Unsupervised Domain Adaptation on Reading Comprehension

Yu Cao[1], **Meng Fang**[2*], **Baosheng Yu**[1], **Joey Tianyi Zhou**[3]
[1]UBTECH Sydney AI Center, School of Computer Science, FEIT, The University of Sydney, Australia
[2] Tencent AI Lab [3] Institute of High Performance Computing, A*STAR, Singapore
ycao8647@uni.sdyney.edu.au, mfang@tencent.com, baosheng.yu@sydney.edu.au, zhouty@ihpc.a-star.edu.sg

Abstract

Reading comprehension (RC) has been studied in a variety of datasets with the boosted performance brought by deep neural networks. However, the generalization capability of these models across different domains remains unclear. To alleviate this issue, we are going to investigate unsupervised domain adaptation on RC, wherein a model is trained on labeled source domain and to be applied to the target domain with only unlabeled samples. We first show that even with the powerful BERT contextual representation, the performance is still unsatisfactory when the model trained on one dataset is directly applied to another target dataset. To solve this, we provide a novel conditional adversarial self-training method (CASe). Specifically, our approach leverages a BERT model fine-tuned on the source dataset along with the confidence filtering to generate reliable pseudo-labeled samples in the target domain for self-training. On the other hand, it further reduces domain distribution discrepancy through conditional adversarial learning across domains. Extensive experiments show our approach achieves comparable accuracy to supervised models on multiple large-scale benchmark datasets.

Introduction

Reading comprehension (RC) is a widely studied topic in Natural Language Processing (NLP) due to its value in human-machine interaction. In past relevant research, a variety of large-scale RC datasets were proposed, e.g., CNN/DAILYMAIL (Hermann et al. 2015), SQUAD (Rajpurkar et al. 2016), NEWSQA (Trischler et al. 2016), COQA (Reddy, Chen, and Manning 2018) and DROP (Dua et al. 2019). With a large number of annotations, these datasets make training end-to-end deep neural models possible (Wang et al. 2017; Yu et al. 2018). The more recent studies showed that BERT (Devlin et al. 2018) model achieves higher answer accuracy than human on SQUAD.

However, only unlabeled data is available in many real-world applications. It is a common challenge that machine can learn knowledge well enough in one domain and then answer questions in other domains without any labels. Unfortunately, the generalization capabilities of some existing RC neural models were proven to be weak across different datasets (Yogatama et al. 2019). In fact, the same conclusion can be drawn for BERT according to our experiment, e.g., the performance drops on CNN dataset using the model trained on SQUAD. Therefore, studies to eliminate such performance gaps between various datasets deserve effort.

A potential direction to handle it is transferring knowledge from a labeled source domain to a different unlabeled target domain, which is known as unsupervised domain adaptation (Pan and Yang 2010), leveraging data from both domains. However, only few works tried to make unsupervised domain adaptation on RC tasks. Although Chung, Lee, and Glass adapted models using a vanilla self-training, its self-labeling approach cannot ensure the labeling accuracy on a target dataset that differs much from the source one. Besides, it is only applied to some small RC datasets, so its effectiveness on large-scale datasets remains unclear and no general representation is learned. Research on large datasets is more meaningful, since they contains more different patterns than small ones. They pose a greater challenge and better fitting realistic conditions, being the basis to build strong deep neural models. In addition, analyzing the possible influential factors for transfer is also necessary, which provide guide for adaptation. Nevertheless, very limited works contribute to it (Talmor and Berant 2019).

In this paper, to make use of numerous unlabeled samples in real applications, we focus on unsupervised domain adaptation on large RC datasets. We propose a novel adaptation method, named as Conditional Adversarial Self-training (CASe). A fine-tuned BERT model will be obtained on the source domain firstly. Then specifically, in the adaptation stage, an alternated training strategy is applied, containing self-training and conditional adversarial learning in each epoch. The pseudo-labeled samples of the target dataset generated by the last model along with low-confidence filtering will be used for self-training. Compared to the method in (Chung, Lee, and Glass 2017), the filtering prevent model from learning error target domain distribution especially for large datasets. The conditional adversarial learning, whose discriminator input combines BERT features and final output logits, is utilized because the conditioning generates more comprehensive information than feature only. It en-

*Corresponding author: Meng Fang (mfang@tencent.com).

courages the model to learn generalized representations and avoid overfitting on the pseudo-labeled data.

Moreover, we test the generalization of BERT among 6 large RC datasets to prove the importance of adaptation since it fails under most conditions. The influential factors that caused the failure are also illustrated via analysis.

We validate the proposed method on different pairs of these 6 datasets, and demonstrate the baseline performance.

Our contributions can be summarized as:

- We propose a new unsupervised domain adaptation method on RC, which is alternated-staged including self-training with low-confidence filtering and conditional adversarial learning.
- We experimentally evaluate the method on 6 popular datasets, and it shows a comparable performance to models trained on target datasets, which can be regarded as a pioneer study and a baseline for future work[1].
- We show the transferability among different datasets not only depends on corpus, but also is affected by question forms significantly.

