Interspeech 2024
1-5 September 2024, Kos, Greece

Self-Train Before You Transcribe

Robert Flynn, Anton Ragni

Department of Computer Science, The University of Sheffield, United Kingdom
{rjflynn2, a.ragni}@sheffield.ac.uk

Abstract

When there is a mismatch between the training and test domains, current speech recognition systems show significant performance degradation. Self-training methods, such as noisy student teacher training, can help address this and enable the adaptation of models under such domain shifts. However, self-training typically requires a collection of unlabelled target domain data. For settings where this is not practical, we investigate the benefit of performing noisy student teacher training on recordings in the test set as a test-time adaptation approach. Similarly to the dynamic evaluation approach in language modelling, this enables the transfer of information across utterance boundaries and functions as a method of domain adaptation. A range of in-domain and out-of-domain datasets are used for experiments demonstrating large relative gains of up to 32.2%. Interestingly, our method showed larger gains than the typical self-training setup that utilises separate adaptation data.
Index Terms: speech recognition, long-context, self-attention

1. Introduction

Speech recognition models are usually trained on a collection of speech and text data, then remain fixed at inference/test time when they are used to transcribe new speech. As a consequence, the model holds a fixed prior, formed from the training data, on the distribution of words and speech for every new utterance that is processed. This fixed prior becomes problematic as the change in distribution between the training data and the inference data becomes larger, resulting in reduced performance [1].

Prior work [2, 3, 4, 5] has shown that self-training methods, such as pseudo-labelling, [6] can be used to adapt models under domain shifts. This typically involves training an initial *teacher* model on a source domain, which is then used to produce the target labels for a *student* model on the target domain. While these methods do not require labelled data, they usually assume that a collection of unlabelled data from the target domain is available. Recent work on test time adaptation [7] (TTA) proposes to instead adapt the model solely at test time, without the need for a separate adaptation set. This is advantageous when the target domain is not known in advance, or unlabelled data is either not available or cannot be shared due to privacy concerns. Additionally, target domain data collection can be expensive, and may need to be repeated over time due to domain drift.

Pseudo-labelling can also be used as a TTA method for automatic speech recognition (ASR) [8]. For this, the pseudo-labelling-based adaptation is performed solely on the current recording, prior to transcribing it. This process also allows for the transfer of information across utterance boundaries via gradient descent, relaxing the independence assumption that is usually made when segmenting a recording into separate utterances. We draw parallels to the dynamic evaluation approach [9, 10] that has proved effective in language modelling, where training models on the previous history enables them to better exploit recurring patterns, that occur outside the effective context window. Extensions to pseudo-labelling such as noisy student teacher (NST) training [11, 12, 3], where noise/augmentation is added to the inputs of the student model to make prediction more challenging, have proven effective for self-training approaches. Hence, in this work, we investigate the use of NST at test time as a method of TTA. A breakdown of our main contributions is given as follows:

1. We propose (§ 2) a method that substantially improves over prior work, demonstrating that NST is more effective than standard pseudo-labelling for TTA. We find (§ 5.4) that the use of augmentation is particularly crucial when the domain mismatch is large.
2. We show (§ 5.3) that our method leads to better performance, while using $100\times$ less data, than a more standard self-training approach that uses separate adaptation data.
3. We show (§ 5.5) that the method improves with longer recordings and that the local context is most useful for the adaptation.

The model checkpoint and code to reproduce our results are made available here[1].

2. Method

The following section presents our method of TTA, which we refer to as Noisy Student Teacher at Inference (NSTI). This is depicted in figure 1, with algorithm 1 covering the process that is used to transcribe recordings with NSTI. For this work CTC [13] based acoustic models are used.

For NSTI 2 spectrograms X and $X^{'}$, where $X^{'} = \mathrm{Transformation}(X)$, are used as inputs to a teacher model M and a student model $M^{'}$ which are identical and always share the same parameters. The Transformation used to produce $X^{'}$ can be an augmentation strategy such as SpecAugment [14]. Output probabilities $P = M(X)$ and $P^{'} = M^{'}(X^{'})$ are obtained from the models. The models are then updated based on the loss between $P^{'}$ and a decoded label sequence Y^{*} that is produced from decoding P. Practically, as the student and teacher models are identical, this method only requires one forward and backwards pass per step through a single model.

