Improving noisy student training for low resource languages in E2E ASR using CycleGAN-inter-domain losses

Chia-Yu Li and Nogc Thang Vu

Institute of Natural Language Processing (IMS)
University of Stuttgart, Germany

SIGUL 2024 @ LREC-COLING 2024

Motivation

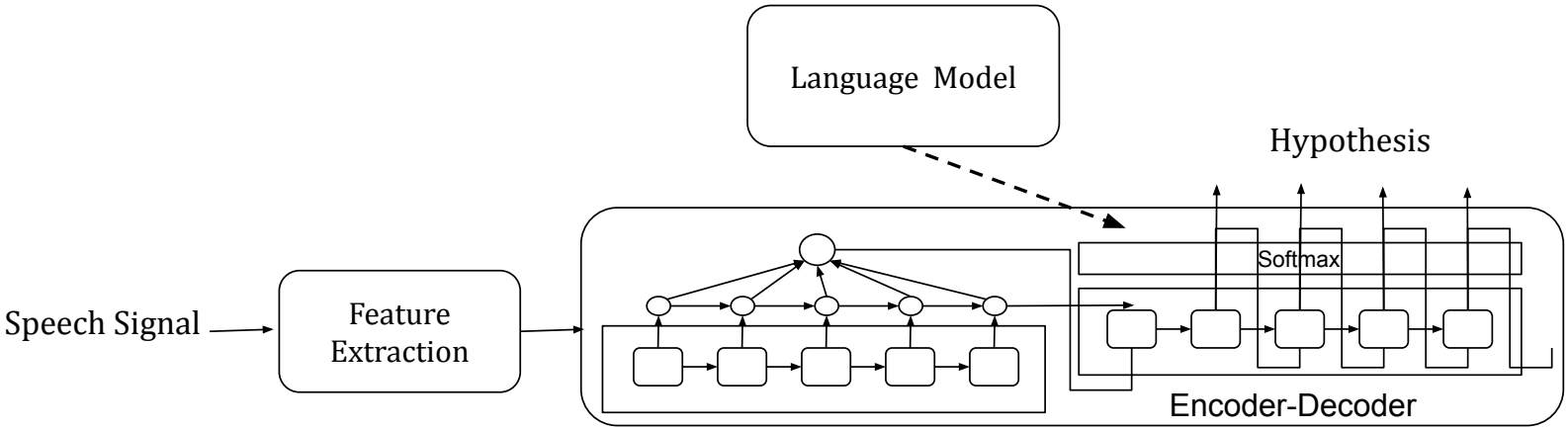


Figure: An architecture of End-to-End ASR (speech-to-text).

Motivation

- The biggest challenge in E2E ASR today is **quickly and cost-effectively** transferring speech processing systems to new languages with minimal manual effort.
- Semi-supervised learning can be applied to reduce the amount of labeled data.
  - small portion of labeled data
  - large portion of unlabeled data
- **Noisy student training (NST)**[1] is an simple and effective iterative self-training method that leverages unlabeled data to enhance accuracy.

[1] Daniel S. Park et al., "Improved Noisy Student Training for Automatic Speech Recognition," in Proc. of Interspeech, 2020

Motivation

- The biggest challenge in E2E ASR today is **quickly and cost-effectively** transferring speech processing systems to new languages with minimal manual effort.
- Semi-supervised learning can be applied to reduce the amount of labeled data.
  - small portion of labeled data
  - large portion of unlabeled data
- **Noisy student training (NST)**[1] is an simple and effective iterative self-training method that leverages unlabeled data to enhance accuracy.

[1] Daniel S. Park et al., "Improved Noisy Student Training for Automatic Speech Recognition," in Proc. of Interspeech, 2020

Motivation

- The biggest challenge in E2E ASR today is **quickly and cost-effectively** transferring speech processing systems to new languages with minimal manual effort.
- Semi-supervised learning can be applied to reduce the amount of labeled data.
  - small portion of labeled data
  - large portion of unlabeled data
- **Noisy student training (NST)**[1] is an simple and effective iterative self-training method that leverages unlabeled data to enhance accuracy.

[1] Daniel S. Park et al., "Improved Noisy Student Training for Automatic Speech Recognition," in Proc. of Interspeech, 2020.

Motivation

Let S denotes the labeled data and U denotes the unlabelled data.

