Adversarial Domain Adaptation for Machine Reading Comprehension

Huazheng Wang\textsuperscript{1*}, Zhe Gan\textsuperscript{2}, Xiaodong Liu\textsuperscript{3}, Jingjing Liu\textsuperscript{2}, Jianfeng Gao\textsuperscript{3}, Hongning Wang\textsuperscript{1}

\textsuperscript{1}University of Virginia, \textsuperscript{2}Microsoft Dynamics 365 AI Research, \textsuperscript{3}Microsoft Research
\{hw7ww,hw5x\}@virginia.edu, \{zhe.gan,xiaodl,jingjl,jfgao\}@microsoft.com

Abstract

In this paper, we focus on unsupervised domain adaptation for Machine Reading Comprehension (MRC), where the source domain has a large amount of labeled data, while only unlabeled passages are available in the target domain. To this end, we propose an Adversarial Domain Adaptation framework (AdaMRC), where (i) pseudo questions are first generated for unlabeled passages in the target domain, and then (ii) a domain classifier is incorporated into an MRC model to predict which domain a given passage-question pair comes from. The classifier and the passage-question encoder are jointly trained using adversarial learning to enforce domain-invariant representation learning. Comprehensive evaluations demonstrate that our approach (i) is generalizable to different MRC models and datasets, (ii) can be combined with pre-trained large-scale language models (such as ELMo and BERT), and (iii) can be extended to semi-supervised learning.

1 Introduction

Recently, many neural network models have been developed for Machine Reading Comprehension (MRC), with performance comparable to human in specific settings (Gao et al., 2019). However, most state-of-the-art models (Seo et al., 2017; Liu et al., 2018; Yu et al., 2018) rely on large amount of human-annotated in-domain data to achieve the desired performance. Although there exists a number of large-scale MRC datasets (Rajpurkar et al., 2016; Trischler et al., 2016; Bajaj et al., 2016; Zhang et al., 2018), collecting such high-quality datasets is expensive and time-consuming, which hinders real-world applications for domain-specific MRC. Therefore, the ability to transfer an MRC model trained in a high-resource domain to other low-resource domains is critical for scalable MRC. While it is difficult to collect annotated question-answer pairs in a new domain, it is generally feasible to obtain a large amount of unlabeled text in a given domain. In this work, we focus on adapting an MRC model trained in a source domain to other new domains, where only unlabeled passages are available.

This domain adaptation issue has been a main challenge in MRC research, and the only existing work that investigated this was the two-stage synthesis network (SynNet) proposed in Golub et al. (2017). Specifically, SynNet first generates pseudo question-answer pairs in the target domain, and then uses the generated data as augmentation to fine-tune a pre-trained MRC model. However, the source-domain labeled data and target-domain pseudo data are directly combined without considering domain differences (see Figure 1(a), where the two feature distributions in two domains are independently clustered). Directly transferring a model from one domain to another could be counter-effective, or even hurt the performance of the pre-trained model due to domain variance.

To achieve effective domain transfer, we need to learn features that are discriminative for the MRC task in the source domain, while simultaneously indiscriminating with respect to the shift between source and target domains. Motivated by this, we propose Adversarial Domain Adaptation for MRC (AdaMRC), a new approach that utilizes adversarial learning to learn domain-invariant transferable representations for better MRC model adaptation across domains (see Figure 1(b), where the two feature distributions learned by AdaMRC are indistinguishable through adversarial learning).

Specifically, our proposed method first generates synthetic question-answer pairs given pas-
sages in the target domain. Different from Golub et al. (2017), which only used pseudo question-answer pairs to fine-tune pre-trained MRC models, our AdaMRC model uses the passage and the generated pseudo-questions in the target domain, in addition to the human-annotated passage-question pairs in the source domain, to train an additional domain classifier as a discriminator. The passage-question encoder and the domain classifier are jointly trained via adversarial learning. In this way, the encoder is enforced to learn domain-invariant representations, which are beneficial for transferring knowledge learned from one domain to another. Based on this, an answer decoder is then used to decode domain-invariant representation into an answer span.