For transcribing recordings using our method, the following process is used: first, the recording is segmented. Self-training is then performed on each segment in the recording in a random

[1] www.github.com/robflynnyh/Self-Train-Before-You-Transcribe

10.21437/Interspeech.2024-858

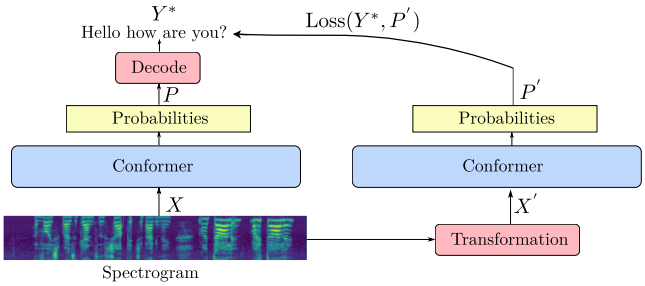

Figure 1: *Depiction of the NSTI method*

Algorithm 1 Noisy Student Teacher at Inference (NSTI)

Require: Recording R, model M, number of epochs n
Segment R into $S_1, S_2, \ldots, S_m$
Shuffle the segments $S_1, S_2, \ldots, S_m$
for $i = 1$ to n **do**
  for each segment S_j **do**
    $X = S_j$
    Apply Transfomation: $X^{'} = \text{Transformation}(X)$
    Compute probabilities: $P = M(X)$, $P^{'} = M^{'}(X^{'})$
    Decode P to obtain target labels: $Y^* = \text{Decode}(P)$
    Obtain gradients using the loss between $P^{'}$ and Y^*
    Update M and $M^{'}$ using the gradients
  end for
end for
Obtain predictions for recording R using the updated model M

order, and repeated for n epochs. A final pass is then performed over the recording to obtain the model's predictions. Note that the method is performed over individual recordings and not the Dev/Test set as a whole.

2.1. Transformations

For the transformation function, the frequency masking component of SpecAugment [14] is used for our primary experiments. The use of time masking was not found to be beneficial. In § 5.4 we also experiment with a range of other transformations, which are as follows: Identity transform, where no change is applied; Random noise, where Gaussian noise is added to the spectrogram; CutOut [15], where n rectangles are masked out from the spectrogram.

3. Prior Work

Self-training methods can be used for the purpose of adaptation [2, 3, 4, 5], these methods typically use a separate training set to adapt a model to a target domain. In contrast, TTA methods use only the data available at test/inference time, such as the current recording or utterance. Entropy minimisation (EM), originally presented as a semi-supervised learning approach [16], has proven effective as a method of TTA for computer vision [7] and ASR [17]. The Single-Utterance Test-time Adaptation (SUTA) [17] is an example of a TTA approach for ASR that uses an EM-based technique. This method adapts models on a single utterance and then discards the adapted model for the following utterance. While SUTA is appropriate in an $i.i.d$ scenario, recordings containing multiple utterances with highly correlated features are present at test time in most scenarios. Therefore, to improve performance our work leverages the entire recording, rather than a single utterance.

To benefit from updates from preceding utterances, AWMC [8], adopts a psuedo-labelling approach towards TTA. This method focuses on an online scenario where utterances are trained on and transcribed in their natural order. Due to problems with model collapse, AWMC uses 2 models to serve as the teacher model, with weights that are an exponential moving average (EMA) of the students. While [8] investigates TTA for an online setting where only prior utterances are available, our work focuses on an offline approach that also uses future utterances for improved performance. Additionally, the use of augmentation is not explored in [8]. In our work, we find that the use of an EMA teacher is not needed to prevent model collapse. This may be due to a more stable base model and the use of augmentation. Attempts to include an EMA teacher also resulted in reduced performance due to the teacher adapting at a slower rate, however in scenarios where model collapse does occur this would be beneficial.