Definition

Noisy student training (NST)

1. Train M_0 on S using SpecAugment. Set $M = M_0$.
2. Fuse M with LM and measure performance.
3. Generate labeled dataset $M(U)$ with fused model.
4. Filter generated data $M(U)$ to obtain $f(M(U))$.
5. Balance filtered data $f(M(U))$ to obtain $b\dot{f}(M(U))$.
6. Mix dataset $b\dot{f}(M(U))$ and S. Use mixed dataset to train new model M_0 with SpecAugment.
7. Set $M = M'$ and go to 2.

- For low-resource languages, the NST is not effective due to M_0 performs poorly.
- How to improve inexpensively the teacher model in NST remains a key challenge especially for low-resource languages.

Motivation

Let S denotes the labeled data and U denotes the unlabelled data.

Definition

Noisy student training (NST)

1. Train M_0 on S using SpecAugment. Set $M = M_0$.
2. Fuse M with LM and measure performance.
3. Generate labeled dataset $M(U)$ with fused model.
4. Filter generated data $M(U)$ to obtain $f(M(U))$.
5. Balance filtered data $f(M(U))$ to obtain $b\dot{f}(M(U))$.
6. Mix dataset $b\dot{f}(M(U))$ and S. Use mixed dataset to train new model M_0 with SpecAugment.
7. Set $M = M'$ and go to 2.

- For low-resource languages, the NST is not effective due to M_0 performs poorly.
- How to improve inexpensively the teacher model in NST remains a key challenge especially for low-resource languages.

Motivation

Let S denotes the labeled data and U denotes the unlabelled data.

Definition

Noisy student training (NST)

1. Train M_0 on S using SpecAugment. Set $M = M_0$.
2. Fuse M with LM and measure performance.
3. Generate labeled dataset $M(U)$ with fused model.
4. Filter generated data $M(U)$ to obtain $f(M(U))$.
5. Balance filtered data $f(M(U))$ to obtain $b\dot{f}(M(U))$.
6. Mix dataset $b\dot{f}(M(U))$ and S. Use mixed dataset to train new model M_0 with SpecAugment.
7. Set $M = M'$ and go to 2.

- For low-resource languages, the NST is not effective due to M_0 performs poorly.
- How to improve inexpensively the teacher model in NST remains a key challenge especially for low-resource languages.

Contribution

- We observe that training a model by CycleGAN and inter-domain losses (CID)[2] with lots of external text significantly boosts performance.
- We enhance CID using automatic hyperparameter tuning and integrate it into the NST training pipeline for low-resource scenarios to boost the teacher model.
- The evaluation of our method on six languages on the Voxforge and Common Voice demonstrate a 10% WERR compared to the baseline model.

[2] Li, C.-Y. and Vu, N. T., "Improving Semi-supervised End-to-end Automatic Speech Recognition using CycleGAN and Inter-domain Losses," in Proc. of SLT", 2023.

Contribution

- We observe that training a model by CycleGAN and inter-domain losses (CID)[2] with lots of external text significantly boosts performance.
- We enhance CID using automatic hyperparameter tuning and integrate it into the NST training pipeline for low-resource scenarios to boost the teacher model.
- The evaluation of our method on six languages on the Voxforge and Common Voice demonstrate a 10% WERR compared to the baseline model.

[2] Li, C.-Y. and Vu, N. T., "Improving Semi-supervised End-to-end Automatic Speech Recognition using CycleGAN and Inter-domain Losses," in Proc. of SLT", 2023.

