



Teacher-Free Knowledge Distillation for Improving Short-Utterance Spoken Language Identification

Spandan Dey¹, Hirak Mondal¹, Sanjay Kumar Kurmi¹

¹Samsung R&D Institute, Bangalore, India

spandan.dey@samsung.com, hirak.mondal@samsung.com, sanjay.krm@samsung.com

Abstract

Spoken language identification (LID) systems exhibit performance degradations as the test input duration reduces. To delve deeper, we show that 36.94% of the misclassifications on 2-second (s) LID inputs occur due to out-of-scope elements like non-speech, named entities, filler words, and overlapped speech. To mitigate this, we propose a teacher-free knowledge distillation (TF-KD) using online label smoothing. This method accumulates prediction logits of correctly classified training segments from the preceding epoch and uses them as soft-labels for distillation in the next epoch. We further enhance TF-KD with dynamic weights, conditional label update, and entropy-based soft-label computation. Compared to existing KD-based solutions for 2s inputs, our approach achieves consistent C_{avg} improvements for both same-corpora and cross-corpora evaluations without training a separate teacher network.

Index Terms: Spoken language identification, knowledge distillation, online label smoothing, short-duration, VoxLingua107

1. Introduction

Spoken language identification (LID) systems are important as front-ends for multilingual speech processing applications, such as automatic speech recognition (ASR), speech to speech translation [1, 2]. In real-time services efficiency of LID systems on short-utterance inputs becomes crucial. It also enables streaming based services and enhances user flexibility and experience towards human-to-computer interaction (HCI) [3]. However, the existing LID literature shows a striking degradation in performance as the test input duration gets reduced [4, 5, 6]. As a justification for this observation, researchers have suggested the increased intra-language distribution variation for shorter test segments that catalyses model confusion and degrades LID performance [7, 8].

Considering the recent end-to-end deep learning based approaches, Shen et al. [8] used x-vector [9] mean compensation with longer speech segments for improving short-duration LID performance. Researchers have also explored knowledge distillation (KD) [10] with longer utterance trained teacher network and shorter utterance trained student network [4, 6]. Extensions are made on conventional KD networks such as with additional representation learning from internal layer's representations (KD-RL) or interactive teacher-student learning for reducing the LID performance degradation on shorter test inputs [11]. In spite these efforts, the amount of research explicitly addressing the issue of short-duration LID is yet limited.

In this study, using the top-ten most widely spoken languages from the VoxLingua107 database [12] we focus on mitigating the degradation of LID performance with shorter inputs. We first conduct a quantitative explanatory analysis for delv-

ing deeper on to why LID systems struggle when the test input duration is reduced to few seconds. Our analysis shows that a significant fraction of 2-second (s) inputs gets misclassified due to the presence of different *out-of-scope* (OOS) elements, such as non-speech, named entities (NEs), filler words, overlapped speech. Their presence becomes dominant in the entire segment as the input duration gets reduced. While voice activity detection (VAD) algorithms can be used to eliminate the non-speech portions, there exists a serious lack of reliable pre-trained models or algorithms for detecting other OOS elements, especially for the multilingual scenarios [13]. Hence, to eliminate the adverse effects of these OOS factors (which accounts for notable fraction of total misclassification), we propose *teacher-free knowledge distillation* (TF-KD) using the online label smoothing (OLS) [14]. We achieve prominent performance improvements over shorter 2s test segments for both same-corpora and cross-corpora evaluations (using Common Voice dataset [15]). The major contributions of our study is summarized as follows:

- We provide an in-depth quantitative analysis to elucidate why LID systems struggle with short-duration inputs, highlighting the dominant presence of OOS elements.
- To mitigate the impact of the OOS elements, we propose a teacher-free knowledge distillation (TF-KD). Even without training a teacher model, using online label smoothing, we achieve substantial short-duration LID performance improvements.
- Proposed TF-KD is further improved with three modifications: (i) Dynamic weight adjustment, (ii) conditional soft-label update, and (iii) entropy-based soft-label computation.