Other work in natural language processing, also investigates a form of TTA [18] where pseudo-labelling is used at test-time to adapt the model via gradient descent. In this work, low-quality pseudo-labels are filtered based on confidence, and the model is trained to predict the pseudo-labels while maintaining similarity to the pre-trained weights. Other work, in language modelling presents a technique referred to as dynamic evaluation [9, 10], where gradient descent is used to update the model based on the prior text history. This enables the model to exploit recurring patterns in the sequence, which occur outside the effective context window. However, it generally relies on the availability of a ground truth text history.

4. Experimental Setup

4.1. Datasets

For training the initial pre-adapted baseline model, the collection of Spotify podcasts totalling 58K hours provided in [19] is used. For the experiments in this work, a range of in-domain and out-of-domain datasets are included in order to properly evaluate our method, which are as follows: **Earnings-22** [20], which consists of earnings report meetings with a diverse range of accents. The train/dev/test splits from [21] are used in this work. Each meeting lasts up to 2 hours in length, with 5.5/5.6 in the dev and test sets respectively. **Rev16**, which is a collection of 30 podcast episodes, we use the 16 episodes reported on in [22]. Rev16 can be viewed as our in-domain test set due to its similarity to our training data. **Tedlium** [23], which is a collection of 10-20 minute TED talks, typically involving a single speaker. **Chime6** [24], which is used as a highly out-of-domain test set, we use only the first distant microphone array from this data, resulting in large amounts of background noise. Channels of the first array are combined by averaging the spectrograms prior to normalisation.

4.2. Model Configuration

The model uses a Conformer [25] based architecture, with the Fast Conformer subsampling configuration [26]. For our method, we find it essential to replace batch normalisation [27] with batch renormalization [28] in the conformer convolution

modules, and do not update the batch statistics during the self-training process. A similar approach is taken in [5], where group normalisation is used, however, we find this to be unstable during the initial training stage. The model consists of 6 layers with roughly 90 million parameters. Training and adaptation is performed with a context window/segment length of 162 seconds.

4.3. Hyperparameters

All hyperparameters (i.e. augmentation, epochs, learning rate) are tuned using a random search on the development set of Tedlium for our primary experiements. Dataset-specific tuning is also explored in § 5.2. All experiments use a batch size of 2, composed of 2 copies of a single segment/utterance. Madgrad [29] is used as the optimizer. For our primary results that use SpecAugment the following hyperparameters are used: 6 frequency masks with a maximum size of 34, 5 epochs and a learning rate of $9e-5$.

For segmenting recordings the sliding window scheme described in [30] is used. Specifically, the recording is chunked into segments/windows equal to the model's context window using a moving window with a stride of 12.5% of the segment length. When transcribing the recording, probabilities from segments that overlap are averaged to obtain the final prediction. However, this process is not essential to the method, and standard utterance boundaries can also be used.

5. Experimental Results

All evaluations reported in the tables are repeated 3 times, with mean statistics reported. We find the variation between repeats to be low, with a standard deviation of around $0.01 - 0.1$. The following subsections will break down and discuss our findings.

5.1. How effective is the method?

Table 1: *WERs (Dev/Test) when using various training settings. *Method from prior work [8].*

Setting	Aug	TED	E-22	Chime6	Rev16
Shuffled	✓	**6.6/5.8**	**18.9/14.9**	**56.6/59.4**	**14.2**
Ordered	✓	**6.6**/5.9	19.5/15.4	57.3/61.8	**14.2**
Online	✓	6.9/6.1	21.7/16.7	59.7/64.7	**14.2**
AWMC*	✗	7.0/6.2	23.4/18.1	88.9/85.3	15.2
AWMC	✓	6.8/6.0	20.7/15.7	70.7/75.9	**14.2**
Unadapted model	N/A	7.1/6.2	23.9/18.3	83.5/86.5	14.5

Table 2: *Real time factor of each setting*

Setting	Aug	1 Epoch	Total
Shuffled	✓	0.027	0.115
Ordered	✓	0.027	0.115
Online	✓	0.023	0.023
AWMC	✗	0.026	0.026
AWMC	✓	0.026	0.097
Unadapted model	N/A	N/A	0.004

Table 1 presents the results for our method using various settings. Settings with augmentation (Aug), use the frequency masking component of SpecAugment [14] The real-time factors (RTF) for each setting are provided in table 2. **Bold** text denotes the best results, red text denotes that the model performs worse after test-time adaptation.