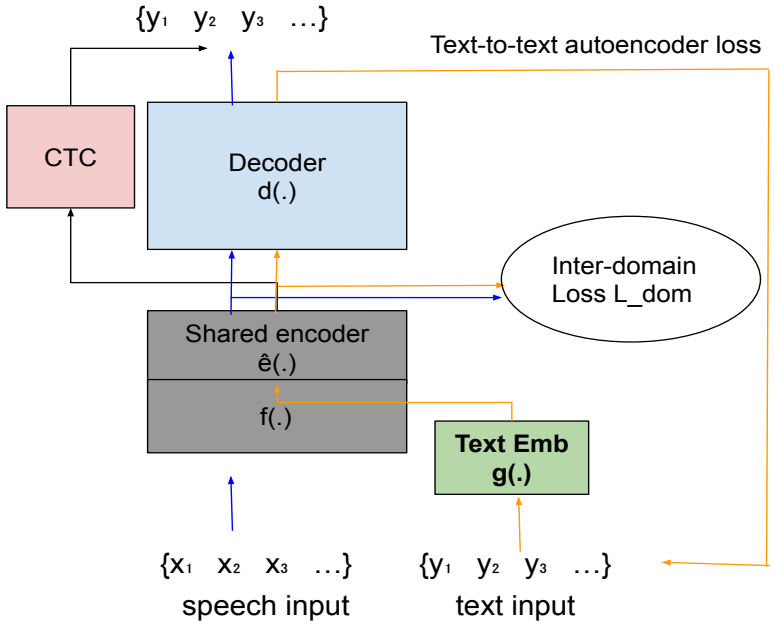
Contribution

- We observe that training a model by CycleGAN and inter-domain losses (CID)[2] with lots of external text significantly boosts performance.
- We enhance CID using automatic hyperparameter tuning and integrate it into the NST training pipeline for low-resource scenarios to boost the teacher model.
- The evaluation of our method on six languages on the Voxforge and Common Voice demonstrate a 10% WERR compared to the baseline model.

[2] Li, C.-Y. and Vu, N. T., "Improving Semi-supervised End-to-end Automatic Speech Recognition using CycleGAN and Inter-domain Losses," in Proc. of SLT", 2023.

Illustration of CID

- Semi-supervised E2E[3]



$$L_{semi} = \alpha L_{pair} + (1-\alpha) L_{unpair}$$

$$L_{pair} = -\sum_{(x,y)\in S} \log \prod_{t=1}^{|y|} \Pr(y_t | y_{t-1}, e(x))$$

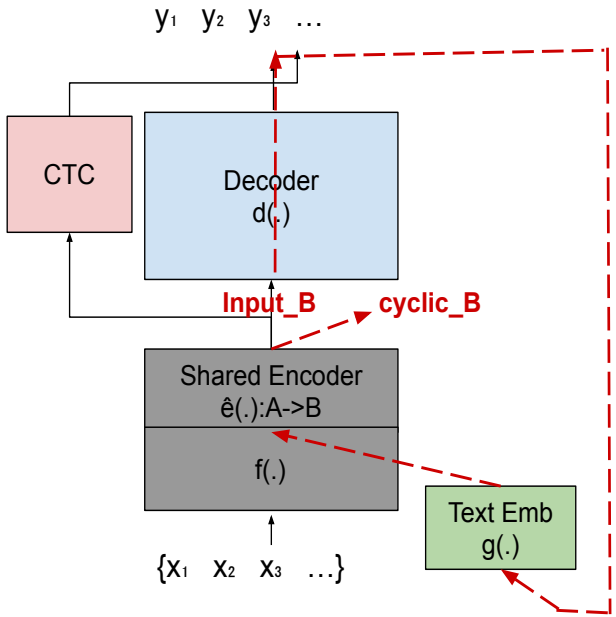
$$L_{unpair} = \beta * L_{dom} + (1-\beta) * L_{text}$$

- Our previous work (CID) adapted L_{unpair} to as follows,

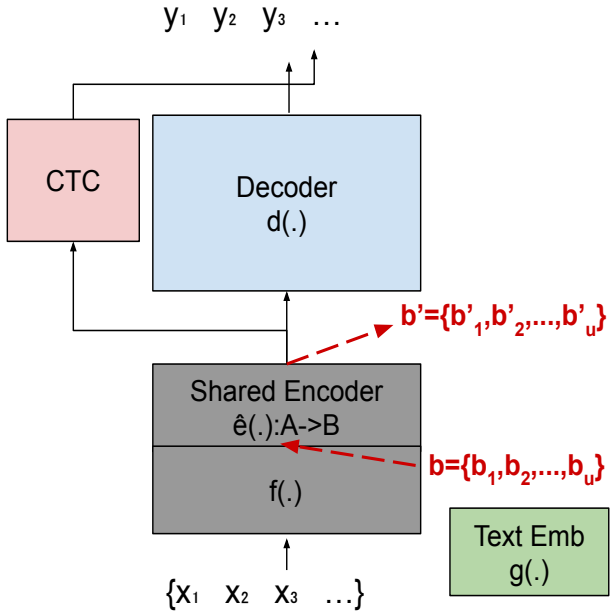
$$L_{unpair} = \beta \mathbf{L_{cyc,dom}} + \mathbf{L_{idt}} + (1-\beta) * L_{text} \quad (1)$$

[3]Shigeki Karita et al., "Semi-Supervised End-to-End Speech Recognition," In Proc. of Interspeech, 2018.

Illustration of CID



(a) The cycle-consistent inter-domain loss.



