Unsupervised Domain Adaptation on Reading Comprehension

Cao et al.

Minbyul Jeong

Data Mining & Information Systems Lab.
Department of Computer Science and Engineering,
College of Informatics, Korea University

1 Introduction

Reading Comprehension Task (Generalization Capabilities)

- Problem -

(1) There are no proving paper about generalization capabilities of pre-trained contextualized representation model (e.g., BERT)

(2) Low performance on Transfer Learning through different dataset (Source domain dataset to Target domain dataset such as SQuAD to DROP)

- Solution -

- Reducing source domain and target domain discrepancy through Conditional Adversarial Self-training method (CASe)

- Contribution -

- Transferability of different datasets depends on corpus & question forms
- Unsupervised domain adaptation method (CASe & low-confidence filtering)
    a. Confidence Filtering to generate reliable pseudo-labeled samples

2 Problem Definition

P = $\langle p_1, p_2, ..., p_M \rangle$: passage with M tokens

Q = $\langle q_1, q_2, ..., q_L \rangle$: query with L tokens

A = $\langle p_{a^s}, p_{a^s+1}, ..., p_{a^e} \rangle$: Answer text piece in the paragraph

$(a^s, a^e), 0 \leq a^s \leq a^e \leq M$: Answer start index and end index

Source domain (Labeled data) & Target domain (Unlabeled data)

Source domain : {P, Q, (a^s, a^e)}

Target domain : {P', Q'}

Assumption : distribution of source domain data is different from a

distribution of target domain data → reduce distribution shift

3 Training

Goal: reduce the discrepancy of two domain distributions

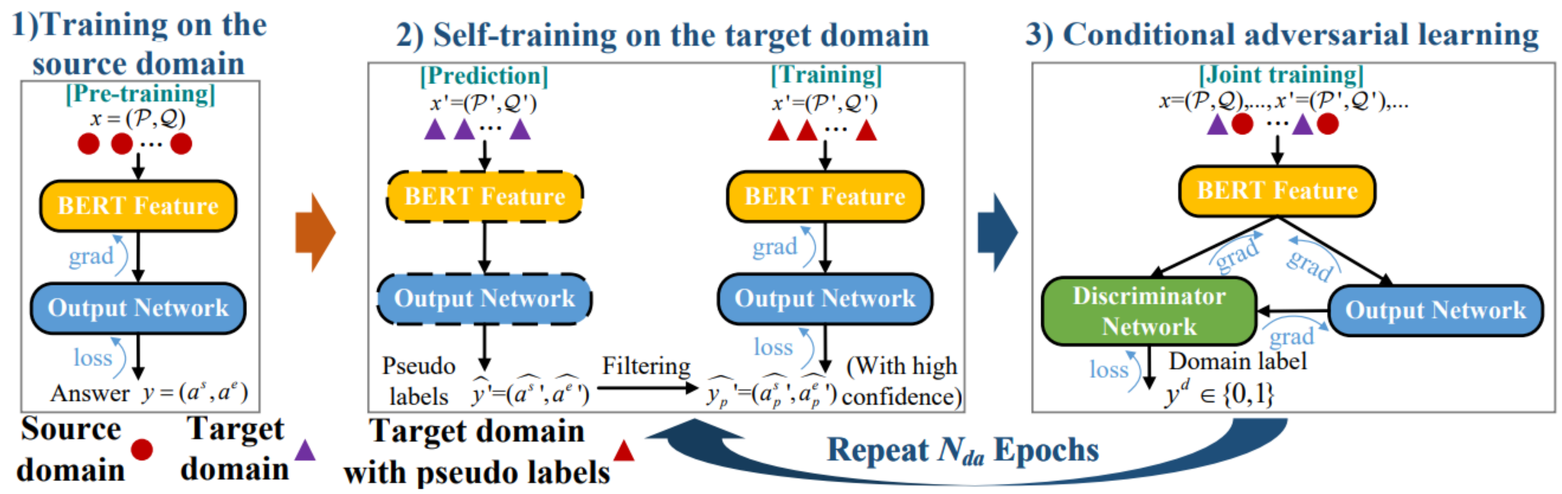
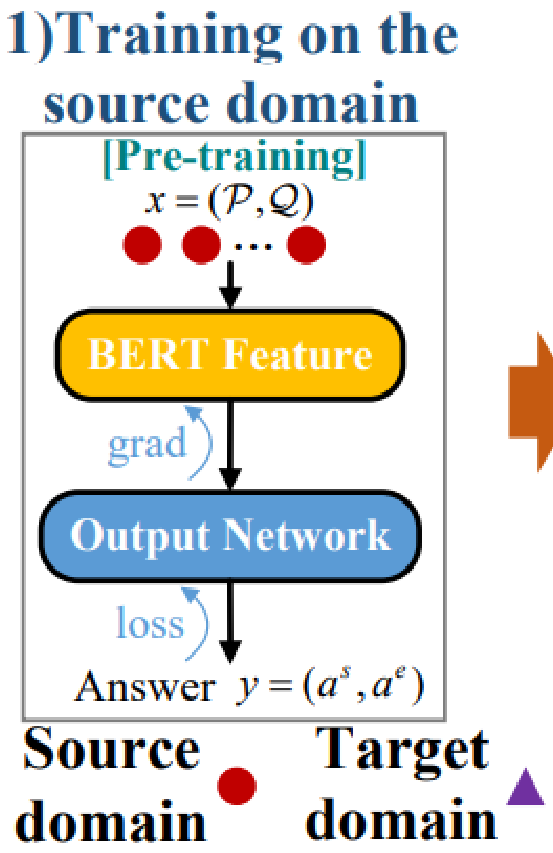