The proposed approach is validated on a set of popular benchmarks, including SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2016), and MS MARCO (Bajaj et al., 2016), using state-of-the-art MRC models including SAN (Liu et al., 2018) and BiDAF (Seo et al., 2017). Since pre-trained large-scale language models, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), have shown strong performance to learn representations that are generalizable to various tasks, in this work, to further demonstrate the versatility of the proposed model, we perform additional experiments to demonstrate that AdaMRC can also be combined with ELMo and BERT to further boost the performance.

The main contributions of this paper are summarized as follows: (i) We propose AdaMRC, an adversarial domain adaptation framework that is specifically designed for MRC. (ii) We perform comprehensive evaluations on several benchmarks, demonstrating that the proposed method is generalizable to different MRC models and diverse datasets. (iii) We demonstrate that AdaMRC is also compatible with ELMo and BERT. (iv) We further extend the proposed framework to semi-supervised learning, showing that AdaMRC can also be applied to boost the performance of a pre-trained MRC model when a small amount of labeled data is available in the target domain.

2 Related Work

Machine Reading Comprehension The MRC task has recently attracted a lot of attention in the community. An MRC system is required to answer a question by extracting a text snippet within a given passage as the answer. A large number of deep learning models have been proposed to tackle this task (Seo et al., 2017; Xiong et al., 2017; Shen et al., 2017; Liu et al., 2018; Yu et al., 2018). However, the success of these methods largely relies on large-scale human-annotated datasets (such as SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2016) and MS MARCO (Bajaj et al., 2016)).

Different from previous work that focused on improving the state of the art on particular MRC datasets, we study the MRC task from a different angle, and aim at addressing a critical yet challenging problem: how to transfer an MRC model learned from a high-resource domain to other low-resource domains in an unsupervised manner.

Although important for the MRC task, where annotated data are limited in real-life applications, this problem has not yet been well investigated. There were some relevant studies along this line. For example, Chung et al. (2018) adapted a pre-trained model to TOEFL and MCTest dataset, and Wiese et al. (2017) applied transfer learning to the biomedical domain. However, both studies assumed that annotated data in the target domain (either questions or question-answer pairs) are available.

To the best of our knowledge, SynNet (Golub et al., 2017) is the only work that also studied domain adaptation for MRC. Compared with SynNet, the key difference in our model is adversarial learning, which enables domain-invariant representation learning for better model adaptation to low-resource domains. Our approach is also related to multi-task learning (Xu et al., 2019; Caruana, 1997; Liu et al., 2015, 2019) and semi-supervised learning (Yang et al., 2017) for MRC.
In this work, we focus on purely unsupervised domain adaptation.

**Domain Adaptation** Domain adaptation aims to make a machine learning model generalizable to other domains, especially without any annotated data in the target domain (or with only limited data) (Ganin and Lempitsky, 2015). One line of research on domain adaptation focuses on transferring the feature distribution from the source domain to the target domain (Gong et al., 2012; Long et al., 2015). Another school of research focuses on learning domain-invariant representations (Glorot et al., 2011) (e.g., via adversarial learning (Ganin et al., 2016; Tzeng et al., 2017)).

Domain adaptation has been successfully applied to many tasks, such as image classification (Tzeng et al., 2017), speech recognition (Doulaty et al., 2015), sentiment classification (Ganin et al., 2016; Li et al., 2017), machine translation (Johnson et al., 2017; Zoph et al., 2016), relation extraction (Fu et al., 2017), and paraphrase identification (Shah et al., 2018). Compared to these areas, the application to MRC presents additional challenges, since besides missing labeled data (i.e., answer spans), the questions in the target domain are also unavailable. To our best knowledge, we are the first to investigate the usage of adversarial domain adaptation for the MRC task.