Related Work

Numerous models were proposed for RC tasks. R-NET integrates mutual attention and self-attention into RNN encoder to refine the representation (Wang et al. 2017). QANET (Yu et al. 2018) leverages similar attention in a stacked convolutional encoder to promote performance. BERT (Devlin et al. 2018) stacks multiple transformers (Vaswani et al. 2017). By applying unsupervised pre-training tasks and then fine-tuning on specific dataset, it achieves state-of-the-art performance in various NLP tasks including RC. However, none of them explores the model generalizability across different datasets, and their transferabilities still remain unknown.

Prior work on domain adaptation has been done for several NLP tasks. Some works apply instance weighting on statistical machine translation (SMT) (Foster, Goutte, and Kuhn 2010) or cross-language text classification (Wan, Pan, and Li 2011). Cross-entropy based method is used to select out-domain sentences for training SMT (Axelrod, He, and Gao 2011). There are also attempts for RC, showing that the performance of RC models on small datasets can be improved by supervised transferring from a large dataset (Min, Seo, and Hajishirzi 2017; Wiese, Weissenborn, and Neves 2017) using annotations from both domains. MultiQA (Talmor and Berant 2019) strengthens the generalizability of RC model by training on samples from various datasets. Though some studies concentrate on the generalization of RC models and analyze their performance on multiple datasets (Yogatama et al. 2019; Liu et al. 2019), they do not analyse the influential factors in detail. A parallel work for RC unsupervised domain adaptation (Chung, Lee, and Glass 2017) utilizes a simple self-labeling for re-training, and it is evaluated on 3 small datasets containing thousands of samples.

Many relevant works focus on unsupervised domain adaptation for general CV tasks. Co-training (Blum and Mitchell 1998) uses two classifiers and two data views to generate labels for unlabeled samples. Both tri-training (Zhou and Li 2005) and asymmetric tri-training (Saito, Ushiku, and Harada 2017) extend co-training by using three classifiers to generate labels, i.e., labels will be added if two classifiers make an agreement. Some approaches try to learn domain-invariant representations by selecting similar instances between domains or adding a classifier to distinguish domains (Gong, Grauman, and Sha 2013; Ganin and Lempitsky 2014). ADDA (Tzeng et al. 2017) leverages the Generative Adversarial Networks (GANs) loss on domain label to train a new network. CDAN (Long et al. 2018) applies conditional adversarial learning which combines features and labels using a multilinear mapping.

Our work is part of research on unsupervised domain adaptation as well as generalization analysis, with an emphasis on large-scale reading comprehension datasets.

Problem Definition

We first describe a standard text-span-based RC task such as SQuAD (Rajpurkar et al. 2016). Given a supporting paragraph $\mathcal{P} = \langle p_1, p_2, ..., p_M \rangle$ with M tokens and a query $\mathcal{Q} = \langle q_1, q_2, ..., q_L \rangle$ with L tokens, the answer $\mathcal{A} = \langle p_{a^s}, p_{a^s+1}, ..., p_{a^e} \rangle$ is a text piece in the original paragraph. This task aims to find out the correct answer span $(a^s, a^e), 0 \leq a^s \leq a^e \leq M$. It means that models used here need to predict two values: the start index and the end index of the answer span.

Unsupervised domain adaptation task for RC then is formally defined as follows. There is a source domain with labeled data and a target domain with unlabeled data. We have n labeled samples $\{(x_i, y_i)\}_{i=1}^{n}$ in the source domain, in which text $x_i = (\mathcal{P}_i, \mathcal{Q}_i)$ and label $y_i = (a_i^s, a_i^e)$, and n' unlabeled target domain samples $\{(x'_j)\}_{j=1}^{n'}$, sharing the same standard RC task as described above. We assume that the data in source domain is sampled from distribution $\mathcal{D}(x, y)$ and the data in target domain is sampled from distribution $\mathcal{D}'(x', y')$, $\mathcal{D} \neq \mathcal{D}'$. Our goal is to find a deep neural model that can reduce the distribution shift and achieves the optimal performance on the target domain.

Domain Adaptation Method

The main purpose of our approach is to provide a way to transfer the model for labeled data in the source domain to the target unlabeled domain. Generally, a model with good generalization can reduce the discrepancy of intermediate states generated from different distributions (Ben-David et al. 2010). We use the BERT model (Devlin et al. 2018), which is a pre-trained contextual model based on unsupervised NLP tasks with a huge 3.3-billion-word corpus. Its model depth and huge training data size ensure that it can generate universal feature representations under a variety of linguistic conditions. And we consider applying adversarial learning to minimize cross-domain discrepancy between $\mathcal{D}(x, y)$ and $\mathcal{D}'(x', y')$ (Tzeng et al. 2017). Moreover, pseudo-label based self-training (Nigam and Ghani 2000) with low-confidence filtering is also utilized for further leveraging unlabeled data in the target domain.

