Interspeech 2024
1-5 September 2024, Kos, Greece

Enhancing Neural Transducer for Multilingual ASR with Synchronized Language Diarization

Amir Hussein[1], *Desh Raj*[1], *Matthew Wiesner*[1,2], *Daniel Povey*[3], *Paola Garcia*[1,2], *Sanjeev Khudanpur*[1,2]

[1]CLSP & [2]HLTCOE, Johns Hopkins University, USA [3]Xiaomi Corp., China

{ahussei6, draj2, wiesner, lgarci27, khudanpur}@jhu.edu, dpovey@xiaomi.com

Abstract

In multilingual environments, seamless language switching, including code-switching (CS) within utterances, is essential for real-time applications. Conventional Automatic Speech Recognition (ASR) combined with language diarization requires post-processing to synchronize language labels with recognized words accurately, presenting a considerable challenge. In this study, we introduce a multitask learning framework that synchronizes Language Identification (LID) with ASR, utilizing a neural transducer architecture. This auxiliary task integrates both acoustic and lexical features to perform LID. Furthermore, we use resulting language representation as an auxiliary input to improve ASR. We demonstrate the efficacy of our proposed approach on conversational multilingual (Arabic, Spanish, Mandarin) and CS (Spanish-English, Mandarin-English) test sets.

Index Terms: speech recognition, multilingual, language identification, neural network transducer

1. Introduction

Digital voice assistants have been widely deployed in recent years, including in multilingual households. These systems are expected to seamlessly switch between multiple languages in real-time, including *intrasentential* language changes, a phenomenon known as code-switching (CS), which is prevalent in daily conversations [1]. This presents a unique challenge for voice technologies, necessitating dynamic language switching during interactions. Studies have demonstrated the crucial role of spoken language identification (LID) alongside automatic speech recognition (ASR), outperforming multilingual systems lacking LID [2–8].

With the rise of deep learning, researchers have adopted End-to-End (E2E) multilingual systems and proposed jointly modeling ASR with LID by extending the vocabulary with language tags [9–11]. This approach reduces the computational overhead of an additional LID module while enhancing recognition performance. However, subsequent studies have demonstrated that decoupling ASR and LID tasks and training them in a multitask fashion outperforms joint modeling of ASR and LID with shared vocabulary [12–14]. Another approach to address language switching is to utilize separate sub-models for each language [15, 16]. However, as the number of languages increases, this approach becomes computationally impractical. Recently, in [17], researchers proposed multilingual ASR with Attention based Encoder Decoder (AED) architectures. They integrated self-conditioned Connectionist Temporal Classification (CTC) as an additional language identification task within intermediate encoder layers to condition subsequent layers on intermediate predictions. However this approach focuses on utterance level LID and is not suitable for streaming applications. Conversely, researchers have turned to the Recurrent Neural Transducer (RNN-T) [18], a frame-synchronous E2E model suitable for streaming, competitive with state-of-the-art AED, and well-suited for on-device applications. Several studies have proposed leveraging RNN-T for multilingual speech processing through joint ASR and LID modeling [9, 14, 15, 19, 20].

In the aforementioned studies, the LID is not associated with words recognized by an ASR system, which is necessary for real-time interaction with minimum latency. This requires an additional module to merge ASR and LID results. To avoid this overhead, we propose jointly performing LID synchronized with the ASR output tokens. However, it has also been shown that naively making the language labels a part of ASR lexicon degrades streaming ASR performance [21, 22]. We therefore adopt a multitask approach, augmenting ASR with an auxiliary task that uses a separate encoder and joiner to predict language labels. Unlike prior work that jointly learns ASR and LID predictions, our proposed approach ensures synchronization between token-level LID and ASR predictions. To achieve this synchronization we utilize Hybrid Auto-regressive Transducer (HAT) loss [23], which factors the distribution over blank versus non-blank symbols. This separation facilitates the synchronization of the two predictions through shared blank symbols [24–26]. Recent studies on LID have demonstrated that combining acoustic and lexical cues significantly enhances prediction accuracy [27]. In our approach, the lexical predictor of RNN-T is shared between ASR and LID, providing seamless integration of acoustic and lexical features for the auxiliary LID task.