Figure 1: Framework of CASe. (Solid boxes: parameters will be updated. Dashed boxes: parameters will not be updated)

(1) Training on the source domain to generate fine-tuned BERT feature
(2) Self-training on the target domain to get distribution-shifted model
(3) Applying Conditional Adversarial Learning on both domains to reduce feature distribution divergence.
(4) Proceeding (2) and (3) step iteratively

3 Training

First step : Training on the source domain to generate fine-tuned BERT feature



Commonly Used Procedure

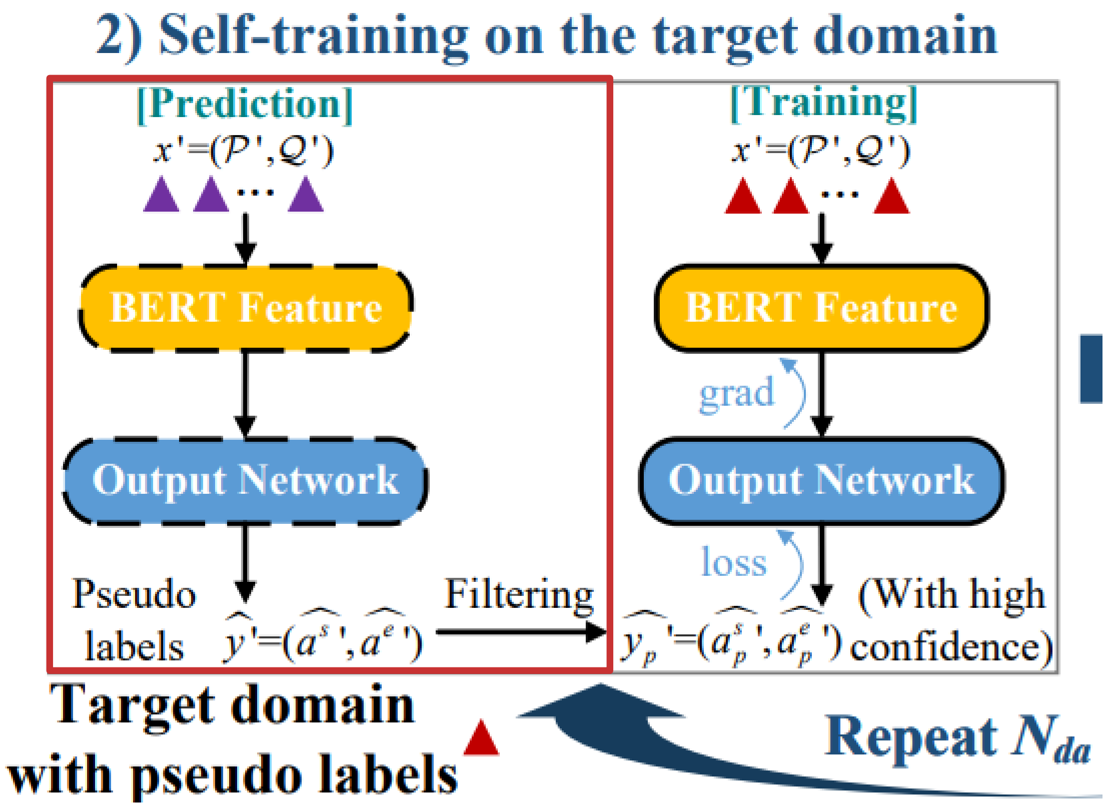
(1) Given passage and query, predicting span of answer index at passage

(2) Comparing predictions and answers to compute the loss

(3) Back propagate to train BERT model.