2. Methodology

We first discuss the prominent solutions from LID literature that tackle the issue of short-input LID. Then, we present the details of our teacher-free knowledge distillation approach.

2.1. Related works: Knowledge distillation

KD-basic: Knowledge distillation (KD) was first explored by Hinton et al. [10] where the prediction logits from a larger neural network (teacher) were used as soft-labels to efficiently supervise a smaller neural network (student). In this work, we first implement the notable KD-based framework by Shen et al. [6] where the authors trained a teacher model with larger 4s training segments and used its logits to supervise a student model (same architecture as teacher) trained with 2s training segments. Let, the teacher and the student model be denoted as θ_t and θ_s , respectively. Consider a 2s segment (\mathbf{x}_s, y_s) from the training data. In KD, the prediction logit from the teacher model $\theta_t(\mathbf{x}_s)$

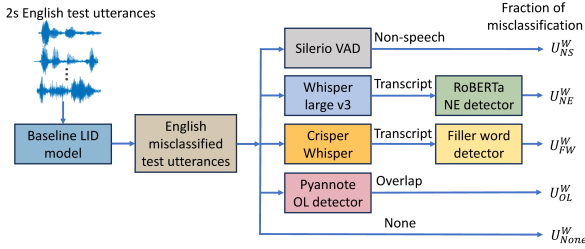


Figure 1: Analysing misclassification in short-utterance LID.

is used to supervise the student model θ_s . The total loss used to train the student model is:

$$L_{\text{tot}} = \lambda L_{\text{ce}} + (1 - \lambda) L_{\text{kd}} \quad (1)$$

Here, λ decides the weights of the two losses. L_{ce} is the commonly followed cross-entropy based classification loss using the hard-labels y_s . L_{kd} is the additional loss to incorporate the class-confusion knowledge learnt from the prediction logits of the teacher model:

$$L_{\text{kd}}(\theta_s(\mathbf{x}_s), \mathbf{q}) = -\frac{1}{L} \sum_{i=1}^L \mathbf{q}[i] \log \theta_s(\mathbf{x}_s)[i] \quad (2)$$

$$\mathbf{q}[i] = \frac{\exp(\theta_t(\mathbf{x}_s)[i])/T}{\sum_{j=1}^L \exp(\theta_t(\mathbf{x}_s)[j])/T} \quad (3)$$

Here, T denotes the temperature parameter to smooth the teacher prediction logits.

KD-RL: Authors in [6] also extended the basic KD frameworks with representation learning (RL) capabilities. Here, additional L_{r1} is also used with L_{tot} in Eq. 1. Let, the n -th layer of θ_t and θ_s be denoted as θ_t^n and θ_s^n , respectively. Here, L_{r1} computes the L_1 norm between the internal representations from the n -th layer between the teacher and student models, $L_{\text{r1}} = \|\theta_s^n(\mathbf{x}_s) - \theta_t^n(\mathbf{x}_s)\|$.

2.2. Explanatory analysis: Why LID performance degrades for shorter inputs?

Hypothesis: Input segments to a trained LID system contain some within-scope (WS) parts and some out-of-scope (OOS) parts. WS parts contain the spoken portions where unique language discriminating cues are present. However, the OOS parts are beyond the merits of the LID model as either they do not contain spoken portions or they contain spoken parts ambiguous across languages. As the input duration gets shorter, the likelihood of a segment to be entirely contained or dominated by an OOS element increases. As a consequence, even if the trained LID model efficiently learnt the language-discriminating cues, its chances of failures for shorter inputs increase prominently.