There are many prevailing unsupervised techniques for domain adaptation. Our proposed approach is inspired by the seminal work of Ganin et al. (2016) to validate its potential of solving domain adaptation problem on a new task, without any supervision for the target domain. There are also other more advanced methods, such as MMD-based adaptation (Long et al., 2017), residual transfer network (Long et al., 2016), and maximum classifier discrepancy (Saito et al., 2018) that can be explored for future work.

## 3 Problem Definition

The problem of unsupervised domain adaptation for MRC is defined as follows. First, let $S = \{p^s, q^s, a^s\}$ denote a labeled MRC dataset from the source domain $s$, where $p^s$, $q^s$ and $a^s$ represent the passage, the question, and the answer of a sample, respectively. An MRC model $M^s$, taking as input the passage $p^s = (p_1, p_2, ..., p_T)$ of length $T$ and the question $q^s = (q_1, q_2, ..., q_{T'})$ of length $T'$, is trained to predict the correct answer span $a^s = (a^s_{\text{start}}, a^s_{\text{end}})$, where $a^s_{\text{start}}, a^s_{\text{end}}$ represent the starting and ending indexes of the answer in the passage $p^s$.

We assume that only unlabeled passages are available in the target domain $t$, i.e., $T = \{p^t\}$, where $p^t$ represents a passage. This is a reasonable assumption as it is easy to collect a large amount of unlabeled passages in a new domain. Given datasets $S$ and $T$, the goal of unsupervised domain adaptation is defined as learning an MRC model $M^t$ based on $S$ and $T$ to answer questions in the target domain $t$.

## 4 AdaMRC

As illustrated in Figure 2, AdaMRC consists of three main components: (i) *Question Generator* (Sec. 4.1), where pseudo question-answer pairs are generated given unlabeled passages in the target domain; (ii) *MRC Module* (Sec. 4.2), where
given an input document and a question, an answer span is extracted from the document; (iii) Domain Classifier (Sec. 4.3), where a domain label is predicted to distinguish a feature vector from either the source domain or the target domain.

Specifically, the MRC module is composed of an encoder and a decoder. The encoder with parameter $\theta_{e}$ embeds the input passage and the question into a feature vector. The decoder with parameter $\theta_{d}$ takes the feature vector as input to predict the answer span. The domain classifier with parameter $\theta_{c}$ takes the same feature vector as input to classify the domain label. All the parameters $(\theta_{e}, \theta_{d}, \theta_{c})$ are jointly optimized, with the objective of training the encoder to correctly predict the answer span, but also simultaneously fool the domain classifier. In other words, the encoder learns to map text input into a feature space that is invariant to the switch of domains. The following sub-sections describe each module, with training details provided in Sec. 4.4.

4.1 Question Generation

First, we use an NER system to extract possible answer spans $a^{i}$ from the passages $p^{i}$ in the target domain, under the assumption that any named entity could be the potential answer of certain questions. Similar answer extraction strategy has been applied in Yang et al. (2017) in a semi-supervised learning setting, while Golub et al. (2017) proposed to train an answer synthesis network to predict possible answers spans. We tried both methods, and empirically observed that a simple NER system provides more robust results, which is used in our experiments.

Now, we describe how the question generation (QG) model is trained. Given the passage $p^{s} = (p_{1}, p_{2}, ..., p_{T})$ and answer $a^{s} = (a_{\text{start}}, a_{\text{end}})$ from the source domain, the QG model with parameter $\theta_{QG}$ learns the conditional probability of generating a question $q^{s} = (q_{1}, q_{2}, ..., q_{T'}$, i.e., $P(q^{s}| p^{s}, a^{s})$. We implement the QG model as a sequence-to-sequence model with attention mechanism (Bahdanau et al., 2015), and also apply the copy mechanism proposed in Gu et al. (2016); Gulcehre et al. (2016) to handle rare/unknown words.