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

3 Training

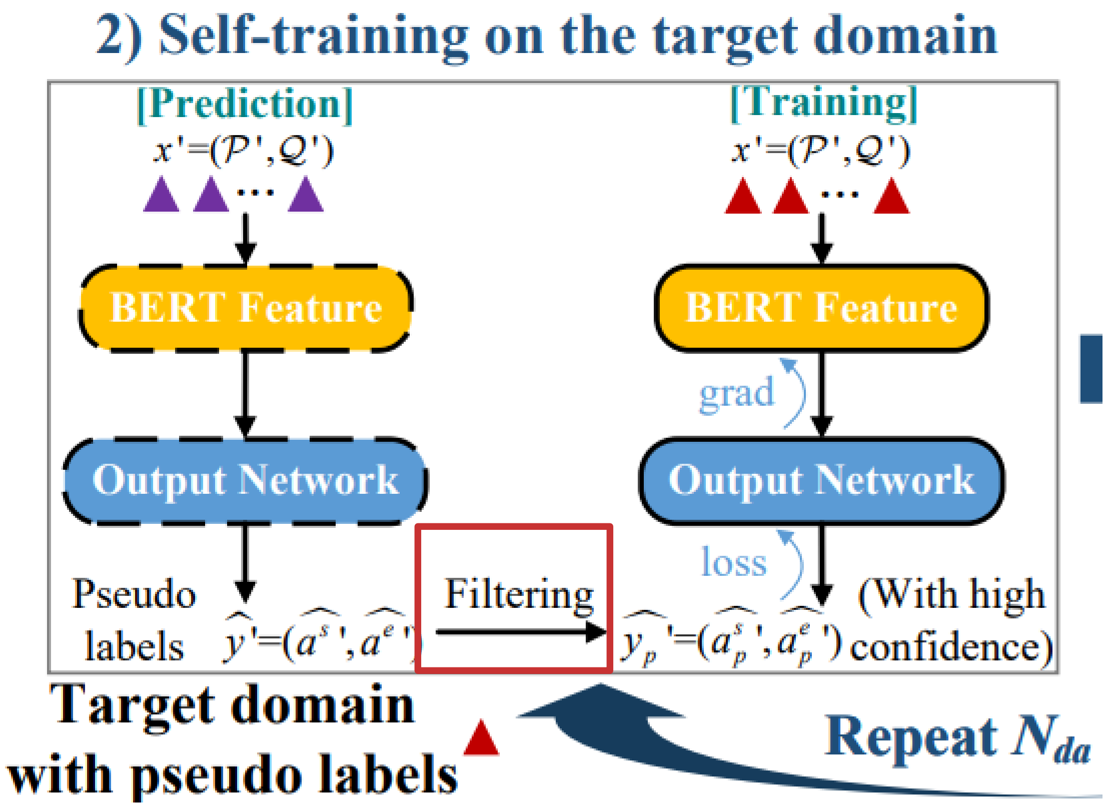

2) Self-training on the target domain
[Prediction]
$x'=(\mathcal{P}',\mathcal{Q}')$
BERT Feature
Output Network
Pseudo labels
$\widehat{y}'=(\widehat{a^s}',\widehat{a^e}')$
Filtering
[Training]
$x'=(\mathcal{P}',\mathcal{Q}')$
BERT Feature
grad
Output Network
loss
(With high confidence)
$\widehat{y_p}'=(\widehat{a_p^s}',\widehat{a_p^e}')$
Target domain with pseudo labels
Repeat N_{da}

Make Pseudo labels in target domain

(1) Use fine-tuned BERT Feature and Output Network to predict Pseudo labels
(2) Select Hyperparameter T_{prob}

(3) Generate probabilities for every predicted answer start and end index
(4) Filter low-confidence samples

3 Training



- Problem

Getting probabilities about all sequence length leads to very small probability value for each index

- Solution

Top 20 best sums of start index logit and end index logit.

