Microsoft

Cross-domain Speech Recognition with Unsupervised Character-level Distribution Matching

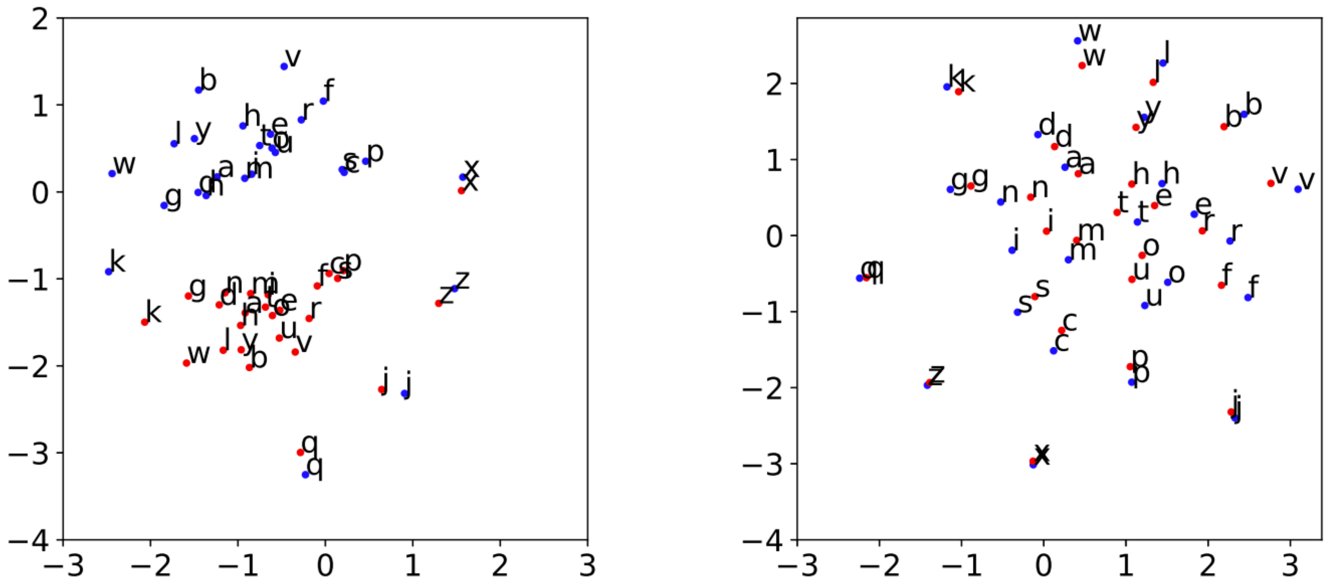
Wenxin Hou[1], Jindong Wang[2], Xu Tan[2], Tao Qin[2], Takahiro Shinozaki[1]
[1] Tokyo Institute of Technology
[2] Microsoft Research Asia

Introduction

- Background
  - Distribution mismatch leads to deterioration in automatic speech recognition (ASR)
  - Example: cross-device, cross-environment ASR
  - It is expensive and time-consuming to collect labeled speech data from massive domains (distributions)
- Unsupervised Domain Adaptation (UDA)
  - Existing methods
    - Data augmentation + representation matching
    - Self-training with pseudo-label filtering approach based on the model's uncertainty using dropout
    - Domain-adversarial training
  - Limitation
    - Ignoring the fine-grained knowledge (characters, phoneme, and word) may result in unsatisfying results

CMatch

- Character-level distribution matching
  - $P(y|X)$
  - Why not word or utterance matching?
    - Word or utterance are highly sparse
    - No segmentation ground-truth in end-to-end ASR models



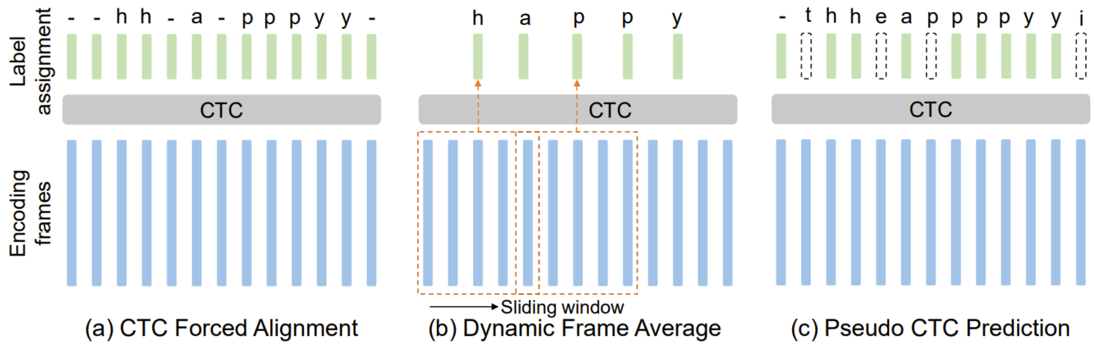
(a) Before CMatch (b) After CMatch

Preliminary

- CTC-Attention Transformer ASR Model
  - Input: 83-dimensional filter banks with pitch features (10 ms frame shift, 25 ms frame length)
  - Network Structure:
    - 12 encoder Layers (self-attention, feed-forward)
    - CTC module: output CTC predictions
    - 6 decoder layers (self-attention, cross-attention, feed-forward)
  - Training: $\mathcal{L}_{\text{ASR}} = (1-\lambda)\mathcal{L}_{\text{ATT}} + \lambda\mathcal{L}_{\text{CTC}}$
  - Decoding: $\hat{Y} = \arg\max_{Y \in \mathcal{Y}} (1-\lambda) \log P_{\text{ATT}}(Y|X) + \lambda \log P_{\text{CTC}}(Y|X)$