The primary method described in algorithm 1 is referred to as the Shuffled setting. The results for this setting demonstrate substantial gains over the unadapted model with word error reductions (WERRs) of 6.5%, 18.6%, 31.3% and 2.0% for Tedlium, Earnings-22 (E-22), Chime6 and Rev16 respectively. Note that our un-adapted baseline already shows competitive performance [23, 21, 1, 22] due to the large and diverse training dataset. Whereas prior TTA work in ASR [8, 17] uses a baseline trained on Librispeech [31], which consequently shows unrealistic gains from adaptation due to the narrow training domain.

In general, the datasets that are the most different from the training domain see the largest gains. Recordings in Tedlium are also shorter which results in less data for adaptation and therefore lower gains (see § 5.5). Results for Chime6 show that the method remains effective and stable even when the domain mismatch is very large and transcription quality is extremely poor. Rev16, our in-domain test set shows the smallest gains. We hypothesise that this is due to the similarity of this data to our large training dataset.

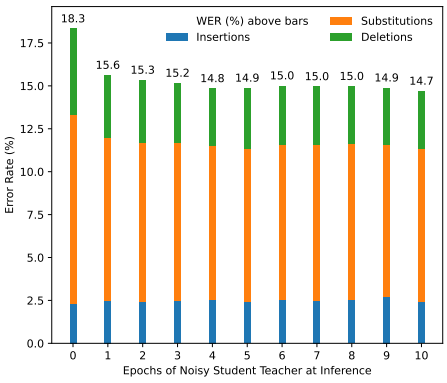

Figure 2: *Error rates after each epoch of NSTI on Earnings-22*

Word error rate (WERs), substitution, insertions and deletion statistics after each epoch of NSTI on Earnings-22 (test) are provided in figure 2. We find that the majority of the benefit from the method is attained after the first epoch, with a WERR of 14.8%. Therefore, adaptation can be stopped here if RTF is an issue. The method benefits from a reduction in the number of substitutions and deletions. Insertions see a small increase of 4.4% by epoch 5, with deletions showing the largest reduction of 30.3%. On the Chime6 dataset, the model is initially unconfident, with a very high deletion rate of 83.8%, therefore the method is very beneficial for this data, with the deletion rate reducing to 37.2% after NSTI.

While shuffling the data during NSTI is helpful, results from the Ordered setting demonstrate that it is not necessary for good performance. We also present an online setting where utterances are processed in their natural order, and the final predictions are taken from the teacher model at each time-step. This shows worse performance as it can not leverage future utterances, and only runs for one epoch. The AWMC method [8] shows a small WERR of 2.1%/1.1% on Earning-22, with the model degrading on Chime6 and Rev16. This is primarily due to not incorporating augmentation, as including augmentation causes the results to improve. As AWMC is an online method the use of this approach with augmentation may be preferable to the online variation of NSTI. However, the WER is higher than other settings on Chime6, which may be due to the EMA teacher model updating at a slower rate.

5.2. Dependency on hyperparameters

The hyperparameters used in the results are tuned on the Tedlium development split. We also experiment with dataset-specific tuning. On Earnings-22 there was no change in results when tuning was performed on the datasets Dev set. For

Chime6 there was a small further WERR of 2.3%/4.0%. This demonstrates that the method is not overly sensitive to hyperparameter choice, and does not require repeated re-tuning when the test domain changes. Which is important for scenarios where NSTI is likely to be useful.

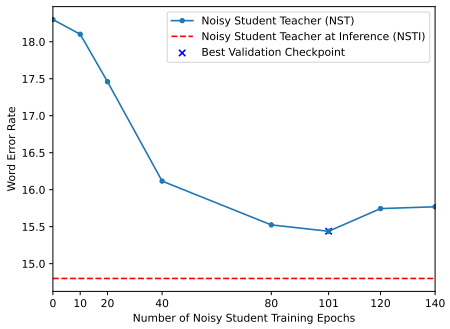

Figure 3: *Comparison between NST and NSTI on Earnings-22*

5.3. Comparison to self-training on a separate training set

We provide comparisons to an NST self-training approach that uses the 105-hour Earnings-22 training set to adapt the model in figure 3. For the NST training we use the method described in [3]. Results show that our method (NSTI) improves over NST by 4% when selecting the best checkpoint on the validation data. During NSTI only the current recording with a duration of around 1 hour is used, hence our method required 100× less data. Performing NSTI after adapting the model on the training set did not result in any further gains. We believe this is due to the sharpening of the model's class distribution after NST training.