Analysis: First we infer the possible OOS factors from LID perspective: (i) Portion of non-speech segments (NS) [16], (ii) named entities (NE) [17, 18], (iii) filler words (FW) [19], (iv) portions with overlapping speech (OL) [20], (v) segments with code-switching (CS). For the analysis, we feed 2s input segments to our baseline LID model (cf. Section 3). We compute the misclassified segments. Then we explore pre-trained models to label each misclassified segment (with ground-truth English) in either of the four OOS labels: NS, NE, FW, OL. We then calculate and compare what fraction of the total misclassified segments are labelled into these OOS categories. We do not consider non-English misclassifications or the impact of code-switching due to lack of reliable multilingual pre-trained models (or deterministic algorithms). In Figure 1, we illustrate the detailed working principle of our analysis.

2.3. Proposed teacher-free knowledge distillation (TF-KD)

Among the above-discussed OOS factors, only the impact of NS can be filtered out prior due to the development of reliable voice activity detection (VAD) systems [16]. For the rest OOS factors, due to lack of multilingual pre-trained models or algorithms, we can not filter them out. These challenges motivate us to explore solutions which are based on the conventional KD-based frameworks but also attempts to reduce the impacts of OOS factors for improving short-utterance LID. For this we propose several teacher-free knowledge distillation (TF-KD) approaches. Unlike conventional KD frameworks, in the proposed TF-KD, we do not need to train a separate teacher network. The details of the proposed TF-KD approaches are discussed next.

Method-1: TF-KD- To make KD approach teacher-free while enabling the knowledge of information-rich soft-labels, we propose the preceding epoch to guide the following epoch. Further, to reduce the impact of the OOS factors (which attribute largely to the total misclassification), in the preceding epoch we accumulate the prediction logits of the correctly classified training segments. This accumulated prediction logit matrix is then normalized and used like soft-label teacher logits as KD to supervise the next epoch. This approach is inspired from the online label smoothing (OLS) algorithm [14] applied for object recognition tasks. Let, during the t -th epoch of training L languages, a segment (\mathbf{x}_i, y_i) is fed to the LID model θ which produces correct softmax logit $p(l|\mathbf{x}_i) \in \mathbb{R}^L$. We initialize a zero matrix $S^t \in \mathbb{R}^{L \times L}$ to accumulate the correct prediction scores. If $\arg \max p(l|\mathbf{x}_i) = y_i$, then we set $S^t[:, y_i] = p(l|\mathbf{x}_i)$. By doing so, at the end of the epoch, we set

$$S^t[:, k] = \frac{1}{N_k^t} \left(\sum_{j=1}^{N_k^t} p(l|\mathbf{x}_j) \right) \quad (4)$$

Here, N_k^t denotes the number of correctly classified training segments in the t -th epoch with true label $k \in [1, L]$, $S^t[:, k]$ denotes the k -th column of the S matrix. Likewise, for each correctly predicted training segment we update the corresponding column with the normalized prediction logits. At the next $t+1$ -th epoch, S^t is used as supervised matrix with soft-labels residing in each column for each ground-truth label $[1, L]$. However, to initialize, at the first epoch, we set uniform distributions for each column of S . The total loss to be optimized in the OLS-based TF-KD is:

$$L_{\text{TF-KD}} = \alpha L_{\text{hard}} + (1 - \alpha) L_{\text{soft}} \quad (5)$$

Here, for better convergence we also include the hard-label based loss with weight decided by α . We have set $\alpha = 0.7$ upon experimental validation. L_{soft} is computed as:

$$L_{\text{soft}} = - \sum_{k=1}^L S^t[:, k] \log p(k|\mathbf{x}_i) \quad (6)$$

Method-2: TF-KD with dynamic weight schedule- In Method-1, following the OLS approach [14], we keep static weights α across the epochs in Eq. 5. We hypothesize that in the initial epochs, (i) the number of correctly predicted training segments can be less, and (ii) the prediction logits may not be the most reliable. Hence, we extend the OLS in Method-2 of TF-KD with dynamically scheduling the value of α as a function of epoch number. For the initial epochs, we have set higher values of α with more emphasis on the hard-label loss. Once the number of epoch is passed by a limit (τ), we keep on gradually decreasing α for increasing the impact of the accumulated soft-labels (cf. Algorithm 1).