Specifically, the QG model consists of a lexicon encoding layer, a BiLSTM contextual encoding layer, and an LSTM decoder. For lexicon encoding, each word token $p_{i}$ of a passage is mapped into a concatenation of GloVe vectors (Pennington et al., 2014), part-of-speech (POS) tagging embedding, and named-entity-recognition (NER) embedding. We further insert answer information by appending an additional zero/one feature (similar to Yang et al. (2017)) to model the appearance of answer tokens in the passage. The output of the lexicon encoding layer is appended with CoVe vectors (McCann et al., 2017), and then passed to the Bidirectional LSTM contextual encoding layer, producing a sequence of hidden states. The decoder is another LSTM with attention and copy mechanism over the encoder hidden states. At each time step, the generation probability of a question token $q_{t}$ is defined as:

$$P(q_{t}) = g_{t}P^{v}(q_{t}) + (1 - g_{t})P^{copy}(q_{t}),$$

where $g_{t}$ is the probability of generating a token from the vocabulary, while $(1 - g_{t})$ is the probability of copying a token from the passage. $P^{v}(q_{t})$ and $P^{copy}(q_{t})$ are defined as softmax functions over the words in the vocabulary and over the words in the passage, respectively. $g_{t}$, $P^{v}(q_{t})$ and $P^{copy}(q_{t})$ are functions of the current decoder hidden state.

4.2 MRC Module

Encoder The encoder in the MRC module contains lexicon encoding and contextual encoding, similar to the encoder used in the question generation module. It also includes a cross-attention layer for fusion. Specifically, the output of the lexicon encoder is appended with the CoVe vector and passed to the contextual encoding layer, which is a 2-layer BiLSTM that produces hidden states of the passage $H^{p} \in \mathbb{R}^{T \times 2m}$ and the question $H^{q} \in \mathbb{R}^{T' \times 2m}$, where $m$ is the hidden size of the BiLSTM. We then use cross attention to fuse $H^{p}$ and $H^{q}$, and construct a working memory of passage $M^{p} \in \mathbb{R}^{T \times 2m}$ (see Liu et al. (2018) for more details). The question memory $M^{q} \in \mathbb{R}^{2m}$ is constructed by applying self-attention on $H^{q}$.

Decoder The decoder, or answer module, predicts an answer span $a = (a_{\text{start}}, a_{\text{end}})$ given a passage $p$ and a question $q$, by modeling the conditional probability $P(a|p, q)$. The initial state $s_{0}$ is set as $M^{q}$. Through $T'$ steps, a GRU (Cho et al., 2014) is used to generate a sequence of state vectors $s_{t} = \text{GRU}(s_{t-1}, x_{t})$, where $x_{t}$ is computed via attention between $M^{p}$ and $s_{t-1}$. Two softmax layers are used to compute the distribution of
A self-attention layer is also applied to different domains, and predicts the domain label $d$ given the start and the end of the answer span at each step. Given the unlabeled dataset in the target domain, we use a sigmoid function to predict the domain label.

4.3 Domain Classifier

The domain classifier takes the output of the encoder as input, including the aforementioned passage representation $M_p^d \in \mathbb{R}^{T \times 2m}$ and the self-attended question representation $M_q^d \in \mathbb{R}^{2m}$ from different domains, and predicts the domain label $d$ by modeling the conditional probability $P(d|p, q)$. A self-attention layer is also applied to $M_p^d$ to reduce its size to $M_p^q \in \mathbb{R}^{2m}$. We then concatenate it with $M_q^d$, followed by a two-layer Multi-Layer Perceptron (MLP), $f(W[M_p^q; M_q])$, and use a sigmoid function to predict the domain label.

4.4 Training

Algorithm 1 illustrates the training procedure of our proposed framework. We first train the question generation model $\theta_{QG}$ on the source domain dataset $S$ by maximizing the likelihood of generating question $q^s$ given passage $p^s$ and answer $a^s$. Given the unlabeled dataset in the target domain, we extract candidate answers $a^t$ on $p^t$ and use $\theta_{QG}$ to generate pseudo questions $q^t$, and then compose a pseudo labeled dataset $T_{gen} = \{p^t, q^t, a^t\}$.