Our contributions include: 1) introducing an auxiliary LID module that combines acoustic and lexical features to improve performance, 2) enabling word-synchronous LID without ASR performance degradation, 3) improving a multilingual ASR system by feeding the LID representation back to ASR, and 4) releasing our code through the open-source `icefall` toolkit[1]. We empirically demonstrate a 10% relative improvement in mixed error rate (MER) on code-mixed utterances in a Mandarin-English CS data set, without compromising word/character error rate (W/CER) on single-language utterances in Arabic, Spanish and Mandarin data sets. We present ablation studies on the Mandarin-English CS data to assess the contribution of each model component on ASR performance.

2. Proposed Approach

Language identification with multilingual ASR is more valuable when associated with recognized words. Therefore, the goal is to identify speech words and their corresponding language labels simultaneously. However, synchronizing the lan-

[1]https://github.com/k2-fsa/icefall

10.21437/Interspeech.2024-1418

guage identification labels with automatic speech recognition (ASR) predictions requires accurately aligning the number of predictions and their positions. To address this challenge, we propose a multitask learning approach with sharing the blank label between ASR and LID tasks to facilitate synchronization, as detailed in Section 2.2.

2.1. ASR with HAT transducer

In standard ASR, the input to the system $\mathbf{X} \in \mathbb{R}^{T\times F}$, constitutes a T-long sequence of F-dimensional acoustic features and the objective is to predict $\mathbf{y} = (y_1, \ldots, y_U) \in \mathcal{V}^U$, a transcript of, say, graphemes or word-pieces of length U. Discriminative training is achieved by minimizing the negative log likelihood $\mathcal{L} = -\log P(\mathbf{y}|\mathbf{X})$. Transducers achieve this by marginalizing over the set of all alignments $\mathbf{a} \in \bar{\mathcal{V}}^{T+U}$ as following:

$$P(\mathbf{y}|\mathbf{X}) = \sum_{\mathbf{a}\in\mathcal{B}^{-1}(\mathbf{y})} P(\mathbf{a}|\mathbf{X}) \tag{1}$$

where $\bar{\mathcal{V}} = \mathcal{V} \cup \{\phi\}$, ϕ is a *blank* label and $\mathcal{B} : \bar{\mathcal{V}}^{T+U} \rightarrow \mathcal{V}^U$ is the deterministic mapping from an alignment $\mathbf{a}$ to the subsequence $\mathbf{y}$ of its non-blank symbols. Transducers parameterize $P(\mathbf{a}|\mathbf{X})$ with an encoder, a prediction network, and a joiner, as shown in the "ASR" branch in Figure 1. The *encoder* maps $\mathbf{X}$ to a representation sequence $\mathbf{f}^{\text{asr}}_{1:t}, t \in \{1, \ldots, T+U\}$, the *predictor* transforms $\mathbf{y}$ sequentially into $\mathbf{g}^{\text{asr}}_{1:u}, u \in \{1, \ldots, U\}$, and the *joiner* combines $\mathbf{f}^{\text{asr}}_{1:t}$ and $\mathbf{g}^{\text{asr}}_{1:u}$ to generate logits $z^{asr}_{t,u}$ whose softmax is the posterior distribution of a_t (over $\bar{\mathcal{V}}$), i.e.

$$\begin{aligned} P(\mathbf{y}|\mathbf{X}) &= \sum_{\mathbf{a}\in\mathcal{B}^{-1}(Y)} \prod_{t=1}^{T+U} P(\mathbf{a}_t|f^{asr}_{1:t}, g^{asr}_{1:u(t)}) \\ &= \sum_{\mathbf{a}\in\mathcal{B}^{-1}(Y)} \prod_{t=1}^{T+U} \text{softmax}(z^{asr}_{t,u(t)}) \end{aligned} \tag{2}$$

where $u(t) \in \{1, \cdots, U\}$ denotes the index in the label sequence at time t. The negative log of the quantity in (2) is known as the transducer loss.