[1]Code is available: https://github.com/caoyu1991/CASe

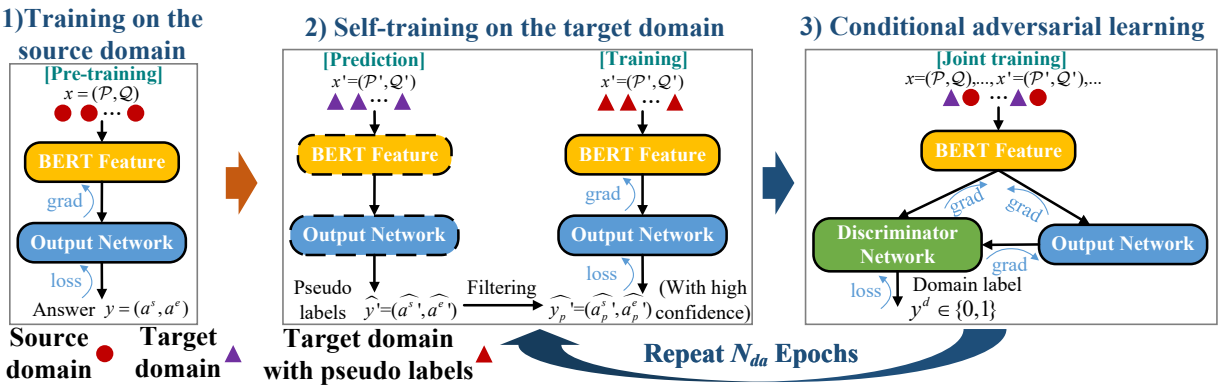


Figure 1: Framework of CASe. (Solid boxes: parameters will be updated. Dashed boxes: parameters will not be updated)

The framework of the proposed Conditional Adversarial Self-training (CASe) approach for unsupervised domain adaptation on RC is illustrated in Figure 1. Our model has three components: a BERT feature network, an output network, and a discriminator network. There are 3 steps in CASe. Firstly, we fine-tune the BERT feature model and output network on the source domain. Secondly, we use self-training on the target domain to get distribution-shifted model. Thirdly, we apply conditional adversarial learning on both domains to further reduce feature distribution divergence. The second and third steps will be proceed iteratively.

Training on the Source Domain

Since we have the labeled data in the source domain, we extend and fine-tune the unsupervised pre-trained base BERT model on these samples. The BERT feature $\bar{\mathbf{f}} \in \mathbb{R}^{m \times d}$ is firstly obtained, in which m and d are the maximum input sequence length and the hidden state dimension in BERT respectively. Then a single-layer linear output network with 2-dimension output vector is added following BERT. One of its output value is used as the answer start logits $\mathbf{g}^s \in \mathbb{R}^m$ and the other one is used as the answer end logits $\mathbf{g}^e \in \mathbb{R}^m$. Finally, the supervised pre-trained BERT model and output network can be obtained by optimizing the following loss function:

$$\mathcal{L} = \frac{1}{2}\left(f_{CE}(\mathbf{g}^s, a^s) + f_{CE}(\mathbf{g}^e, a^e)\right), \quad (1)$$

where f_{CE} is the cross entropy loss function, a^s and a^e are labels for the answer start and end indices, respectively.

To further enhance the regularization of BERT, we add a batch normalization layer (Ioffe and Szegedy 2015) between the BERT feature $\bar{\mathbf{f}} \in \mathbb{R}^{m \times d}$ and the output network.

Self-training on the Target Domain

After obtaining the pre-trained model from the source domain, we use it to predict sample labels in the target domain. Although data distribution is possibly different between domains, we can still make an assumption that different domains share some similar characteristics. That is, some predicted answers will be similar to or the same as correct answer spans even in a new domain. These predictions combined with corresponding samples $x' = (\mathcal{P}, \mathcal{Q})$ in the target domain, named as pseudo-labeled samples, can be used to teach the model about a new distribution.

Similar to the method in asymmetric tri-training (Saito, Ushiku, and Harada 2017), to avoid significant error propagation, we select predictions of high confidence as pseudo labels. Since our model generates probabilities for every predicted answer start and end index, a threshold T_{prob} will be employed to filter low-confidence samples.

Normally, we apply a softmax function to all output logits and regard generated values as possibilities for indices being the answer start or end index. However, the passage length is usually very large in RC tasks, leading to a very small probability value for each index. This method reduces the numerical distinctions between possibilities and brings more noise, which affects the effectiveness of threshold-based filtering. We thus select a set $\mathcal{U}$ of n_{best} start and end index pairs firstly. These pairs have top-n_{best} sums of start index logits g_i^s and end index logits $g_j^e, 0 \le i \le j \le M$ for corresponding answer spans involved in the target domain, i.e.,

$$\mathcal{U} = \{(i,j)_1, ..., (i,j)_{n_{best}}\} = \arg\max_{n_{best}, (i,j)} (g_i^s + g_j^e). \quad (2)$$

A softmax function then is applied to these n_{best} sums. The span with the highest value after softmax will be regarded as the predicted span and its value is defined as the generating probability p^g for current sample, i.e.,

$$p^g = \max(\text{softmax}(\{g_i^s + g_j^e\})), (i,j) \in \mathcal{U}. \quad (3)$$

Samples with $p^g \ge T_{prob}$ will be put into pseudo-labeled sample set using the predicted start and end indices as their labels, $\widehat{a^{s\prime}}$ and $\widehat{a^{e\prime}}$. The model is trained similar to (1), but a^s and a^e are replaced by $\widehat{a^{s\prime}}$ and $\widehat{a^{e\prime}}$, respectively.

In each epoch during adaptation, pseudo-labeled samples are always generated by the last model and previous ones will be abandoned, while T_{prob} keeps the same.

Conditional Adversarial Learning

Adversarial learning leverages a discriminator to predict domain classes. But most models only use feature representations for prediction (Tzeng et al. 2017; Ganin and Lempitsky 2014), which may be insufficient because the joint distribution of features and labels is not identical across domains.