(b) The identity mapping loss.

$$L_{cyc,dom} = \mathcal{D}(input\_B, cycle\_B) = \mathcal{D}(e(x), \hat{e}(g(d(e(x))))) \quad (2)$$

$$L_{idt} = \|\hat{e}(b) - b\|_1 \quad (3)$$

CID solely with external text

Model	**paired data** (hours)	**unpaired text** (#lines)	**without LM** WER(%)	**with LM** WER(%)
Initial model	5[4]	0	63.6	63.1
CID model	5[4]	10K[5]	38.6	36.3
CID model	5[4]	100K[5]	31.2	29.4
CID model	5[4]	300K[5]	30.8	29.1

Table: WERs on the Voxforg German test set.

[4] Voxforge German data from http://www.voxforge.org/

[5] D. Goldhahn et al., "Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages." in Proc. of LREC, 2012.

Enhanced CID

- Two hyperparameters in CID: supervised ratio α and speech-to-text ratio β

$$L_{semi} = \alpha L_{pair} + (1 - \alpha) L_{unpair}$$

$$L_{unpair} = L_{idt} + \beta L_{cyc,dom} + (1 - \beta) L_{text}$$

- We propose automatic hyperparameter tuning as follows:
  - We set α to 0.9 for the first three epochs and gradually decays after three epochs.
  - We integrate β into the training process by using minimal, maximal, average, or median operations on the unsupervised losses with β from 0.0 to 1.0.

Enhanced CID

- Two hyperparameters in CID: supervised ratio α and speech-to-text ratio β

$$L_{semi} = \alpha L_{pair} + (1 - \alpha) L_{unpair}$$

$$L_{unpair} = L_{idt} + \beta L_{cyc,dom} + (1 - \beta) L_{text}$$

- We propose automatic hyperparameter tuning as follows:
  - We set α to 0.9 for the first three epochs and gradually decays after three epochs.
  - We integrate β into the training process by using minimal, maximal, average, or median operations on the unsupervised losses with β from 0.0 to 1.0.

Enhanced CID

Model	α	**adapted** L_{unpair}	**CER(%)**
Baseline[6]	0.5		46.9
MIN-UNPAIR-LOSS	0.5	$\min_{\beta\in[0,1.0]} \mathcal{L}_{unpair}$	30.6
MAX-UNPAIR-LOSS	0.5	$\max_{\beta\in[0,1.0]} \mathcal{L}_{unpair}$	39.5
AVG-UNPAIR-LOSS	0.5	$\overline{\mathcal{L}_{unpair}}$	50.6
MED-UNPAIR-LOSS	0.5	Median($\mathcal{L}_{unpair}$)	50.4
DECAY-MIN-UNPAIR-LOSS	decay	$\min_{\beta\in[0,1.0]} \mathcal{L}_{unpair}$	29.6
DECAY-MAX-UNPAIR-LOSS	decay	$\max_{\beta\in[0,1.0]} \mathcal{L}_{unpair}$	44.1
DECAY-AVG-UNPAIR-LOSS	decay	$\overline{\mathcal{L}_{unpair}}$	46.6
DECAY-MED-UNPAIR-LOSS	decay	Median($\mathcal{L}_{unpair}$)	30.3

Table: The models' CERs on the Common Voice Finnish test set.

[6] C.-Y. Li and T. Vu,"Improving Semi-supervised End-to-end Automatic Speech Recognition using CycleGAN and Inter-domain Losses," in Proc. of SLT, 2022.