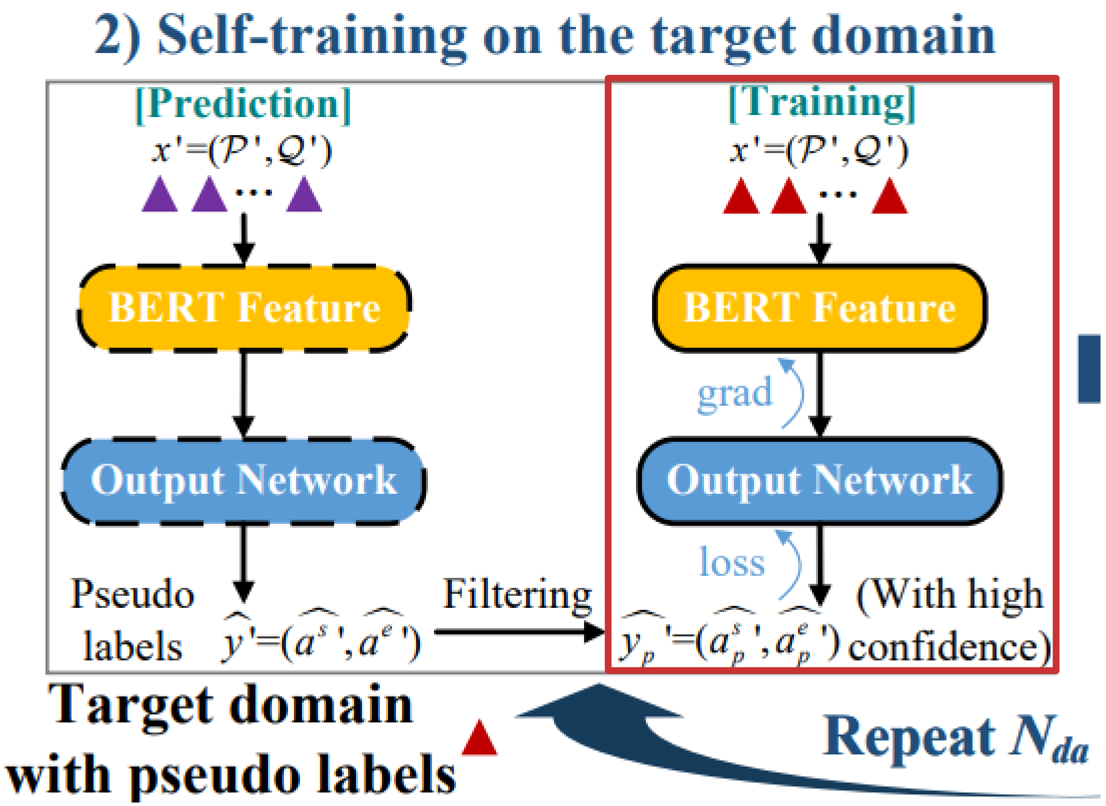
$$\mathcal{U} = \{(i,j)_1, ..., (i,j)_{n_{best}}\} = \arg\max_{(i,j)} {}_{n_{best}} (g_i^s + g_j^e).$$

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

Samples with $p^g \geq T_{prob}$

During Adaptation: **pseudo-labeled samples are always generated by the last model**

3 Training



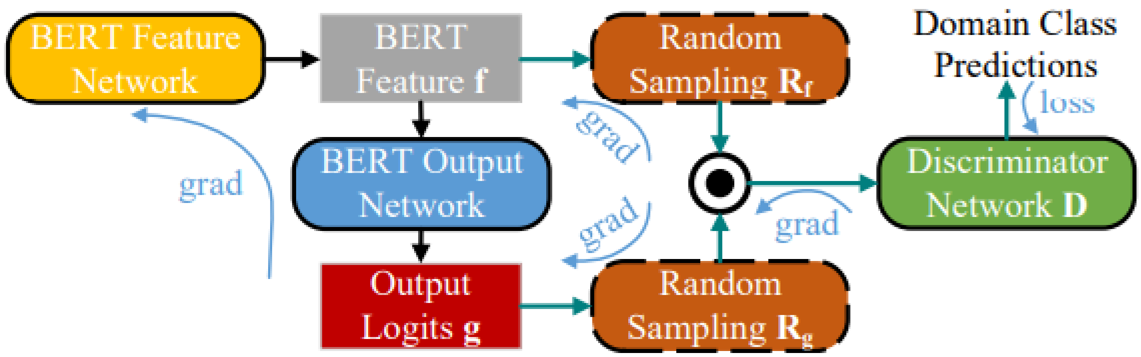
2) Self-training on the target domain
[Prediction]
$x'=(\mathcal{P}',\mathcal{Q}')$
BERT Feature
Output Network
Pseudo labels
$\widehat{y}'=(\widehat{a^s}',\widehat{a^e}')$
Filtering
[Training]
$x'=(\mathcal{P}',\mathcal{Q}')$
BERT Feature
grad
Output Network
loss
$\widehat{y_p}'=(\widehat{a_p^s}',\widehat{a_p^e}')$
(With high confidence)
Target domain with pseudo labels
Repeat N_{da}

Same Procedure at first step

Only difference is the answer span!