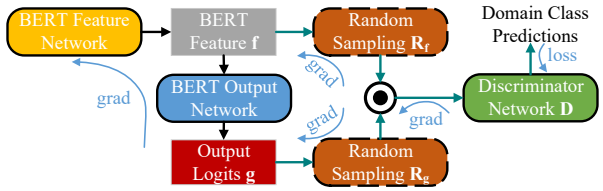


Figure 2: Architecture of the conditional adversarial network used in CASe.

Since our span-based RC tasks can be regarded as a multi-class classification problem and the span properties vary across domains, it poses more challenges for discriminators based only on features. Inspired by the Conditional Adversarial Network (CDAN) (Long et al. 2018), we utilize conditional adversarial learning fusing feature $\mathbf{f}$ and output logits $\mathbf{g}$ for a comprehensive representation, whose network architecture is illustrated in Figure 2. It is noted that $\mathbf{f} \in \mathbb{R}^{m \times d}$ is the BERT feature after the batch normalization layer.

One approach to condition discriminator D on $\mathbf{g}$ is using multilinear map, which is the outer product $\mathbf{x} \otimes \mathbf{y}$ of two vectors and is superior than concatenation (Song et al. 2010). However, it results in dimension explosion and the output dimension is $m \times d \times 2m$ in our application, which is impossible to be embedded. Following CDAN, we tackle it in a randomized approach. The multilinear map of two pairs of features and outputs can be approximated by

$$\langle \mathbf{f} \otimes \mathbf{g}, \mathbf{f}' \otimes \mathbf{g}' \rangle \approx \langle Z_R(\mathbf{f}, \mathbf{g}), Z_R(\mathbf{f}', \mathbf{g}') \rangle , \qquad (4)$$

where Z_R is a randomly sampled multilinear map and generates a vector of dimension $d_R \ll m \times d \times 2m$. Given two randomly initialized matrices fixed during training $\mathbf{R_f} \in \mathbb{R}^{d_R \times m}$ and $\mathbf{R_g} \in \mathbb{R}^{d_R \times 2m}$, Z_R can be defined as

$$Z_R(\mathbf{f}, \mathbf{g}) = \frac{1}{\sqrt{d_R}} (\mathbf{R_f} avg_{\text{col}}(\mathbf{f})) \circ (\mathbf{R_g g}) . \qquad (5)$$

Here, $\mathbf{g} = \mathbf{g}^s \oplus \mathbf{g}^e \in \mathbb{R}^{2m}$. avg_{col} means average along columns, transforming the feature matrix into a vector in $\mathbb{R}^m$, $\circ$ is element-wise multiplication.

The discriminator is a 3-layer linear network, whose final layer has a 1-dimension output with sigmoid as the activation function to get a scalar between 0 and 1. And we directly adopt $Z_R(\mathbf{f}, \mathbf{g})$ as its input for computation efficiency.

All 3 components, BERT feature network, output network, and discriminator network, are jointly optimized in this stage because discriminator conditions both features and outputs. The loss function is the binary cross entropy loss

$$\mathcal{L}_{adv} = y^d \log(\widehat{y^d}) + (1 - y^d) \log(1 - \widehat{y^d}), \qquad (6)$$

where $\widehat{y^d}$ is the prediction value from D for domain label, while $y^d \in \{0, 1\}$ is the ground truth label, 0 stands for the source domain and 1 for the target domain. Samples x, x' from both domains will be used for joint training.

However, such an optimization imposes equal importance to different samples, while samples that are hard to transfer will pose negative effect on domain adaptation. We quantify the uncertainty of a sample using entropy $E(\mathbf{p}) = -\sum_{i=1}^{M} (p_i^s \log p_i^s + p_i^e \log p_i^e)$, to ensure a more effective transfer. p_i^s and p_i^e are probabilities for i-th token being the answer start or end index, which can be obtained by applying softmax to whole output logits $\mathbf{g}^s$ and $\mathbf{g}^e$. We encourage the discriminator to place a higher priority for samples that are easy to transfer. In other words, samples with lower entropy will have higher weights during the conditional adversarial learning (CASe+E). The adversarial loss function can be reformed using the weight w derived from entropy, i.e.,

$$\mathcal{L}_{adv-E} = w \cdot \mathcal{L}_{adv}, w = 1 + e^{-E(\mathbf{p})}. \qquad (7)$$

No matter which loss is employed, the conditional adversarial learning makes the feature model and the output model more transferable and generalizable.

Algorithm

The entire procedure of CASe is shown in Algorithm 1. It is noted that no adversarial learning is included in the last epoch of domain adaptation. This aims to make the final model better fit the target domain, because adversarial learning will enhance generalization while affects fitting in specific domains. In step 16 we balance the label number of different domains by removing samples randomly from the larger dataset in merging to avoid unbalanced training.