Enhanced CID

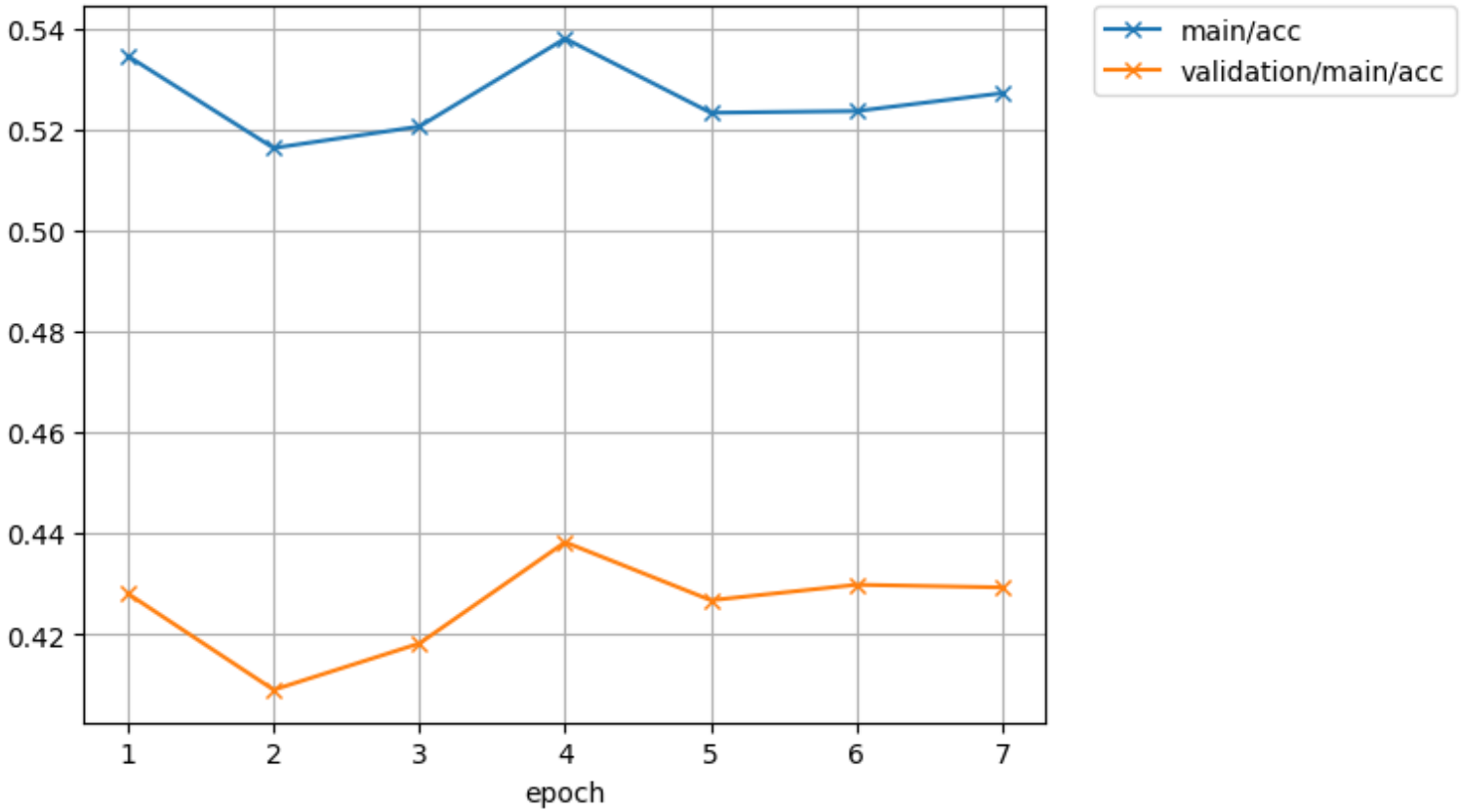


Figure: The Baseline (CID) accuracy on train/valid set.