Convert correct answer span to high confidence pseudo labels

3 Training



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

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

Z_R is a randomly sampled multilinear map

$$\mathbf{f} \in \mathbb{R}^{m \times d}$$

$$\mathbf{g} = \mathbf{g}^s \oplus \mathbf{g}^e \in \mathbb{R}^{2m}$$

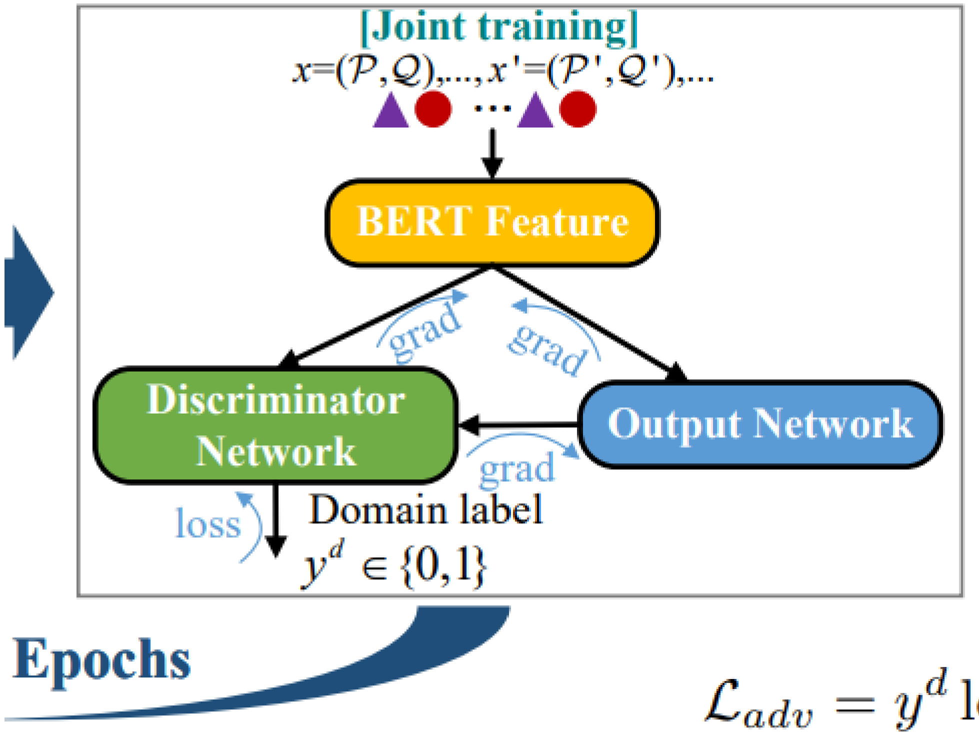
$$\mathbf{R_f} \in \mathbb{R}^{d_R \times m}$$

$$\mathbf{R_g} \in \mathbb{R}^{d_R \times 2m}$$

$$d_R \ll m \times d \times 2m$$

3 Training

3) Conditional adversarial learning



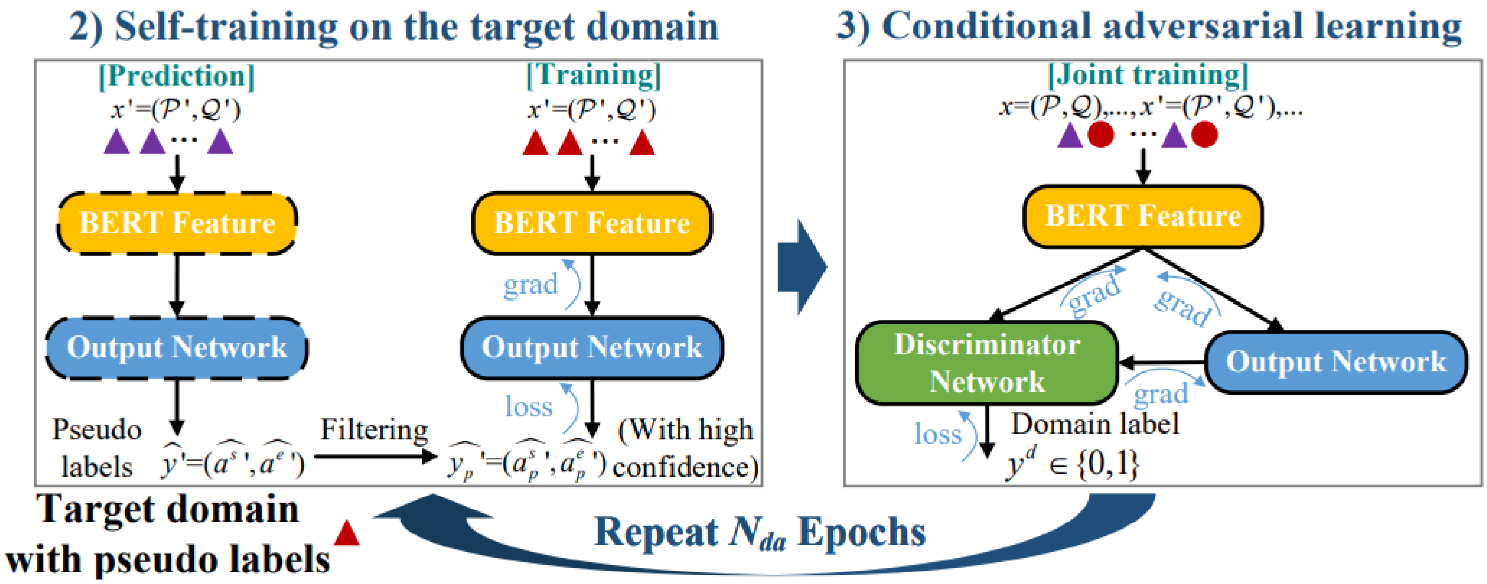
Discriminator Network

(1) 3-layer linear network

(2) Final layer has sigmoid activation function to get a scalar of 0 and 1.