CMatch: Character-level Distribution Matching

- Frame-level Label Assignment



- CTC forced alignment
  - Take the labels from the most probable path selected by CTC forward-backward algorithm as the frame-level assignment
  - Effective but computationally expensive
- Dynamic Frame Average
  - Assign frames for each character by sliding window averaging
  - Work in a strict condition that the character output is a uniform distribution
- Pseudo CTC Prediction
  - CTC model naturally predicts the label assignment frame by frame which can be directly utilized
  - Filter out the CTC predictions with a threshold 0.9 based on their softmax scores to improve the accuracy

$$\hat{Y}_n = \arg\max_{Y_n} P_{\text{CTC}}(Y_n|X_n), \quad 1 \leq n \leq N$$

Distribution Matching

- Maximum Mean Discrepancy (MMD)
  - MMD is a non-parametric criterion to empirically evaluate the divergence between two distribution
  - Formulation:

$$\mathrm{MMD}(\mathcal{H}_k, P, Q) = \sup_{||\phi||_{\mathcal{H}_k} \le 1} \mathbb{E}_{X_S \sim P} \phi(X_S) - \mathbb{E}_{X_T \sim Q} \phi(X_T)$$

  - Biased empirical estimate:

$$\mathrm{MMD}(\mathcal{H}_k, X_S, X_T) =$$
$$\sup_{||\phi||_{\mathcal{H}_k} \le 1} \left(\frac{1}{|X_S|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \phi(x_t) \right)$$

- Character-level Distribution Matching Loss

$$\mathcal{L}_{\text{cmatch}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \mathrm{MMD}(\mathcal{H}_k, X_S^c, X_T^c)$$

Learning Algorithm

- Overall Loss

$$\mathcal{L} = \frac{1}{2}\left(\mathcal{L}_{\text{ASR}}^{\text{src}} + \mathcal{L}_{\text{ASR}}^{\text{tgt}}\right) + \gamma \mathcal{L}_{\text{cmatch}}$$

- Learning algorithm

Algorithm 1 Learning algorithm of CMatch

Input: Source domain (X_S, Y_S), target domain X_T.

1: Train network M_S on source domain (X_S, Y_S).
2: Obtain pseudo label $\hat{Y}_T$ with M_S.
3: **while** not done **do**
4:     Obtain the frame-level labels.
5:     Joint optimization using the overall loss
6: **end while**
7: **return** Adapted model $M_{S \to T}$ and target transcripts.

Experimental Setup

- Dataset: Libri-Adapt
  - Cross-device: Matrix Voice (M), PlayStation Eye (P), and ReSpeaker (R)
  - Cross-environment: clean, rain, wind, laughter
  - Number of utterances (hours)
    - Training: 25685 (93.77)
    - Validation: 2854 (10.71)
    - Testing: 2600 (5.60)
- Baselines
  - Source-only
  - MMD-ASR
  - Domain Adversarial Training (ADV)

Cross-domain Adaptation Results

- In-domain

Domain	WER
Matrix Voice (M)	24.25
PlayStation Eye (P)	20.07
ReSpeaker (R)	23.78
Average	22.70

- Device Adaptation

Task	Source-only	MMD	ADV	CMatch
$M \rightarrow P$	23.87	20.87	21.11	**20.38**
$M \rightarrow R$	25.21	22.21	22.27	**21.77**
$P \rightarrow M$	31.15	27.22	28.29	**26.17**
$P \rightarrow R$	23.99	21.90	21.74	**20.43**
$R \rightarrow M$	32.45	28.27	29.95	**27.77**
$R \rightarrow P$	23.48	21.09	21.23	**20.58**
Average	26.69	23.59	24.10	**22.85**

14.39% improvement

- Noise Adaptation

Target	Source-only	MMD	ADV	CMatch
Rain	38.21	33.61	34.65	**32.90**
Wind	29.70	26.06	26.73	**23.12**
Laughter	33.36	29.85	30.41	**28.55**
Average	33.76	29.84	30.60	**28.19**

16.50% improvement

Additional Experiments

- Ablation Study
  - Both self-training and distribution matching are effective

Variant	Device	Noise
Source-only	26.69	33.76
w/ self-training	22.99	28.31
w/ distribution matching	23.87	30.43
All	22.85	28.19

- Analyzing the Label Assignment
  - Our pseudo method can be efficient and effective

Task	PseudoCTCPred	FrameAverage	CTCAlign
$M \rightarrow P$	20.38	20.21	20.23
$M \rightarrow R$	21.77	21.80	21.75
$P \rightarrow M$	26.17	26.02	25.84
$P \rightarrow R$	20.43	20.36	20.44
$R \rightarrow M$	27.77	27.94	27.73
$R \rightarrow P$	20.58	20.55	20.52
Average	22.85	22.81	22.75

- Adapting with Decoder
  - Decoder adaptation is not necessary

Target	w/o decoder	first	last	all
Rain	32.90	32.92	32.85	33.12
Wind	23.12	23.18	23.18	23.28
Laughter	28.55	28.66	28.56	28.63
Average	28.19	28.25	28.20	28.34

Summary

- We propose CMatch to match the character-level distributions from the source and target domain

- We empirically analyze the contribution of Transformer encoders and decoders as well as different label assignment strategies

- CMatch outperforms existing approaches on both device and noise adaptation tasks by leveraging the fine-grained information

Q & A