Method-3: TF-KD with dynamic weight schedule and conditional soft-label update- In the Method-1 and Method-2, the accumulated prediction logit matrix S^t at each epoch t is assigned to supervise the immediate next-epoch. However, during the training, not all epoch results in monotonic decrease in the validation loss. Training with the soft-labels accumulated from such epochs may hinder the convergence. Hence, instead of assigning S^t to supervise each $t + 1$ -th epoch, we first check if the validation loss at the t -th epoch gets reduced from the $t - 1$ -th epoch, then only S^t is utilized to supervise the $t + 1$ -th epoch. Otherwise, the supervise matrix is kept unaltered from the epoch where validation loss reduced last. This extension ensures that our TF-KD framework can only get the soft-label guidances which helps in better learning the language-discriminating cues.

Method-4: TF-KD with dynamic weight schedule, conditional soft-label update, and entropy-based soft-label computation- Even after correct prediction, prediction scores can lack confidence especially when the out-of-scope factors can present. In such cases, to further reduce the impact of less-confident correct classifications, instead of simple averaging, we modify Eq. 4 with entropy-based weighted averaging to produce more reliable soft-labels as presented below:

$$S^t[:, k] = \frac{1}{N_k^t} \left(\sum_{j=1}^{N_k^t} w_j \cdot p(l|\mathbf{x}_j) \right) \quad (7)$$

w_j is set as inverse of the prediction score entropy:

$$w_j = 1 / \left(\sum_{i=1}^L -p(l|\mathbf{x}_j)[i] \log p(l|\mathbf{x}_j)[i] \right) \quad (8)$$

For overall understanding, the detailed working procedure of the TF-KD is described in Algorithm 1.

Algorithm 1 Overall working principle of the proposed teacher-free knowledge distillation (Method-4).

```

1: procedure TF-KD METHOD-4( $\mathbf{X}, \mathbf{y}, \alpha_{\max}, \alpha_{\min}, \Delta_{\alpha}, \tau$ )
2:   Initialize  $t = 0$ ,  $S$  with uniform distributions,  $S_{\text{sup}} = S$ 
3:   while  $t \leq N_{\text{epoch}}$  do
4:      $t \leftarrow t + 1$ 
5:      $S^t \leftarrow \text{AccumulateCorrectPredLogit}(\mathbf{X}, \mathbf{y}, t)$ 
6:      $\alpha \leftarrow \text{ScheduleAlpha}(t, \alpha_{\max}, \alpha_{\min}, \Delta_{\alpha}, \tau)$ 
7:     Compute  $L_{\text{soft}}$  with  $S_{\text{sup}}$  (either  $S^{t-1}$  or  $S_{\text{prev}}$ )
8:      $L_{\text{TF-KD}} \leftarrow \alpha L_{\text{hard}} + (1 - \alpha) L_{\text{soft}}$ 
9:     Update model parameters using  $L_{\text{TF-KD}}$ 
10:     $S_{\text{sup}}, S_{\text{prev}} = (S^t, S^t)$  if  $[t < 2$  or
ValidationLossDecreased( $t$ )] else  $(S_{\text{prev}}, S_{\text{prev}})$ 
11: procedure ACCUMULATECORRECTPREDLOGIT( $\mathbf{X}, \mathbf{y}, t$ )
12:   Initialize  $S^t$  with zeros
13:   for each segment  $(\mathbf{x}_i, y_i)$  in  $(\mathbf{X}, \mathbf{y})$  do
14:     Compute softmax logit  $p(l|\mathbf{x}_i)$ 
15:     if  $\arg \max(p(l|\mathbf{x}_i)) = y_i$  then
16:        $w_i = 1 / (\sum_{j=1}^L -p(l|\mathbf{x}_i)[j] \log p(l|\mathbf{x}_i)[j])$ 
17:        $S^t[:, y_i] \leftarrow S^t[:, y_i] + w_i \cdot p(l|\mathbf{x}_i)$ 
18:   Normalize each column of  $S^t$ 
19:   return  $S^t$ 
20: procedure SCHEDULEALPHA( $t, \alpha_{\max}, \alpha_{\min}, \Delta_{\alpha}, \tau$ )
21:   return  $\alpha_{\max}$  if  $t < \tau$  else  $\min(\alpha_{\min}, \alpha_{\max} - \Delta_{\alpha} \cdot t)$ 
22: procedure VALIDATIONLOSSDECREASED( $t$ )
23:   return validation loss at epoch  $t < \text{epoch } t - 1$ 