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

3 Training


2) Self-training on the target domain
[Prediction]
$x'=(\mathcal{P}',\mathcal{Q}')$
BERT Feature
Output Network
Pseudo labels $\widehat{y}'=(\widehat{a^s}',\widehat{a^e}')$
Filtering
[Training]
$x'=(\mathcal{P}',\mathcal{Q}')$
BERT Feature
grad
Output Network
loss
$\widehat{y_p}'=(\widehat{a_p^s}',\widehat{a_p^e}')$ (With high confidence)
Target domain with pseudo labels
3) Conditional adversarial learning
[Joint training]
$x=(\mathcal{P},\mathcal{Q}),\ldots,x'=(\mathcal{P}',\mathcal{Q}'),\ldots$
BERT Feature
grad
grad
Discriminator Network
Output Network
grad
loss
Domain label $y^d \in \{0,1\}$
Repeat N_{da} Epochs

3 Training

- Problem

Equal importance to different samples that are hard to transfer will pose negative effect on domain adaptation

Check uncertainty of each sample by E(p)

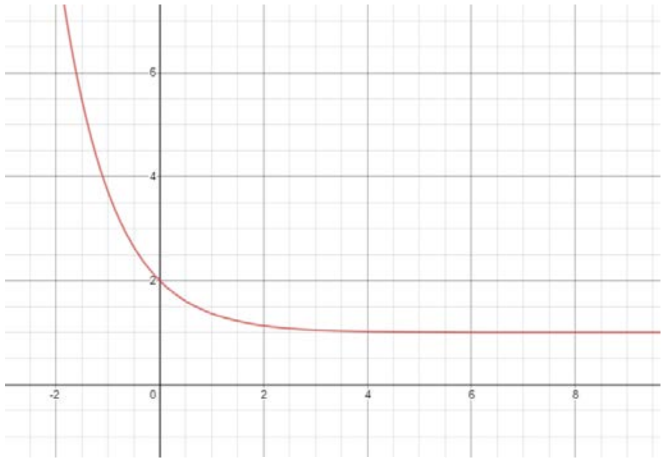
High priority for easy transfer sample → Lower E(p) → higher weight to the adversarial loss
Low priority for hard transfer sample → Higher E(p) → lower weight to the adversarial loss

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

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

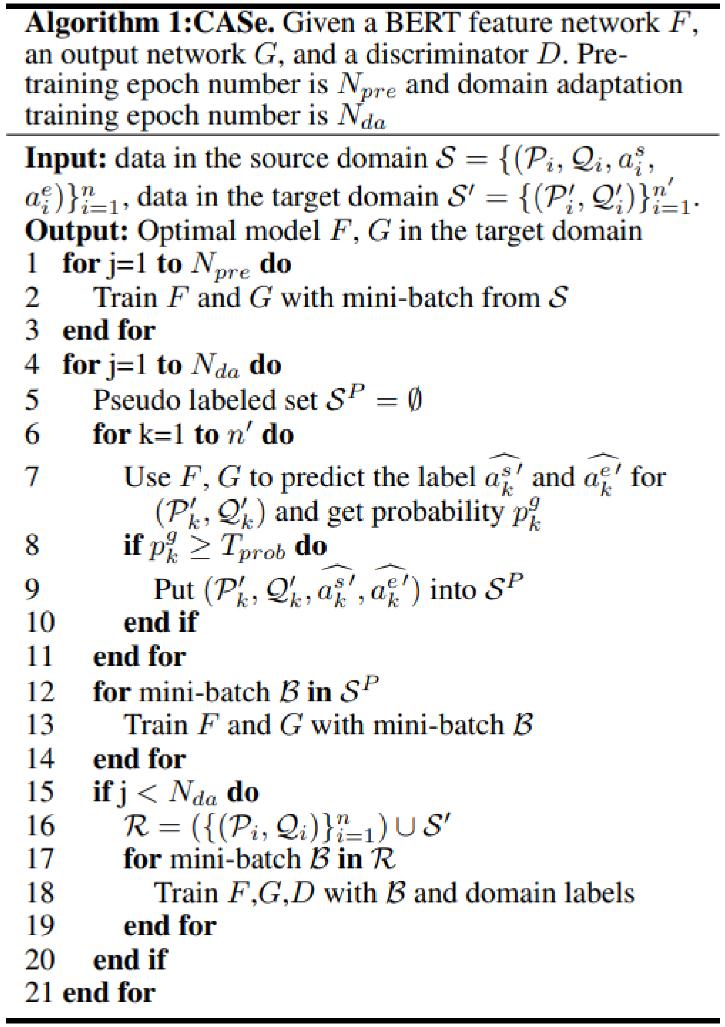
$$E(\mathbf{p}) = -\sum_{i=1}^{M}(p_i^s \log p_i^s + p_i^e \log p_i^e)$$