We initialize the MRC model $\theta$ for the target domain with the pre-trained MRC model $\theta^s$ from the source domain, and then fine-tune the model using both the source domain dataset $S$ and the target domain dataset $T_{gen}$. The goal of the decoder $\theta_d$ is to predict $P(a|p, q)$. The objective function is denoted as:

$$L_D(\theta_e, \theta_d) = \frac{1}{|S|} \sum_{i=1}^{|S|} \log P(a^{(i)}|p^{(i)}, q^{(i)})$$

(2)

where the superscript $(i)$ indicates the $i$-th sample. It is worthwhile to emphasize that unlike Golub et al. (2017), we only use source domain data to update the decoder, without using pseudo target domain data. This is because the synthetic question-answer pairs could be noisy, and directly using such data for decoder training may lead to degraded performance of the answer module, as observed both in Sachan and Xing (2018) and in our experiments.

The synthetic target domain data and source domain data are both used to update the encoder $\theta_e$ and the domain classifier $\theta_c$. The classifier predicts a domain label $d$ given the encoder $\theta_e$. The objective function is:

$$L_C(\theta_e, \theta_c) = \frac{1}{N} \sum_{i=1}^N \log P(d^{(i)}|p^{(i)}, q^{(i)})$$

(3)

where $N = |S| + |T_{gen}|$. In order to learn domain-invariant representations from the encoder, we update $\theta_c$ to maximize the loss while updating $\theta_e$ to minimize the loss in an adversarial fashion. The overall objective function is defined as:

$$L(\theta_e, \theta_d, \theta_c) = L_D(\theta_e, \theta_d) - \lambda L_C(\theta_e, \theta_c)$$

(4)

where $\lambda$ is a trade-off parameter that balances the two terms.

To optimize our model, instead of alternately updating the adversaries like in GAN (Goodfellow et al., 2014), we use the gradient-reversal layer (Ganin and Lempitsky, 2015) to jointly optimize all the components, as suggested in Chen et al. (2018).

5 Experiments

5.1 Experimental Setting

Datasets We validate our proposed method on three benchmarks: SQuAD (Rajpurkar et al.,
Dataset | Domain | Train | Dev | Test
--- | --- | --- | --- | ---
SQuAD (v1.1) | Wiki | 87,600 | 10,570 | −
NewsQA | News | 92,549 | 5,166 | 5,165
MS MARCO (v1) | Web | 82,430 | 10,047 | 9,650

Table 1: Statistics of the datasets.

Evaluation metrics
For SQuAD and NewsQA, we report results on two evaluation metrics: Exact Match (EM), which measures the percentage of span predictions that match any of the ground truth answers exactly; and Macro-averaged F1 score, which measures the average overlap between the prediction and the ground-truth answer. For MS MARCO, since the answer is free-formed, we use BLEU and ROUGE-L scores for evaluation.

Implementation details
1. We use spaCy\(^2\) to generate POS and NER taggings, which are used in answer extraction and the lexicon encoding layer of the QG and MRC models. The QG model is fixed after trained on source-domain labeled data. The hidden size of the LSTM in the QG model is set to 125. Parameters of the SAN model follow Liu et al. (2018). The hidden size of the MLP layer in the domain classifier is set to 125. Both the QG and the MRC model are optimized via Adamax (Kingma and Ba, 2015) with mini-batch size set to 32. The learning rate is set to 0.002 and is halved every 10 epochs. To avoid overfitting, we set the dropout rate to 0.3. For each mini-batch, data are sampled from both domains, with \(k_s\) samples from the source domain and \(k_t\) samples from the target domain. We set \(k_s : k_t = 2 : 1\) as default in our experiments. For the trade-off parameter \(\lambda\), we gradually change it from 0 to 1, following the schedule suggested in Ganin and Lempitsky (2015).