```

3. Experiment details

Database: We have used utterances of ten most widely spoken languages in the world from the VoxLingua107 [12] database (Vox-10). These languages are: English, Mandarin, Hindi, Spanish, French, Arabic, Bengali, Portuguese, Russian, and Urdu. VoxLingua107 is one of the largest and most widely explored databases in the recent LID literature [5, 21, 22], consisting 6,628 hours of audios downloaded from video streaming platforms and averaging 62 hours per language. We perform a session disjoint split (with random seed value 42) of the collected database using 80:10:10 ratio for training, validation, and evaluation sets, respectively. The utterances are in .wav format and sampled in 16 kHz. Besides, for additional cross-corpora evaluation, we have also used the Mozilla Common Voice dataset [15].

Data pre-processing and feature extraction: For training our baseline and proposed TF-KD based LID frameworks, we segment the training and validation set utterances in 3s chunks. We primarily evaluate our models on short 2s segments. For implementing the KD-based literature solutions, we use longer 4s chunks and shorter 2s chunks for training the teacher and student model, respectively [6]. We apply VADs to filter out the non-speech chunks. Thereafter, we extract 80-dimensional log Mel-spectrogram features and apply mean subtraction based normalization on them.

Classifier architecture and training procedure: Following the recent LID literature [23, 24, 25, 26], we use the ECAPA-TDNN architecture [27] for training the LID frameworks. Cross-entropy loss is used with AdamW optimizer to train the models. We primarily set batch size of 64, learning rate of 0.001, and train the models for 50 epochs. We also use a validation loss based learning rate scheduler. For KD-based literature approaches, we use loss weight $\lambda = 0.7$. For the proposed TF-KD approaches, we apply $\alpha = 0.7$ with learning rate 0.0001. For dynamic weight, we experimentally (on validation set) fix $\alpha_{\max} = 0.8$, $\alpha_{\min} = 0.3$, $\Delta_{\alpha} = 0.02$, and $\tau = 2$. Following the NIST LRE [28, 29] and OLR challenges [30] in LID, we use cost-based metrics, such as equal error rate (EER) and $C_{\text{avg}} (*100)$ (primary metric).

4. Experiment results

We first explore why LID performance degrades for shorter inputs. Then, we compare the performance of our proposed TF-KD based methods with the existing KD-based solutions.

4.1. LID performance degradation for shorter utterances: Explanatory analysis

At first, we train a LID model with 3s segments without applying VAD for training and evaluation. We evaluate it with diverse test segment durations and report the results in Table 1. These results emphasize how LID performance gets strikingly degraded as we keep on reducing the test segment duration. To note, we assume 2s as the minimum duration required to gather enough linguistic context to discriminate languages. Following these preliminary results, we next explore how the different OOS factors impact for this performance degradation.

Following Figure 1, we consider the 2s English misclassified segments and compute the number of segments detected as (i) non-speech, (ii) named entity, (iii) filler word, and (iv) overlap speech. We also segregate the NE-detected segments into sub-categories, such as location (LOC), organization (ORG), person (PER), and miscellaneous (MISC). In Table 2, we show

Table 1: Baseline LID performance (trained without VAD) on different test duration. Accuracy is reported for reference.