Algorithm 1:CASe. Given a BERT feature network F, an output network G, and a discriminator D. Pre-training epoch number is N_{pre} and domain adaptation training epoch number is N_{da}

Input: data in the source domain $\mathcal{S} = \{(\mathcal{P}_i, \mathcal{Q}_i, a_i^s, a_i^e)\}_{i=1}^{n}$, data in the target domain $\mathcal{S}' = \{(\mathcal{P}_i', \mathcal{Q}_i')\}_{i=1}^{n'}$.
Output: Optimal model F, G in the target domain

1 **for** j=1 **to** N_{pre} **do**
2 Train F and G with mini-batch from $\mathcal{S}$
3 **end for**
4 **for** j=1 **to** N_{da} **do**
5 Pseudo labeled set $\mathcal{S}^P = \emptyset$
6 **for** k=1 **to** n' **do**
7 Use F, G to predict the label $\widehat{a_k^{s\prime}}$ and $\widehat{a_k^{e\prime}}$ for $(\mathcal{P}_k', \mathcal{Q}_k')$ and get probability p_k^g
8 **if** $p_k^g \geq T_{prob}$ **do**
9 Put $(\mathcal{P}_k', \mathcal{Q}_k', \widehat{a_k^{s\prime}}, \widehat{a_k^{e\prime}})$ into $\mathcal{S}^P$
10 **end if**
11 **end for**
12 **for** mini-batch $\mathcal{B}$ **in** $\mathcal{S}^P$
13 Train F and G with mini-batch $\mathcal{B}$
14 **end for**
15 **if** j $< N_{da}$ **do**
16 $\mathcal{R} = (\{(\mathcal{P}_i, \mathcal{Q}_i)\}_{i=1}^{n}) \cup \mathcal{S}'$
17 **for** mini-batch $\mathcal{B}$ **in** $\mathcal{R}$
18 Train F,G,D with $\mathcal{B}$ and domain labels
19 **end for**
20 **end if**
21 **end for**

Experiment

In this section, we first evaluate the generalization of BERT among 6 recently release RC datasets and analyze influential

factors. Then the performance of proposed CASe for unsupervised domain adaptation on these datasets be given, along with ablation study and the effects of hyperparameters.

Dataset

SQuAD (Rajpurkar et al. 2016) contains 87k training samples and 11k validation (dev) samples, with questions in natural language given by workers based on paragraphs from Wikipeida, and answers are in text span forms.
CNN and **DailyMail** (Hermann et al. 2015) contains 374k training and 4k dev samples, 872k training and 64k dev samples respectively. Their questions are in cloze forms and answers are masked entities in passages.
NewsQA (Trischler et al. 2016) contains 120k samples in total, in which QA pairs were generated by crowded workers in natural forms with text spans based on stories from **CNN**.
CoQA (Reddy, Chen, and Manning 2018) contains 109k training samples and 8k dev samples, questions are given as conversation forms with multiple turns and answers are in various types including text spans and yes/no.
DROP (Dua et al. 2019) contains 77k training samples and 9.5k dev samples, given by workers on Wikipedia. It mainly focuses on numerical reasoning and involves answers in numbers or dates except text spans.

Since CNN and DailyMail is much larger than other datasets, we uniformly sampled subsets from two datasets as data source to speed up experiments. The keep ratio is 1/4 and 1/10 respectively, resulting in similar scales as others.

In addition, we pre-processed samples to conduct answer spans for several datasets. The answers in CNN and DailyMail are mask symbols such as "*@entity1*" which may appears several times in the text. We use a heuristic method to extract spans: 1) find all position indices $\{a_i\}$ of answer masks in a passage; 2) find all position indices $\{\{e_i^1\}, ..., \{e_i^K\}\}$ of all K question entities in passage; 3) calculate the sum of absolute index distances between an answer appearance a_j and every question entity nearest to it, and a_j with the smallest sum will be used as answer index. All masks in these two datasets are also replaced with homologous original tokens. CoQA contains answers not in text span form. We follow the F1-socre-based method in original paper to obtain the best answer spans. And the concatenation of all previous QA pairs along with the original question in current turn is used as new question. Samples with yes/no as answers or no answer span being found will be discarded. Similarly, we only remain answerable questions with text spans as answers in NewsQA and DROP.

The characterizations of 6 processed datasets are shown in Table 1. DROP is significantly smaller than others because answers of quantitive reasoning samples are not extractive.

Implementation Detail

We implement CASe based on the BERT implementation in PyTorch by Hugging Face, using the *base-uncased* pre-trained model with 12 layers and 768-dim hidden state. The maximum input length m is 512 in which the maximum query length is 40. The random sampling dimension d_R is 768. The input dimension of the first layer in the adversarial network is 768. And its intermediate dimension is 512, using ReLU as the activation function in first two layers. Generating probability threshold T_{prob} is set as 0.4 and $n_{best} = 20$. Adam optimizer (Kingma and Ba 2014) is employed with learning rate 3×10^{-5} in the source domain training, 2×10^{-5} in the self-training and 10^{-5} in the adversarial learning, with batch size 12. A dropout with rate 0.2 is applied on both the BERT feature network and the discriminator. We set the epoch number $N_{pre} = 3$ in pre-training and $N_{da} = 4$ in domain adaptation.

Besides, since the input length may be larger than m, we truncate a passage using a sliding window to fit the input length whose moving step is 128. And text pieces excluding the answers will be discarded in training.

Dataset	Train	Dev	Corpus	Question
SQuAD	87,599	10,570	Wikipedia	crowd
CNN	93,627	3,833	CNN news	cloze
DailyMail	87,253	6,372	Daily mail	cloze
NewsQA	76,341	4,327	CNN news	crowd
CoQA	86,077	6,272	Multiple*	crowd
DROP	28,267	3,389	Wikipedia	crowd

Table 1: Characterizations of datasets **after processing**. (*Including corpus from MCTest, CNN, Wikipedia etc.)