Make more transferable, generalizable model



$1 + e^{-x}$

3 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 $\widehat{a_k^{s}}'$ and $\widehat{a_k^{e}}'$ for
             $(\mathcal{P}_k', \mathcal{Q}_k')$ and get probability $p_k^g$
8          if $p_k^g \geq T_{prob}$ do
9              Put $(\mathcal{P}_k', \mathcal{Q}_k', \widehat{a_k^{s}}', \widehat{a_k^{e}}')$ 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
```

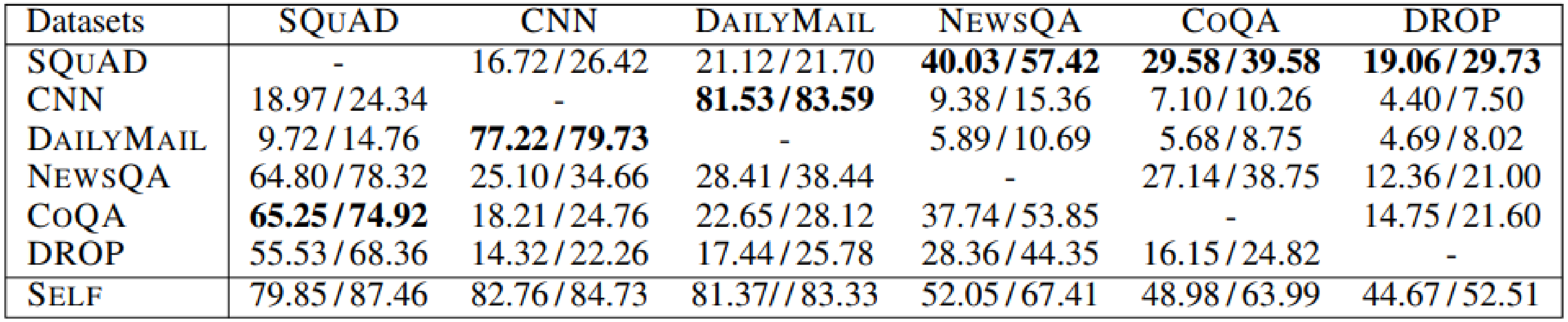
1~3 lines: Source domain [Pre-training]

6~11 lines: Second step [Prediction]

12~14 lines: Self training

15~20 lines: Adversarial Tranining

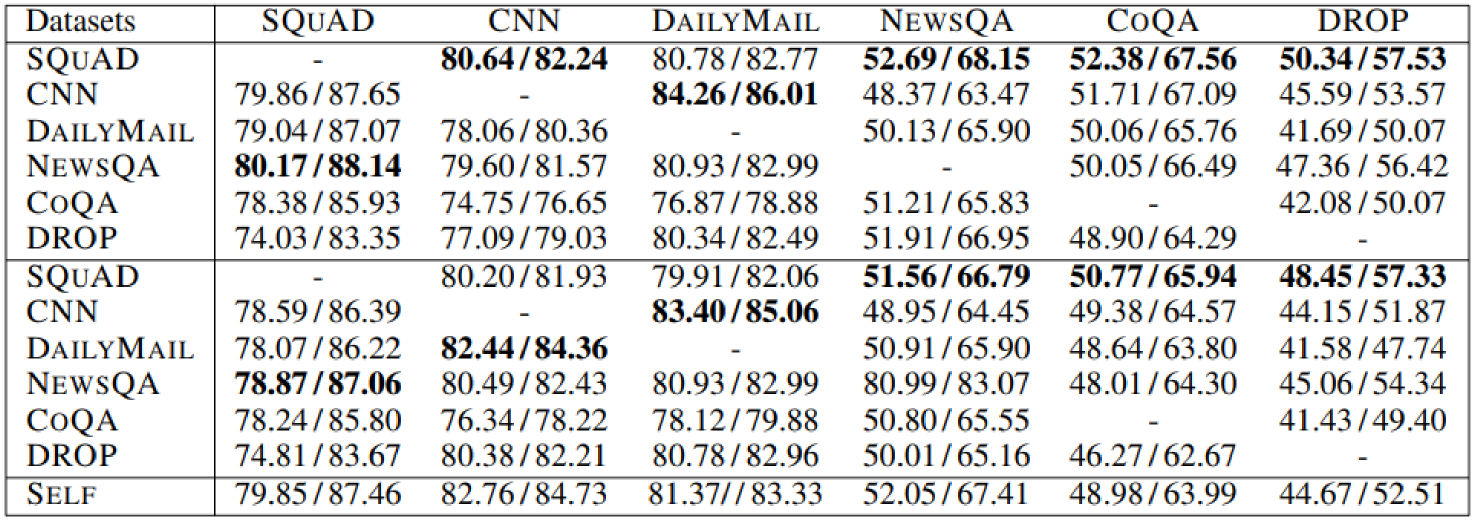
4 Evaluation



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.**

4 # Evaluation



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.

4 Evaluation

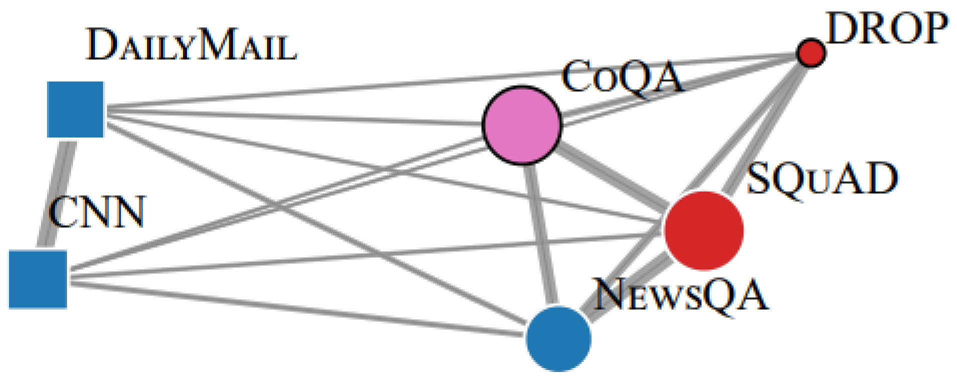