Enhanced CID

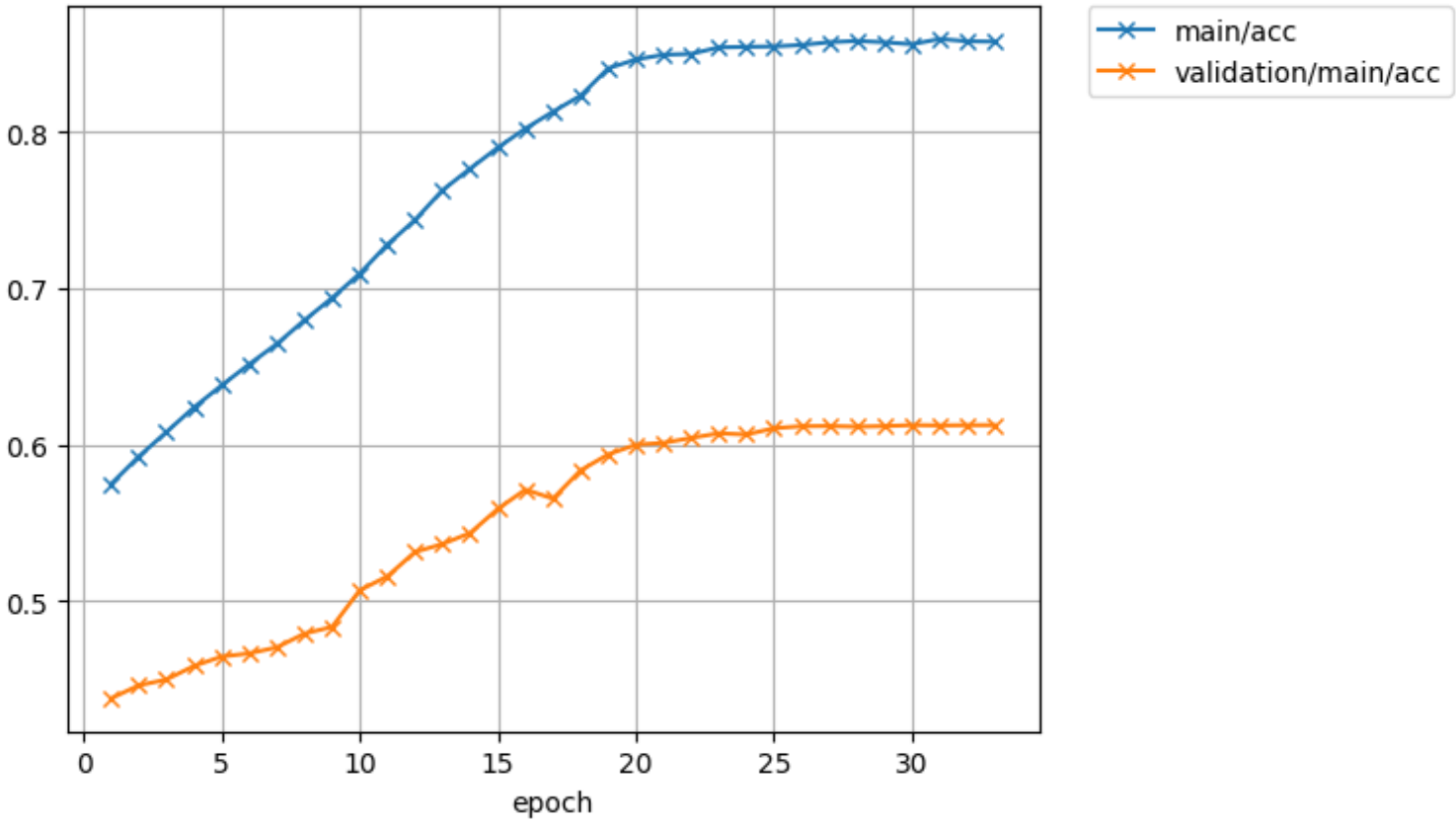


Figure: The MIN-UNPAIR-LOSS accuracy on train/valid set.

Proposed Method: NST with enhanced CID (cNST)