Test segment duration	Metric		
	Acc (%)	EER (%)	C_{avg}
2s	51.16	23.53	22.62
3s	58.08	20.25	19.11
4s	62.69	18.16	17.11
Utterance	69.00	15.32	13.93

Table 2: Impact of out-of-scope labelled segments on overall LID misclassification for 2s Vox-10 English test segments.

Segment category	Number of test segments	Remarks
Total test utterances (U)	7911	23.57 % misclassification
Misclassified (U^w)	1865	
Non-speech detected (U_{NS}^w)	268	-
Named entity detected (U_{NE}^w)	292	LOC: 70, ORG: 43, PER: 69, MISC: 110
Filler word detected (U_{FW}^w)	249	-
Overlap speech detected (U_{OS}^w)	25	-
$U_{NS}^w \cup U_{NE}^w \cup U_{FW}^w \cup U_{OS}^w$	689	36.94% of misclassification are due to the OOS factors

that out of 7,911 2s English test segments, 1865 are misclassified. Among these 1865 misclassification, these four OOS factors alone contribute by 36.94%. The analysis motivates us to focus on effectively mitigate the adverse effects of these OOS factors to make LID systems robust towards shorter inputs.

4.2. LID performance of the proposed TF-KD methods

We next present the LID performances (for both short 2s segments and overall utterance-wise evaluation) of the proposed TF-KD methods. For comparison, we implement the baseline trained with 3s and 2s training chunks (after applying VAD). We also implement the state-of-the-art solutions available from literature that explicitly address the issue of short-input LID. The corresponding results in Table 3 show that the proposed TF-KD approaches clearly outperform the baseline and literature solutions. Further, we observe the TF-KD Method-4, which extends the online label smoothing with dynamic weights, conditional label update, and entropy-based soft-label computation, performs the best among all the proposed TF-KD methods.

4.3. Ablation experiments

Impact of α in the OLS loss for TF-KD: In Eq. 5, α is a crucial hyperparameter that balances the conventional cross-entropy hard-label loss and the online label smoothing loss for TF-KD. Increasing α can enhance training convergence, but it may reduce the influence of informative soft-labels in LID training. By varying $\alpha = [0, 0.9]$ in Method-1, we explore LID performances, as shown in Figure 2. The results indicate that a moderately high α value optimally balances both losses, leading to effective LID performance.

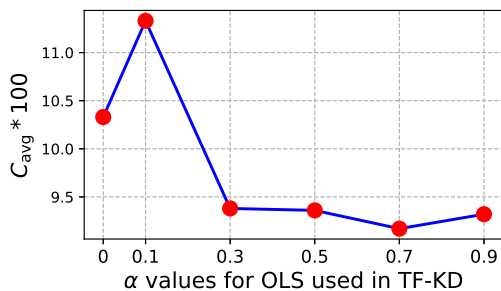


Figure 2: LID performances ($C_{avg} * 100$) for different α values in OLS used for TF-KD (Method-1).

Table 3: LID performance comparison of the proposed TF-KD based approaches on shorter 2s Vox-10 segments and overall utterance-wise evaluation. For comparison, we have also implemented the literature solutions for short-utterance LID.

LID framework		2s test segment		Whole utterance	
		EER (%)	Cavg	EER (%)	Cavg
Baseline model (ECAPA-TDNN)	Train-chunk 3s	10.45	11.20	4.46	5.00
	Train-chunk 2s	9.08	10.06	3.96	4.19
Literature	KD [11, 6]	9.07	9.47	3.96	4.14
	KD-RL [6]	9.08	9.18	4.16	4.81
	Compensation [8]	9.81	10.20	3.96	4.11
	Method-1	9.91	9.17	3.95	3.79
Proposed TF-KD	Method-2	9.12	8.30	3.97	3.60
	Method-3	9.22	8.25	3.85	3.35
	Method-4	8.94	8.24	3.80	3.41

Table 4: LID performances using the Vox-10 and Common Voice databases with both same-corpora and cross-corpora evaluations.