Generalization and Influential Factors

We firstly test the generalization capability of BERT by fine-tuning it on one dataset and directly applying it to another dataset without any change. We call such models as **zero-shot** models. The performance on dev sets for transferring among 6 datasets is shown in Table 2.

In a high-level observation, the performance of zero-shot models drops significantly in most cases except the transferring between CNN and DailyMail. The average 55.8% reduction in exact match (EM) and 50.0% reduction in F1 compared to models trained on the target dataset (Self) prove that BERT cannot generalize well to unseen datasets, despite a huge corpus is used in unsupervised pre-training.

Taking a closer look, we can find the reductions vary across different dataset pairs. The drops of transferring among 4 datasets, SQuAD, NewsQA, CoQA and DROP, are smaller than transferring to/from rest 2 datasets, especially from latter 3 ones to SQuAD. And the transferring between CNN and DailyMail achieves equivalent performance to Self. CNN and NewsQA share the same corpus but the transferring fails due to different question forms(natural vs. cloze), and the corpus discrepancy of SQuAD and NewsQA leads to homologous result. On the other hand, the same question forms and similar corpora of CNN and DailyMail make successful transferring. Therefore, it can be concluded that not only the corpus but also the question form affect the generalization. It is also observed that the different focus as well as reasoning types affect the transfer between datasets even with same corpus and question type, i.e. simple single-sentence reasoning in SQuAD vs. complex reasoning (comparison, selection) in DROP.

We visualize the relations between 6 datasets using force-directed graph in Figure 3. The force between every two

Datasets	SQuAD	CNN	DailyMail	NewsQA	CoQA	DROP
SQuAD	-	16.72 / 26.42	21.12 / 21.70	**40.03 / 57.42**	**29.58 / 39.58**	**19.06 / 29.73**
CNN	18.97 / 24.34	-	**81.53 / 83.59**	9.38 / 15.36	7.10 / 10.26	4.40 / 7.50
DailyMail	9.72 / 14.76	**77.22 / 79.73**	-	5.89 / 10.69	5.68 / 8.75	4.69 / 8.02
NewsQA	64.80 / 78.32	25.10 / 34.66	28.41 / 38.44	-	27.14 / 38.75	12.36 / 21.00
CoQA	**65.25 / 74.92**	18.21 / 24.76	22.65 / 28.12	37.74 / 53.85	-	14.75 / 21.60
DROP	55.53 / 68.36	14.32 / 22.26	17.44 / 25.78	28.36 / 44.35	16.15 / 24.82	-
Self	79.85 / 87.46	82.76 / 84.73	81.37// 83.33	52.05 / 67.41	48.98 / 63.99	44.67 / 52.51

Table 2: Performance of zero-shot models on dev set when transferring among datasets. Rows correspond to source datasets and columns to target datasets. Self means training and testing on the same dataset. Left value in each cell is for **exact match (EM)** while the right one is for **F1 score.**

Datasets	SQuAD	CNN	DailyMail	NewsQA	CoQA	DROP
SQuAD	-	**80.64 / 82.24**	80.78 / 82.77	**52.69 / 68.15**	**52.38 / 67.56**	**50.34 / 57.53**
CNN	79.86 / 87.65	-	**84.26 / 86.01**	48.37 / 63.47	51.71 / 67.09	45.59 / 53.57
DailyMail	79.04 / 87.07	78.06 / 80.36	-	50.13 / 65.90	50.06 / 65.76	41.69 / 50.07
NewsQA	**80.17 / 88.14**	79.60 / 81.57	80.93 / 82.99	-	50.05 / 66.49	47.36 / 56.42
CoQA	78.38 / 85.93	74.75 / 76.65	76.87 / 78.88	51.21 / 65.83	-	42.08 / 50.07
DROP	74.03 / 83.35	77.09 / 79.03	80.34 / 82.49	51.91 / 66.95	48.90 / 64.29	-
SQuAD	-	80.20 / 81.93	79.91 / 82.06	**51.56 / 66.79**	**50.77 / 65.94**	**48.45 / 57.33**
CNN	78.59 / 86.39	-	**83.40 / 85.06**	48.95 / 64.45	49.38 / 64.57	44.15 / 51.87
DailyMail	78.07 / 86.22	**82.44 / 84.36**	-	50.91 / 65.90	48.64 / 63.80	41.58 / 47.74
NewsQA	**78.87 / 87.06**	80.49 / 82.43	80.93 / 82.99	80.99 / 83.07	48.01 / 64.30	45.06 / 54.34
CoQA	78.24 / 85.80	76.34 / 78.22	78.12 / 79.88	50.80 / 65.55	-	41.43 / 49.40
DROP	74.81 / 83.67	80.38 / 82.21	80.78 / 82.96	50.01 / 65.16	46.27 / 62.67	-
Self	79.85 / 87.46	82.76 / 84.73	81.37// 83.33	52.05 / 67.41	48.98 / 63.99	44.67 / 52.51

Table 3: Domain adaptation performance of CASe on dev sets of datasets. The top of the table shows results for CASe+E (Entropy-weighted loss), while the bottom for standard CASe. Rows are source datasets and columns are target datasets. The left value in each cell is **exact match(EM)**, while right one is **F1 score**. Self stands for training and testing on the same dataset.

datasets can be calculate via $F_{ij} = P_{ij}/P_j + P_{ji}/P_i$. P_{ij} is the average performance of EM and F1 from source dataset i to target dataset j, and P_i is the average performance of Self model on dataset i. Edge widths are positively correlated to force F between nodes, while the size of each node reflects dataset scale. It is noted that datasets cluster more significantly according to **question forms** (node shapes), comparing to **corpora** (node colors) who also affect it.