Self-training on separate adaptation data may still be preferred in scenarios where the RTF is an issue and the target domain is known and constant. This is because NSTI needs to be performed on every recording. For example, adapting on a recording from Earnings-22 and then evaluating on other recordings from the same test set led to a WERR of -3.9% after the first epoch, and -17.4% after the fifth.

5.4. Comparison of transformation functions

Table 3: *WERs when using various transformation functions.*

Transform	TED	E-22	Chime6	Rev16
SpecAugment	6.6/**5.8**	**18.9**/14.9	56.6/59.4	**14.2**
Identity	7.0/6.1	22.6/17.4	100.0/100.0	15.0
Noise	6.6/5.9	24.2/19.2	99.6/97.3	18.5
CutOut	**6.4/5.8**	**18.9/14.5**	**54.9/56.9**	37.9
Unadapted model	7.1/6.2	23.9/18.3	83.5/86.5	14.5

A comparison of the different transformation functions described in § 2.1 is provided in table 3. The identity transform (no transformation) shows the worst performance, with small improvements on Tedlium/Earnings-22 and worse performance than the unadapted model on Chime6/Rev16. This explains the poor performance of AWMC [8], which uses this transformation. We find that the use of augmentation is especially crucial for datasets, such as Chime6, where the deletion rate is high, otherwise, the model learns to only output blank tokens.

Using random noise as the transformation function resulted in degradation on all datasets other than Tedlium. Cutout [15] showed the best performance on all datasets other than Rev16, where the model performed much worse than the unadapted model. The degradation seen on various datasets for the Identity, Noise, and Cutout transforms is because these methods were more sensitive to hyperparameter choices (which were tuned on Tedlium). Overall, the frequency masking component of SpecAugment showed the most consistent performance and outperformed the unadapted model by a large margin on all datasets.

5.5. Impact of recording duration

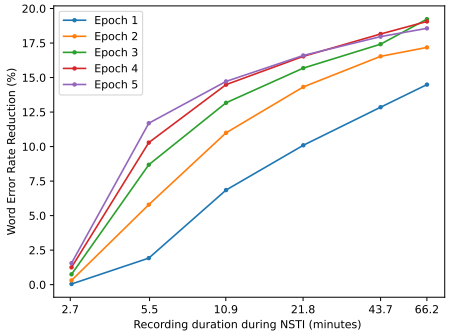

Figure 4: *WER as the recording duration for NSTI is increased*

Figure 4 presents results analysing the effect of recording duration on WERR. For this experiment, all recordings from Earnings-22 Dev/Test that are 1 hour or greater in duration are used. Each recording is partitioned into segments of 2.7, 5.5, 10.9, 21.8 and 43.7 minutes using the moving window scheme discussed in § 4.3. NSTI is then performed separately on each of these partitions. Results when using the entire recording, which range from 61-76 minutes ($\mu = 66.2$), are also provided.

When the recording duration is equal to the context window of the model (2.7 minutes) we see very small gains of less than 1%. As the model is already attending over this data it is likely that some form of adaptation is already being performed. As expected the local context beyond the model's context window is more valuable, for example performing two epochs with a recording duration of 10.9 minutes is more beneficial than 1 epoch at 21.8 minutes. This helps explain the discrepancy in performance we see in § 5.3 between NSTI and NST. Self-training on separate adaptation data requires much more data because this data is much less representative of any recording at test time.