Training database	LID framework	Vox-10	Common Voice
		Cavg	Cavg
Vox-10	Baseline	11.20	20.40
	KD-RL	9.18	19.11
	TF-KD (Method-4)	8.24	16.22
Common Voice	Baseline	25.68	7.28
	KD-RL	26.73	9.07
	TF-KD (Method-4)	22.03	4.61

Verification with cross-corpora evaluation using additional database: To verify the effectiveness of our proposed approaches, we consider performing both same-corpora and cross-corpora experiments using another database. For each 10 language, we consider 10,000 utterances (for avoiding class imbalance) using the Common Voice dataset [15] (version 20.0). The utterances are randomly split into train, validation, and test sets using 80:10:10 ratio. We train the baseline, KD-RL [6], and proposed TF-KD Method-4 based frameworks using the Common Voice dataset and compare their performances (on 2s test segments) in Table 4. Additionally, we report the cross-corpora evaluations using the Vox-10 trained LID models. The obtained results clearly show that proposed TF-KD approach consistently outperform the baseline and literature solutions, solidifying the effectiveness of the proposed LID framework.

5. Conclusions

We conduct extensive analysis to find out why LID systems degrade drastically when the test input duration gets reduced. Our analysis show that the presence of out-of-scope factors, such as non-speech, named entities, overlap and filler speech becomes too dominant in shorter inputs, which can attribute to a significant fraction of total misclassification. To reduce the impact of these OOS factors, we propose a teacher-free knowledge distillation using online label smoothing. Our approach outperforms conventional knowledge-distillation based approaches for both same-corpora and cross-corpora evaluations without training a separate teacher network. In the future, we would like to extend our study with OOS-factor detection based multi-task learning for enhancing the explainability to the end users.

6. References

- [1] H. Li, B. Ma, and K. A. Lee, “Spoken language recognition: from fundamentals to practice,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.
- [2] E. Ambikairajah *et al.*, “Language identification: A tutorial,” *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82–108, 2011.
- [3] Q. Wang, Y. Yu, J. Pelecanos, Y. Huang, and I. L. Moreno, “Attentive temporal pooling for conformer-based streaming language