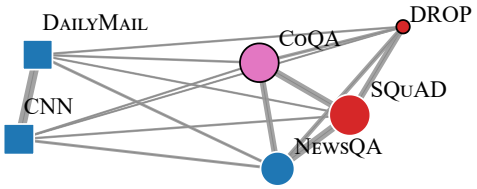


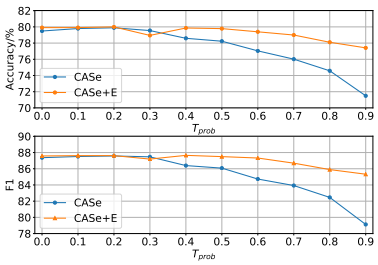
Figure 3: Visualization of relations between datasets based on performance. Node shape represents question form (rectangle:cloze, circle:natural). Node color represents corpus (red: Wikipedia, blue: news, purple: multiple).

Domain Adaptation Performance of CASe

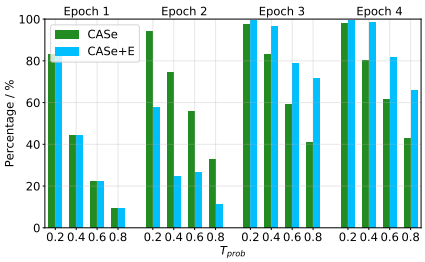
We now evaluate the performance of proposed CASe method for unsupervised domain adaptation on RC datasets, including standard CASe and CASe with entropy-weighted loss in adversarial learning (CASe+E). The results are shown in Table 3. Generally speaking, no matter which loss function is used in adversarial learning, CASe achieves significant performance improvement compared to zero-shot models. Despite annotated data is unavailable in the target domain, most results are comparable to Self models, and some of them are even better. In conclusion, CASe transfers knowledge from one domain to another one successfully.

Domain adapted models between two very alike datasets, CNN and DailyMail, shows a higher accuracy than Self. They are similar on both corpora and question forms, which means more valid data can be utilized for self-training to get a model with deeper comprehension. Zero-shot model performs poorly when transferring between natural-question-based datasets and cloze-question-based datasets, e.g., SQuAD to CNN. But CASe can nearly eliminate such gaps between transferred model and Self models due to the new distribution learned in self-training and generalized representation optimized in adversarial learning. The performance of most adaptations on CoQA and DROP is better than Self because they benefit from more extra data.

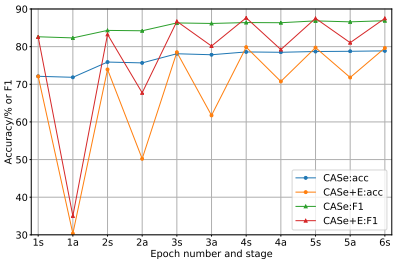
Entropy-based loss weighting also show its effectiveness because it makes learning focus on samples simple to be transferred so as to obtain more correct knowledge in the tar-



(a) Performance varies with T_{prob} (Upper: EM, lower: F1).



(b) Numbers of pseudo-labeled samples generated in each epoch under different T_{prob}.



(c) Performance varies with adaptation stages and epoch numbers when $T_{prob} = 0.4$.

Figure 4: Influence of hyperparameters on adaptation performance of CASe and CASe+E under CNN to SQuAD (**C→S**).

get domain. And CASe+E shows 0.5% to 2% higher in accuracy than CASe under most conditions except some specific dataset pairs such as DAILYMAIL to CNN.

Ablation Study We do ablation test on 4 domain adaptation dataset pairs, which are CNN to SQUAD (**C→S**), DAILYMAIL to CNN (**D→C**), CNN to NEWSQA (**C→N**) and SQUAD to COQA (**S→Co**), including adaptation between datasets with same/different question forms and/or corpora. The EM results on ablated models are shows in Table 4, in which - *conditional* means using unconditional adversarial learning instead of conditional one, while - *Adv learning* for removing whole adversarial learning, - *Self-training* for removing self-training and - *Batch norm* for removing batch normalization, all based on CASe. It is observed that self-training plays the most important role under all configurations. Performance drops without discriminator conditioning on output or whole adversarial learning. Batch normalization has slight effect, removing it promotes the results under two configurations while it has opposite effect under others.

Generalization after domain adaptation We test the performance of transferred models on the source datasets to check their generalization, which is shown in Table 5. 4 datasets pairs in ablation study is involved plus NEWSQA to DROP (**N→Dr**). There are performance declines compared to models trained on the source datasets, except **D→C** in which datasets have very similar properties. It means our CASe method results in a good transferred model at the meantime leads to knowledge loss in the source domain.

Impact of T_{prob} Figure 4(a) demonstrates the performance of CASe and CASe+E on **C→S** varied with different generating probability T_{prob} in terms of accuracy and F1 scores. CASe+E shows higher stability and performance than CASe under different T_{prob}. CASe and CASe+E reach their peaks at 0.3 and 0.4 respectively, while both of them show descending trends when $T_{prob} \geq 0.4$.