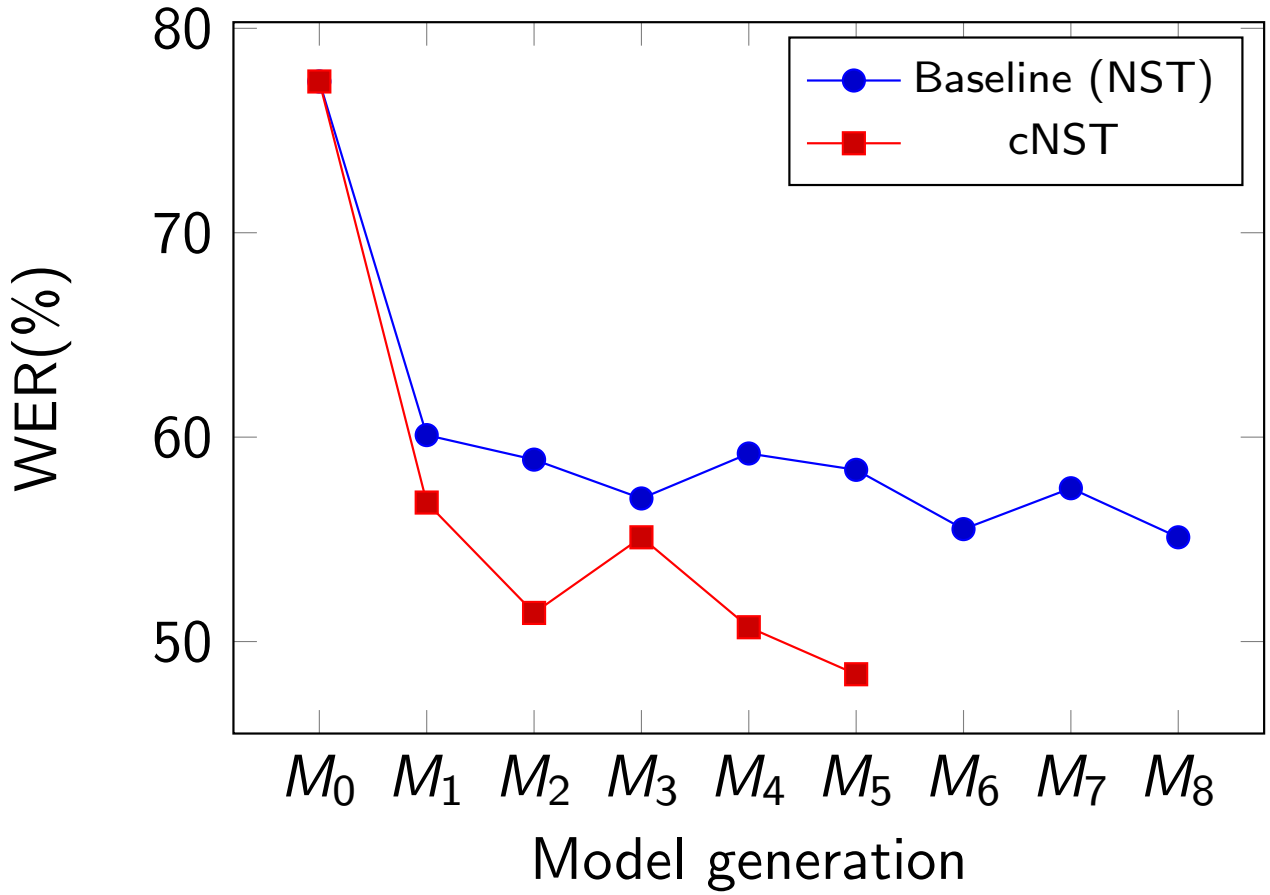
Definition

1. Train M_0 on S using SpecAugment. Set $M = M_0$.
2. **Train M_1 on S and $U = \{Y'\}$ by enhanced CID and using SpecAugment. Set $M = M_1$.**
3. Fuse M with LM and measure performance.
4. Generate labeled dataset $M(U)$ with fused model.
5. ~~Filter generated data $M(U)$ to obtain $f(M(U))$.~~
6. ~~Balance filtered data $f(M(U))$ to obtain $b \times f(M(U))$.~~
7. Mix dataset $M(U)$ and S. Use mixed dataset to train new model M_0 with SpecAugment.
8. Set $M = M'$ and go to 3.

