and James R. Glass. Supervised and unsupervised transfer learning for question answering. In *NAACL-HLT*, 2018.

[Clark *et al.*, 2019] Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. Bam! born-again multi-task networks for natural language understanding. In *ACL*, 2019.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[Duan *et al.*, 2017] Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. Question generation for question answering. In *EMNLP*, 2017.

[Furlanello *et al.*, 2018] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In *ICML*, 2018.

[Golub *et al.*, 2017] David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. Two-stage synthesis networks for transfer learning in machine comprehension. In *EMNLP*, 2017.

[Hinton *et al.*, 2015] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[Hu *et al.*, 2018] Minghao Hu, Yuxing Peng, Furu Wei, Zhen Huang, Dongsheng Li, Nan Yang, and Ming Zhou. Attention-guided answer distillation for machine reading comprehension. In *EMNLP*, 2018.

[Joshi *et al.*, 2017] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, 2017.

[Kim and Rush, 2016] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *EMNLP*, 2016.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Kwiatkowski *et al.*, 2019] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *TACL*, 2019.

[Liu *et al.*, 2020] Kai Liu, Xin Liu, An Yang, Jing Liu, Jinsong Su, Sujian Li, and Qiaoqiao She. A robust adversarial training approach to machine reading comprehension. In *AAAI*, 2020.

[Min *et al.*, 2017] Sewon Min, Min Joon Seo, and Hannaneh Hajishirzi. Question answering through transfer learning from large fine-grained supervision data. In *ACL*, 2017.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[Rajpurkar *et al.*, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[Seo *et al.*, 2017] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *ICLR*, 2017.

[Song *et al.*, 2018] Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. Leveraging context information for natural question generation. In *NAACL-HLT (Short Papers)*, 2018.

[Su *et al.*, 2019] Jinsong Su, Jiali Zeng, Jun Xie, Huating Wen, Yongjing Yin, and Yang Liu. Exploring discriminative word-level domain contexts for multi-domain neural machine translation. *TPAMI*, 2019.

[Talmor and Berant, 2019] Alon Talmor and Jonathan Berant. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. In *ACL*, 2019.

[Tan *et al.*, 2019] Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Multilingual neural machine translation with knowledge distillation. In *ICLR*, 2019.

[Trischler *et al.*, 2017] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *ACL*, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[Xu *et al.*, 2019] Yichong Xu, Xiaodong Liu, Yelong Shen, Jingjing Liu, and Jianfeng Gao. Multi-task learning with sample re-weighting for machine reading comprehension. In *NAACL-HLT*, 2019.

[Yang *et al.*, 2018] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, 2018.

[Yu *et al.*, 2018] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*, 2018.

[Zeng *et al.*, 2018] Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. Multi-domain neural machine translation with word-level domain context discrimination. In *EMNLP*, 2018.

[Zeng *et al.*, 2019] Jiali Zeng, Yang Liu, jinsong su, yubing Ge, Yaojie Lu, Yongjing Yin, and jiebo luo. Iterative dual domain adaptation for neural machine translation. In *EMNLP*, 2019.