The numbers of generated pseudo-labeled samples in every epoch on **C→S** with different T_{prob} are shown in Figure 4(b). Obviously, a lower threshold results in more samples and longer training time. Although CASe generate more samples stably than previous epoch, samples generated by CASe+E may decrease in the 2nd epoch, but more samples will be generated latter compared to CASe. Thus CASe+E achieves better results under most conditions because more valid samples are utilized. Considering the overall performance as well as the trade-off between accuracy and complexity, we set T_{prob} as 0.4 in our experiment.

	C→S	**D→C**	**C→N**	**S→Co**
CASe+E	66.46	78.06	48.37	52.38
CASe	65.24	82.44	48.95	50.77
- *conditional*	64.47	82.26	47.31	50.25
- *Adv learning*	65.05	81.21	47.89	49.05
- *Self-training*	16.55	77.07	14.26	23.81
- *Batch norm*	65.97	81.91	48.27	51.08

Table 4: EM results of CASe ablation test on 4 dataset pairs.

	C→S	**D→C**	**C→N**	**S→Co**	**N→Dr**
CASe+E	66.37	82.19	64.65	52.97	40.07
CASe	68.61	81.61	65.43	51.48	40.17
SELF	80.77	80.85	80.77	66.51	52.05

Table 5: EM results on source datasets after adaptaiton.

Impact of epoch number In Figure 4(c), we present the performance of CASe and CASe+E after different stages in every epoch on **C→S**. E.g., *1s* means result after the self-training stage in 1st epoch, *2a* means results after conditional adversarial learning stage in 2nd epoch. CASe+E shows obvious fluctuations between the self-training and the adversarial learning compared to CASe. Not matter CASe or CASe+E, the performance tends to be saturated after 3 complete epochs. That is the reason why we set N_{da} as 4.

Conclusion

In this paper, we explore the possibility of transferring reading comprehension model from a large-scale labeled dataset to another unlabeled one. Our experiment proves that even the BERT model cannot generalize well between different datasets, and the divergence of both corpora and question forms results in this failure. Then we propose a new unsupervised domain adaptation method, Conditional Adversarial Self-training (CASe). After fine-tuning a BERT model on source data, it uses self-training and conditional adversarial

learning alternately in every epoch to make the model better fit the target domain and reduce the domain distribution discrepancy. The experimental results among 6 RC datasets demonstrate the effectiveness of CASe. It promotes performance remarkably over zero-shot models, showing similar accuracies to supervised trained on the target domain.

Acknowledgements

We thank Boqing Gong and the anonymous reviewers for insightful comments and feedback.

References

Axelrod, A.; He, X.; and Gao, J. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing*, 355–362. Association for Computational Linguistics.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine learning* 79(1-2):151–175.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 92–100. ACM.

Chung, Y.-A.; Lee, H.-Y.; and Glass, J. 2017. Supervised and unsupervised transfer learning for question answering. *arXiv preprint arXiv:1711.05345*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dua, D.; Wang, Y.; Dasigi, P.; Stanovsky, G.; Singh, S.; and Gardner, M. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

Foster, G.; Goutte, C.; and Kuhn, R. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, 451–459. Association for Computational Linguistics.

Ganin, Y., and Lempitsky, V. 2014. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*.

Gong, B.; Grauman, K.; and Sha, F. 2013. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning*, 222–230.

Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, 1693–1701.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 1640–1650.

Min, S.; Seo, M.; and Hajishirzi, H. 2017. Question answering through transfer learning from large fine-grained supervision data. *arXiv preprint arXiv:1702.02171*.

Nigam, K., and Ghani, R. 2000. Analyzing the effectiveness and applicability of co-training.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10):1345–1359.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Reddy, S.; Chen, D.; and Manning, C. D. 2018. Coqa: A conversational question answering challenge. *arXiv preprint arXiv:1808.07042*.

Saito, K.; Ushiku, Y.; and Harada, T. 2017. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2988–2997. JMLR. org.

Song, L.; Boots, B.; Siddiqi, S.; Gordon, G. J.; and Smola, A. 2010. Hilbert space embeddings of hidden markov models.

Talmor, A., and Berant, J. 2019. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. *arXiv preprint arXiv:1905.13453*.

Trischler, A.; Wang, T.; Yuan, X.; Harris, J.; Sordoni, A.; Bachman, P.; and Suleman, K. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7167–7176.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wan, C.; Pan, R.; and Li, J. 2011. Bi-weighting domain adaptation for cross-language text classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. R-net: Machine reading comprehension with self-matching networks. *Natural Lang. Comput. Group, Microsoft Res. Asia, Beijing, China, Tech. Rep* 5.

Wiese, G.; Weissenborn, D.; and Neves, M. 2017. Neural domain adaptation for biomedical question answering. *arXiv preprint arXiv:1706.03610*.

Yogatama, D.; d'Autume, C. d. M.; Connor, J.; Kocisky, T.; Chrzanowski, M.; Kong, L.; Lazaridou, A.; Ling, W.; Yu, L.; Dyer, C.; et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Yu, A. W.; Dohan, D.; Luong, M.-T.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Zhou, Z.-H., and Li, M. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge & Data Engineering* (11):1529–1541.