In this work, we adopt the Hybrid Auto Regressive Transducer (HAT) [23], which first factorizes the distribution of a_t over blank versus all non-blanks, and models it via a Bernoulli distribution. The posterior probability of non-emission $b_{t,u}$ is computed from the logit $z_{t,u}[0]$ via sigmoid activation σ, as

$$b_{t,u} := P(a_t = \phi|f_{1:t}, g_{1:u}) = \sigma(z_{t,u}[0]), \tag{3}$$

and the softmax is applied to the remaining non-blank logits $z_{t,u}[1:U]$ to compute the distribution over lexical tokens conditioned on a_t being non-blank, as

$$P(a_t = \mathbf{y}_u|f_{1:t}, g_{1:u}) = (1 - b_{t,u}) \cdot \text{softmax}(z_{t,u}[1:]) \tag{4}$$

2.2. Joint ASR and LID multitask learning

The problem of jointly learning ASR and LID from acoustic features $\mathbf{X}$ can be formulated as estimating the conditional probability $P(\tilde{\mathbf{y}}|\mathbf{X})$, where $\tilde{\mathbf{y}} = \{(y_1, l_1), \ldots, (y_u, l_u)\}$ represents a tuple of ASR and LID labels. Given that our proposed approach models LID using an auxiliary encoder with a separate output space, $P(\tilde{\mathbf{y}}|\mathbf{X})$ can be decomposed into $P^{\text{asr}}(\mathbf{y}|\mathbf{X})$ and $P^{\text{lid}}(\mathbf{l}|\mathbf{X})$. In our approach, the prediction of LID labels depends on both acoustic and preceding lexical features: $P^{\text{lid}}(\mathbf{l}|\mathbf{X}) = \prod_i P(l_i|y_{1:i-1}, \mathbf{X})$. We extend the HAT framework to incorporate an additional LID task by including an auxiliary LID transducer, as depicted in Figure 1. The LID encoder

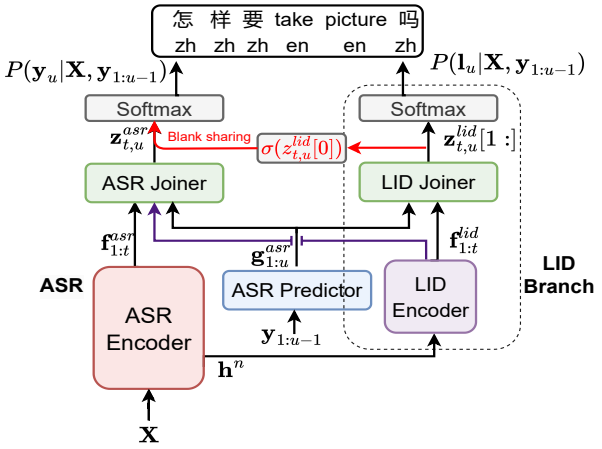

Figure 1: *Illustration of proposed multitask ASR and LID architecture.*

receives intermediate representations h^n from the n^{th} layer of the ASR encoder, producing $f^{\text{lid}}_{1:t}$. The LID joiner, $\text{Joiner}_{\text{lid}}(\cdot)$, combines $f^{\text{lid}}_{1:t}$ and $\mathbf{g}^{\text{asr}}_{1:u}$ to produce the LID logits $z^{\text{lid}}_{t,u}$. The model calculates the LID probabilities as follows:

$$\begin{aligned} z^{\text{lid}}_{t,u} &= \text{Joiner}_{\text{lid}}(f^{\text{lid}}_{1:t}, \mathbf{g}^{\text{asr}}_{1:u}) \\ P^{\text{lid}}(\mathbf{l}_u|\mathbf{X}) &= P(\mathbf{l}_u|f^{\text{lid}}_{1:t}, \mathbf{g}^{\text{asr}}_{1:u}) \\ &= (1 - b^{\text{lid}}_{\text{t,u}}) \cdot \text{softmax}(z^{\text{lid}}_{t,u}[1:]) \end{aligned} \tag{5}$$