6. Conclusion

ASR models often degrade considerably in performance when the domain mismatch between training and testing/inference is large. In this work, to help address these challenges, we propose a self-training approach, which applies the noisy student teacher training framework on recordings at test time before transcribing them. Our proposed method improves considerably over prior test time adapatation methods for ASR. Additionally, the results demonstrate that this work leads to better performance than a more typical approach which uses a separate 105 hour training set for adaptation. We find that this is due to the high corellation between a current utterance and surrounding utterances in a recording. While, the method is very effective, it does not take into account the sequential nature of utterances in a recording, we plan to investigate this, alongside other augmentation strategies in our future work.

7. Acknowledgements

This work was supported by the CDT in Speech and Language Technologies (SLT) and their Applications funded by UKRI [grant number EP/S023062/1].

8. References

[1] T. Likhomanenko, Q. Xu, V. Pratap, P. Tomasello, J. Kahn, G. Avidov, R. Collobert, and G. Synnaeve, "Rethinking evaluation in asr: Are our models robust enough?" *arXiv preprint arXiv:2010.11745*, 2020.

[2] S. Khurana, N. Moritz, T. Hori, and J. Le Roux, "Unsupervised domain adaptation for speech recognition via uncertainty driven self-training," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6553–6557.

[3] Y. Higuchi, N. Moritz, J. L. Roux, and T. Hori, "Momentum pseudo-labeling for semi-supervised speech recognition," *arXiv preprint arXiv:2106.08922*, 2021.

[4] Z. Meng, J. Li, Y. Gaur, and Y. Gong, "Domain adaptation via teacher-student learning for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 268–275.

[5] Y. Higuchi, N. Moritz, J. Le Roux, and T. Hori, "Advancing momentum pseudo-labeling with conformer and initialization strategy," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7672–7676.

[6] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2. Atlanta, 2013, p. 896.

[7] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," *arXiv preprint arXiv:2006.10726*, 2020.

[8] J.-H. Lee, D.-H. Kim, and J.-H. Chang, "Awmc: Online test-time adaptation without mode collapse for continual adaptation," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[9] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.

[10] B. Krause, E. Kahembwe, I. Murray, and S. Renals, "Dynamic evaluation of neural sequence models," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2766–2775.

[11] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 687–10 698.

[12] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," *arXiv preprint arXiv:2005.09629*, 2020.

[13] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[14] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[15] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[16] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," *Advances in neural information processing systems*, vol. 17, 2004.

[17] G.-T. Lin, S.-W. Li, and H.-y. Lee, "Listen, adapt, better wer: Source-free single-utterance test-time adaptation for automatic speech recognition," *arXiv preprint arXiv:2203.14222*, 2022.

[18] A. Kedia and S. C. Chinthakindi, "Keep learning: Self-supervised meta-learning for learning from inference," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 63–77.

[19] A. Clifton, A. Pappu, S. Reddy, Y. Yu, J. Karlgren, B. Carterette, and R. Jones, "The spotify podcast dataset," *arXiv preprint arXiv:2004.04270*, 2020.

[20] M. Del Rio, P. Ha, Q. McNamara, C. Miller, and S. Chandra, "Earnings-22: A practical benchmark for accents in the wild," *arXiv preprint arXiv:2203.15591*, 2022.

[21] S. Gandhi, P. Von Platen, and A. M. Rush, "Esb: A benchmark for multi-domain end-to-end speech recognition," *arXiv preprint arXiv:2210.13352*, 2022.

[22] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[23] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Esteve, "Ted-lium 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in *Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20*. Springer, 2018, pp. 198–208.

[24] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj *et al.*, "Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," *arXiv preprint arXiv:2004.09249*, 2020.

[25] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[26] D. Rekesh, S. Kriman, S. Majumdar, V. Noroozi, H. Juang, O. Hrinchuk, A. Kumar, and B. Ginsburg, "Fast conformer with linearly scalable attention for efficient speech recognition," *arXiv preprint arXiv:2305.05084*, 2023.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[28] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," *Advances in neural information processing systems*, vol. 30, 2017.

[29] A. Defazio and S. Jelassi, "Adaptivity without compromise: a momentumized, adaptive, dual averaged gradient method for stochastic optimization," *J Mach Learn Res*, vol. 23, pp. 1–34, 2022.

[30] R. Flynn and A. Ragni, "How much context does my attention-based asr system need?" *arXiv preprint arXiv:2310.15672*, 2023.

[31] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.