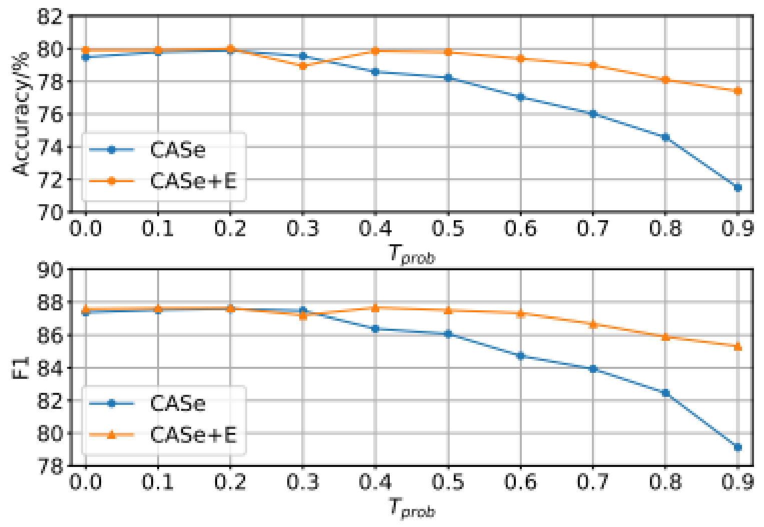
Figure 3: Visualization of relations between datasets based on performance. Node shape represents question form (rectangle:cloze, circle:natural). Node color represents corpus (red: Wikipedia, blue: news, purple: multiple).

$$F_{ij} = P_{ij}/P_j + P_{ji}/P_i$$

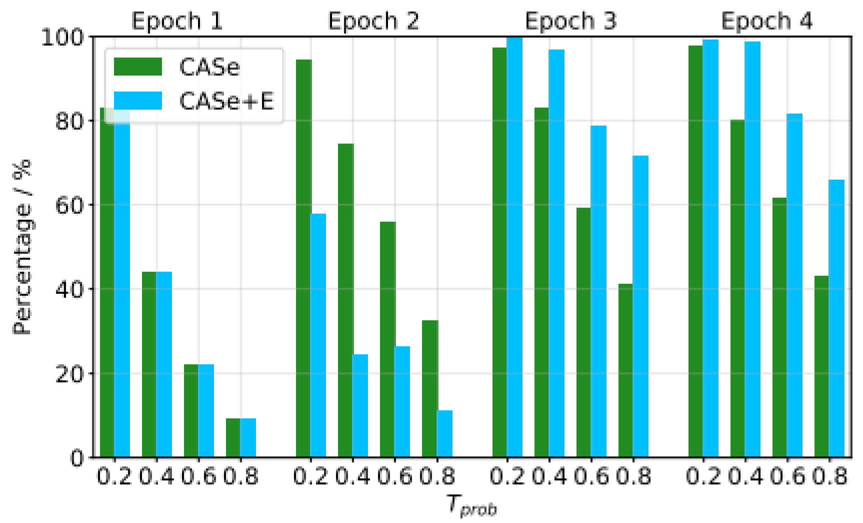
P_i : average performance of self model on dataset i.

P_{ij} : average performance of EM and F1 from source dataset i to target dataset j.

4 Evaluation



(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→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.

5 Conclusion

- BERT model cannot generalize well between different dataset (corpora & question form)

- Self-training & conditional adversarial learning method generalizes well on target domain dataset

- Evaluated well on ubiquitous generalization problem on Question Answering Task