5.2 Experimental Results
We implement the following baselines and models for comparison.

1. SAN: we directly apply the pre-trained SAN model from the source domain to answer questions in the target domain.

Table 2 summarizes the experimental results. We observe that the proposed method consistently outperforms SAN and the SynNet+SAN model on all datasets. For example, in the SQuAD\(\rightarrow\)NewsQA setting, where the source-domain dataset is SQuAD and the target-domain dataset is NewsQA, AdaMRC achieves 38.46% and 54.20% in terms of EM and F1 scores, outperforming the pre-trained SAN by 1.78% (EM) and 1.41% (F1), respectively, as well as surpassing SynNet by 3.27% (EM) and 4.59% (F1), respectively. Similar improvements are also observed in NewsQA\(\rightarrow\)SQuAD, SQuAD\(\rightarrow\)MS MARCO and MS MARCO\(\rightarrow\)SQuAD settings, which demonstrates the effectiveness of the proposed model.

Interestingly, we find that the improvement on adaptation between SQuAD and NewsQA is greater than that between SQuAD and MS MARCO. Our assumption is that it is because

1\(^\text{Code will be released for easy access.}\)

2\(\text{https://spacy.io/}\)
SQuAD and NewsQA datasets are more similar than SQuAD and MS MARCO, in terms of question style. For example, questions in MS MARCO are real web search queries, which are short and may have typos or abbreviations; while questions in SQuAD and NewsQA are more formal and well written. Furthermore, the ground-truth answers in MS MARCO are human-synthesized and usually much longer (16.4 tokens in average) than those in the other datasets, while our answer extraction process focuses on named entities (which are much shorter). We argue that extracting named entities as possible answers is still reasonable for most of the reading comprehension tasks such as SQuAD and NewsQA. The problem of synthesizing answers across different domains will be investigated in future work.

SynNet vs. pre-trained SAN baseline  One observation is that SynNet performs worse than the pre-trained SAN baseline. We hypothesize that this is because the generated question-answer pairs are often noisy and inaccurate, and directly fine-tuning the answer module with synthetic data may hurt the performance, which is also observed in Sachan and Xing (2018), especially when a well-performed MRC model is used as the baseline. Note that we do observe improvements from SynNet+BiDAF over the pre-trained BiDAF model, which will be discussed in Sec. 6.2.

Comparing with upper-bound  The “AdaMRC with GT questions” model (in Section 5.2) serves as the upper-bound of our proposed approach, where ground-truth questions are used instead of synthesized questions. By using ground-truth questions, performance is further boosted by around 1%. This suggests that our question generation model is effective as the margin is relatively small, yet it could be further improved. We plan to study if recent question generation methods (Du et al., 2017; Duan et al., 2017; Sun et al., 2018; Benmalek et al., 2019) could further help to close the performance gap in future work.

6 Analysis

6.1 Visualization

To demonstrate the effectiveness of adversarial domain adaptation, we visualize the encoded representation via t-SNE (Maaten and Hinton, 2008) in Figure 1. We observe that with AdaMRC, the two distributions of encoded feature representa-
Method                | EM    | F1    | Ratio   | SAN     | AdaMRC + SAN    
----------------------|-------|-------|---------|---------|----------------|
SAN                   | 36.68 | 52.79 | 0%      | 36.68   | 38.46/54.20    
AdaMRC + SAN          | 38.46 | 54.20 | 5%      | 47.61   | 48.50/63.17    
SAN + ELMo            | 39.61 | 55.18 | 10%     | 48.66   | 49.64/63.94    
AdaMRC + SAN + ELMo   | 40.96 | 56.25 | 20%     | 50.75   | 51.14/65.38    
BERT_BASE             | 42.00 | 58.71 | 50%     | 53.24   | 53.34/67.30    
AdaMRC + BERT_BASE    | 42.59 | 59.25 | 100%    | 56.48   | 56.29/68.97    

Table 4: Results of using ELMo and BERT. Setting: adaptation from SQuAD to NewsQA.