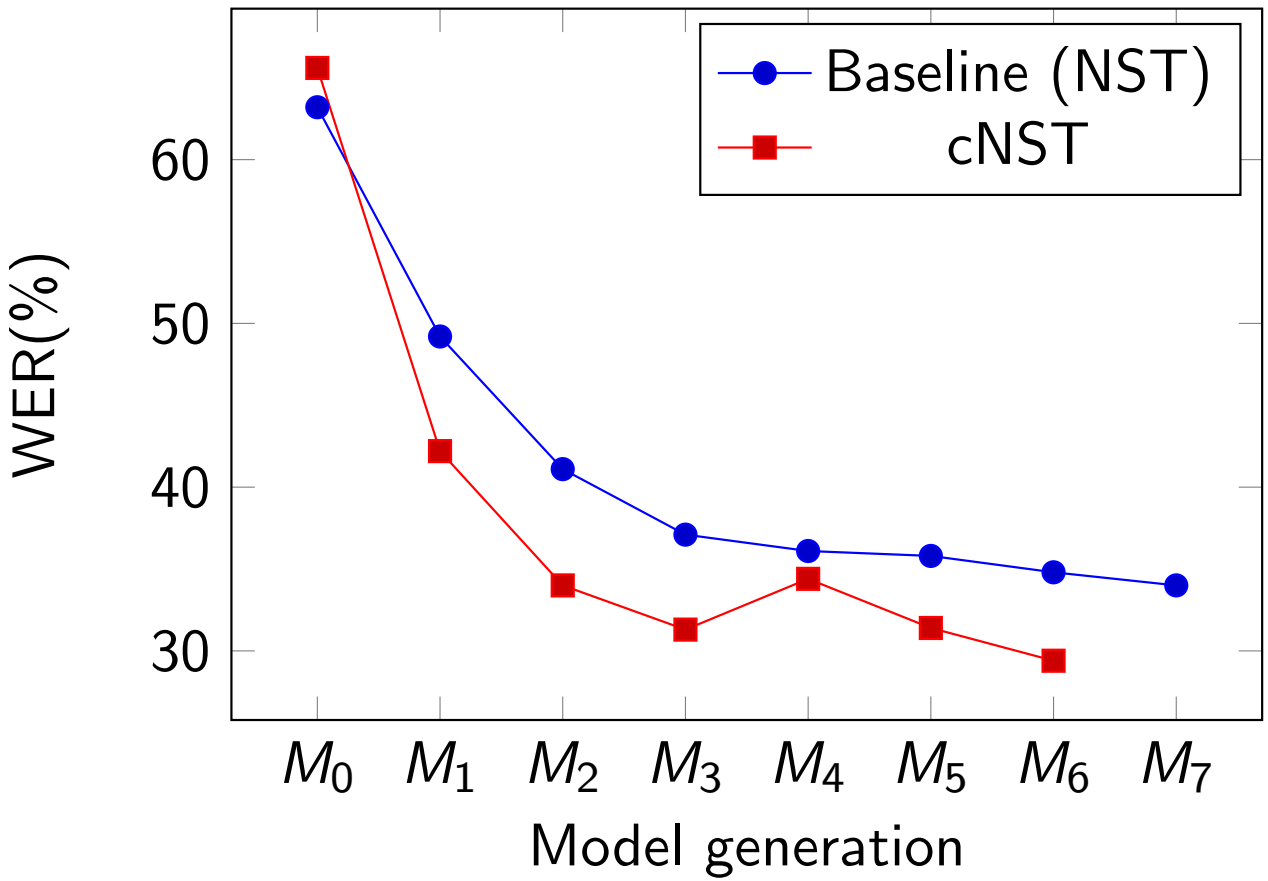
Experimental Setup

- Dataset
  - Common Voice: it is a massively multilingual collection of transcribed speech recorded by user on Mozilla website. We choose European languages which has limited data.
  - Voxforge: it consists of user submitted audio clips using their own microphone and eight European languages.
- External text: Leipzig corpus consists of annual collections of documents from various sources such as wikis, news, and the website.
- The system is implemented under Espnet and is trained by following the default recipe.

WERs against model generation (Finnish language)



WERs against model generation (Greek language)



cNST effectiveness across corpus

Model	**Voxforge (WER%)**			**Common Voice (WER%)**		
	German	Italien	Dutch	Hungarian	Finnish	Greek
Initial Model	63.1	71.2	63.1	84.8	77.4	63.2
NST	49.7	47.1	58.2	72.0	55.1	34.0
cNST	**27.3**	**42.0**	**56.3**	**58.6**	**48.4**	**29.4**
WERR	45.1	10.8	3.26	18.6	12.7	13.5

Table: The best student models' WERs comparison.

Recognition output

Models	WER(%)	INS	DEL	SUB
Initial Model	63.1	1.8	20.6	40.7
NST	49.7	1.0	21.0	27.9
CID	9.4	3.3	4.0	22.0
cNST	**27.3**	**3.2**	**3.6**	**20.5**

Table: The WER, insertion, deletion, and substitution at word level on the Voxforge German test set.

Conclusion

- We observe that training the model by CID with lots of external text significantly boosts performance.
- We enhance CID by incorporating automatic hyperparameter tuning.
- We propose to improve the NST training pipeline for low-resource languages by leveraging enhanced CID.
- The experimental results show that our propose method accelerates the iterative self-training process and demonstrate the effectiveness across six European languages from two datasets, surpassing the baseline by 10% WERR.

Conclusion

- We observe that training the model by CID with lots of external text significantly boosts performance.
- We enhance CID by incorporating automatic hyperparameter tuning.
- We propose to improve the NST training pipeline for low-resource languages by leveraging enhanced CID.
- The experimental results show that our propose method accelerates the iterative self-training process and demonstrate the effectiveness across six European languages from two datasets, surpassing the baseline by 10% WERR.

Conclusion

- We observe that training the model by CID with lots of external text significantly boosts performance.
- We enhance CID by incorporating automatic hyperparameter tuning.
- We propose to improve the NST training pipeline for low-resource languages by leveraging enhanced CID.
- The experimental results show that our propose method accelerates the iterative self-training process and demonstrate the effectiveness across six European languages from two datasets, surpassing the baseline by 10% WERR.

Conclusion

- We observe that training the model by CID with lots of external text significantly boosts performance.
- We enhance CID by incorporating automatic hyperparameter tuning.
- We propose to improve the NST training pipeline for low-resource languages by leveraging enhanced CID.
- The experimental results show that our propose method accelerates the iterative self-training process and demonstrate the effectiveness across six European languages from two datasets, surpassing the baseline by 10% WERR.

Thanks for your attention!