- identification in long-form speech,” in *Odyssey: The Speaker and Language Recognition Workshop*. ISCA, 2022, pp. 255–262.
- [4] F. Wang, L. Huang, T. Li, Q. Hong, and L. Li, “Conformer-based language embedding with self-knowledge distillation for spoken language identification,” in *INTERSPEECH*. ISCA, 2023, pp. 5286–5290.
- [5] F. Jia, N. R. Koluguri, J. Balam, and B. Ginsburg, “A compact end-to-end model with local and global context for spoken language identification,” in *INTERSPEECH*. ISCA, 2023, pp. 5321–5325.
- [6] P. Shen, X. Lu, S. Li, and H. Kawai, “Knowledge distillation-based representation learning for short-utterance spoken language identification,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2674–2683, 2020.
- [7] Shen, Peng and Lu, Xugang and Li, Sheng and Kawai, Hisashi, “Feature representation of short utterances based on knowledge distillation for spoken language identification.” in *INTERSPEECH*. ISCA, 2018, pp. 1813–1817.
- [8] P. Shen, X. Lu, K. Sugiura, S. Li, and H. Kawai, “Compensation on x-vector for short utterance spoken language identification.” in *Odyssey: The Speaker and Language Recognition Workshop*. ISCA, 2020, pp. 47–52.
- [9] D. Snyder *et al.*, “Spoken language recognition using x-vectors.” in *Odyssey: The Speaker and Language Recognition Workshop*. ISCA, 2018, pp. 105–111.
- [10] G. Hinton, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
- [11] P. Shen, X. Lu, S. Li, and H. Kawai, “Interactive learning of teacher-student model for short utterance spoken language identification,” in *ICASSP*. IEEE, 2019, pp. 5981–5985.
- [12] J. Valk and T. Alumäe, “VoxLingua107: a dataset for spoken language recognition,” in *Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 652–658.
- [13] S. Dey, M. Sahidullah, and G. Saha, “An overview on Indian spoken language recognition from machine learning perspective.” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 128, pp. 1–45, 2022.
- [14] C.-B. Zhang, P.-T. Jiang, Q. Hou, Y. Wei, Q. Han, Z. Li, and M.-M. Cheng, “Delving deep into label smoothing,” *IEEE Transactions on Image Processing*, vol. 30, pp. 5984–5996, 2021.
- [15] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, “Common voice: A massively-multilingual speech corpus,” in *Language Resources and Evaluation Conference*. European Language Resources Association, May 2020, pp. 4218–4222.
- [16] S. Team, “Silero VAD: pre-trained enterprise-grade voice activity detector (VAD), number detector and language classifier,” <https://github.com/snakers4/silero-vad>, 2024.
- [17] 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.
- [18] P. Srinivasan, R. Venkatakrishnan *et al.*, “Transformer-based models for named entity recognition: A comparative study,” in *International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2023, pp. 1–5.
- [19] M. Zusag, L. Wagner, and B. Thallinger, “CrisperWhisper: Accurate timestamps on verbatim speech transcriptions,” in *INTERSPEECH*. ISCA, 2024, pp. 1265–1269.
- [20] H. Bredin, “pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe,” in *INTERSPEECH*. ISCA, 2023, pp. 1983–1987.
- [21] M. Valente, F. Brugnara, G. Morrone, E. Zovato, and L. Badino, “Exploring spoken language identification strategies for automatic transcription of multilingual broadcast and institutional speech,” in *INTERSPEECH*. ISCA, 2024, pp. 1645–1649.
- [22] T. M. Bartley, F. Jia, K. C. Puvvada, S. Krیمان, and B. Ginsburg, “Accidental learners: Spoken language identification in multilingual self-supervised models,” in *ICASSP*. IEEE, 2023, pp. 1–5.
- [23] S. Dey, M. Sahidullah, and G. Saha, “Towards cross-corpora generalization for low-resource spoken language identification,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.
- [24] A. Prasad, A. Carofilis, G. Vanderreydt, D. Khalil, S. Madikeri, P. Motlicek, and C. Schuepbach, “Fine-tuning self-supervised models for language identification using orthonormal constraint,” in *ICASSP*. IEEE, 2024, pp. 11 921–11 925.
- [25] S. Dey, P. Singh, and G. Saha, “Wavelet scattering transform for improving generalization in low-resourced spoken language identification,” in *INTERSPEECH*. ISCA, 2023, pp. 1953–1957.
- [26] R. Duroselle, M. Sahidullah, D. Jouviet, and I. Illina, “Language recognition on unknown conditions: The LORIA-Inria-MULTISPEECH system for AP20-OLR challenge,” in *INTERSPEECH*. ISCA, 2021, pp. 3256–3260.
- [27] B. Desplanques, J. Thienpondt, and K. Demuynck, “ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification,” in *INTERSPEECH*. ISCA, 2020, pp. 1–5.
- [28] S. O. Sadjadi, T. Kheyrkhan, A. Tong, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, “The 2017 NIST language recognition evaluation,” in *Odyssey: The Speaker and Language Recognition Workshop*. ISCA, 2018.
- [29] Y. Lee, C. Greenberg, E. Godard, A. A. Butt, E. Singer, T. Nguyen, L. Mason, and D. Reynolds, “The 2022 NIST language recognition evaluation,” in *INTERSPEECH*, 2023, pp. 1928–1932.
- [30] Z. Li, M. Zhao, Q. Hong, L. Li, Z. Tang, D. Wang, L. Song, and C. Yang, “AP20-OLR challenge: Three tasks and their baselines,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2020, pp. 550–555.