**Results on DuoRC** We further test our model on the newly-released DuoRC dataset (Saha et al., 2018). This dataset contains two subsets: movie descriptions collected from Wikipedia (SelfRC) and from IMDB (ParaphraseRC). Although the two subsets are describing the same movies, the documents from Wikipedia are usually shorter (580 words in average), while the documents from IMDB are longer and more descriptive (926 words in average). We consider them as two different domains and perform domain adaptation from Wikipedia to IMDB. This experiment broadens our definition of domain.

In the DuoRC dataset, the same questions are asked on both Wikipedia and IMDB documents. Thus, question synthesis is not needed, and comparison with SynNet is not feasible. Note that the answers of the same question could be different in the two subsets (only 40.7% of the questions have the same answers in both domains). We preprocess the dataset and test the answer-span extraction task following Saha et al. (2018). Results are reported in Table 3. AdaMRC improves the performance over both SAN (1.26%, 1.54% in EM and F1) and BiDAF (1.27%, 2.02% in EM and F1). This experiment validates that our method can be applied to different styles of domain adaptation tasks as well.

### 6.3 AdaMRC with Pre-trained Language Models

To verify that our approach is compatible with large-scale pre-trained language models, we evaluate our model with ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). To apply ELMo to SAN, we use the model provided by AllenNLP⁴, and append a 1024-dim ELMo vector to the contextual encoding layer, with dropout rate set to 0.5. For BERT, we experiment with the pre-trained BERT_BASE uncased model⁵ due to limited computational resources. We use the original design of finetuning BERT for the MRC task in Devlin et al. (2019), instead of combining BERT with SAN. Results are provided in Table 4. We observe that using ELMo and BERT improves both AdaMRC and the baseline model. However, the improvement over ELMo and BERT is relatively smaller than SAN. We believe this is because pre-trained language model provides additional domain-invariant information learned from external data, and therefore limits the improvement of domain-invariant feature learning in our model. However, it is worth noting that combining AdaMRC with BERT achieves the best performance, which validates that AdaMRC is compatible with data augmentation from external sources.

### 6.4 Semi-supervised Setting

As an additional experiment, we also evaluate the proposed AdaMRC framework for semi-supervised domain adaptation. We randomly sample $k$ portion of labeled data from the target domain, and feed them to the MRC model. The ratio of labeled data ranges $k$ from 0% to 100%. Table 5 shows that AdaMRC outperforms SAN. However, the gap is decreasing when the labeling ratio increases. When the ratio is 20% or smaller, there is noticeable improvement. When the ratio is set to 50%, the two methods result in similar performance. When the ratio is increased to 100%, i.e., fully supervised learning, the performance of AdaMRC is slightly worse than SAN. This is possibly because in a supervised learning setting, the encoder is trained to preserve domain-specific feature information. The overall results suggest that our proposed AdaMRC is also effective in semi-supervised setting, when a small portion of target-domain data is provided.

⁴https://allennlp.org/

⁵https://github.com/google-research/bert
Refugee camps in eastern Chad house about 300,000 people who fled violence in the Darfur region of Sudan. The U.N. High Commissioner for Refugees said on Monday that more than 12,000 people have fled militia attacks over the last few days from Sudan’s Darfur region to neighboring Chad...

Answer: 12,000

GT Question: How many have recently crossed to Chad?

Pseudo Question: How many people fled the Refugee region to Sudan?

Sources say the classified materials were taken from the East Tennessee Technology Park. Roy Lynn Oakley, 67, of Roane County, Tennessee, appeared in federal court in Knoxville on Thursday. Oakley was briefly detained for questioning in the case in January...

Answer: Roy Lynn Oakley

GT Question: Who is appearing in court ?

Pseudo Question: What is the name of the classified employee in Tennessee on East Tennessee?

The Kyrgyz order became effective on Friday when President Kurmanbek Bakiyev reportedly signed legislation that the parliament in Bishkek backed on Thursday, the Pentagon said. Pentagon spokesman Bryan Whitman said the Kyrgyz Foreign Ministry on Friday officially notified the U.S. Embassy in Bishkek that a 180-day withdrawal process is under way...

Answer: President Kurmanbek Bakiyev

GT Question: Who is the President of Kyrgyzstan ?

Pseudo Question: What is the name of the classified employee in Tennessee on East Tennessee?

A high court in northern India on Friday acquitted a wealthy businessman facing the death sentence for the killing of a teen in a case dubbed “the house of horrors.” Moninder Singh Pandher was sentenced to death by a lower court in February. The teen was one of 19 victims – children and young women – in one of the most gruesome serial killings in India in recent years...

Answer: one of 19

GT Question: What was the amount of children murdered?

Pseudo Question: How many victims were in India?

Table 6: Examples of generated questions given input paragraphs and answers, comparing with the ground-truth human-written questions.

6.5 Examples of Generated Questions

The percentage of generated questions starting with “what”, “who”, “when” and “where” are 63.2%, 12.8%, 2.3%, and 2.1%, respectively. We provide several examples of generated questions in Table 6. We observe that the generated questions are longer than human-written questions. This is possibly due to the copy mechanism used in the question generation model, which enables directly copying words into the generated questions. On the one hand, the copy mechanism provides detailed background information for generating a question. However, if not copying correctly, the question could be syntactically incorrect. For instance, in the third example, “signed legislation that the parliament” is copied from the passage. The copied phrase is indeed describing the answer “President Kurmanbek Bakiyev”; however, the question is syntactically incorrect and the question generator should copy “the parliament backed on Thursday” instead.

There is generally good correspondence between the answer type and generated questions. For example, the question generator will produce “What is the name of” if the answer is about a person, and ask “How many” if the answer is a number. We also observe that the generated questions may encounter semantic errors though syntactically fluent. For instance, in the first example, the passage suggests that people fled from Sudan to Chad, while the generated question describes the wrong direction. However, overall we think that the current question generator provides reasonable synthesized questions, yet there is still large room to improve. The observation also confirms our analysis that the synthetic question-answer pairs could be noisy and inaccurate, thus could hurt the performance when fine-tuning the answer module with generated data.

7 Conclusion

In this paper, we propose a new framework, Adversarial Domain Adaptation for MRC (AdaMRC), to transfer a pre-trained MRC model from a source domain to a target domain. We validate our proposed framework on several datasets and observe consistent improvement over baseline methods. We also verify the robustness of the proposed framework by applying it to different MRC models. Experiments also show that AdaMRC is compatible with pre-trained language model and semi-supervised learning setting. We believe our analysis provides insights that can help guide further research in this task.
References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In ICLR.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268.

Ryan Benmalek, Madian Khabsa, Suma Desu, Claire Cardie, and Michele Banko. 2019. Keeping notes: Conditional natural language generation with a scratchpad encoder. In ACL.

Rich Caruana. 1997. Multitask learning. Machine learning.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. TACL.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Yu-An Chung, Hung-Yi Lee, and James Glass. 2018. Supervised and unsupervised transfer learning for question answering. In NAACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In ICLR.

Boging Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In CVPR.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In NIPS.

Jiatao Gu, Zhendong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In ACL.

Boqing Gong, Xinwen Zhang, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In CVPR.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. TACL.

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

Zheng Li, Yun Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In IJCAI.

Xiaodong Liu, Jianfeng Gao, Xiaocong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In NAACL.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018. Stochastic answer networks for machine reading comprehension. In ACL.

Mingsheng Long, Yue Cao, Jianmin Wang, and Weizhu Chen. 2019. Multi-task deep neural networks for natural language understanding. In ACL.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning transferable features with deep adaptation networks. In ICM.