where $b^{\text{lid}}_{t,u} = \sigma(z^{\text{lid}}_{t,u}[0])$. To synchronize blank emissions between the ASR and LID branches, we reuse the blank logit predicted by the LID in the ASR as follows:

$$z^{\text{asr}}_{t,u} = [z^{\text{lid}}_{t,u}[0], z^{\text{asr}}_{t,u}[1:]] \tag{6}$$

To allow ASR dependency on language features the LID representations are fed into the ASR system and combined within the ASR joiner, $\text{Joiner}_{\text{asr}}(\cdot)$, as follows:

$$\begin{aligned} z^{\text{asr}}_{t,u} &= \text{Joiner}_{\text{asr}}(f^{\text{asr}}_{1:t}, f^{\text{lid}}_{1:t}, \mathbf{g}^{\text{asr}}_{1:u}) \\ P^{\text{asr}}(\mathbf{y}_u|\mathbf{X}) &= P(\mathbf{y}_u|f^{\text{asr}}_{1:t}, f^{\text{lid}}_{1:t}, \mathbf{g}^{\text{asr}}_{1:u}) \\ &= (1 - b^{\text{lid}}_{t,u}) \cdot \text{softmax}(z^{\text{asr}}_{t,u}) \end{aligned} \tag{7}$$

The model is optimized using a multi-task learning objective that combines the ASR loss $\mathcal{L}^{\text{asr}}_{\text{hat}}$ and the LID loss $\mathcal{L}^{\text{lid}}_{\text{hat}}$:

$$\mathcal{L} = (1 - \alpha_{\text{lid}})\mathcal{L}^{\text{asr}}_{\text{hat}} + \alpha_{\text{lid}}\mathcal{L}^{\text{lid}}_{\text{hat}} \tag{8}$$

where α_{lid} is the interpolation weight. For ASR and LID branches we utilize a pruned version of the HAT loss similar to pruned RNN-T [28].

3. Experimental Setup

Data: We demonstrate the effectiveness of our proposed method through evaluations in two distinct scenarios: code-switching, utilizing the Mandarin-English SEAME dataset [29], and multilingual, using three conversational datasets, Fisher-CallHome Spanish [30], IWSLT22 Tunisian [31], and BOLT Mandarin [32]. Detailed statistics for these datasets are presented in Table 1. The Fisher dataset includes approximately 15 hours of code-switching data for training and 2 hours for testing, which we prepared following [33].

Table 1: *Statistics for the multilingual and code switching (CS) ASR corpora.*

	Corpus	Lang	#Hours		
			Train	Dev	Test
Multilingual	Fisher/Callhome	sp	186.3	9.3	4.5/1.8
	Tunisian	ar	161.0	6.3	3.6
	BOLT	zh	110.6	8.5	8.5
CS	SEAME	zh-en	96.5	5.2	4 / 7.5

Data pre-processing: We use Lhotse [34] toolkit for speech data preparation. All audios are augmented with speed perturbations (0.9, 1.0 and 1.1) and transformed into 80-dimensional feature frames extracted on 25ms frames with frame shift of 10ms. Additionally, we augment the features using on-the-fly SpecAugment [35]. For the monolingual datasets we use a shared BPE vocabulary of size 5000 and for Seame we combine 2622 Mandarin characters with 1378 English BPE units.
Models: The ASR encoders are based on Zipformer architecture [36]. We conduct all experiments by customizing the Icefall toolkit [2]. For all experiments the ASR encoder consisting of 6 blocks with numbers of attention heads for each block are set to {4, 4, 4, 8, 4, 4}, feed forward dimensions are set to {512, 768, 1024, 1024, 1024, 768}, and convolution kernel sizes are set to {31, 31, 15, 15, 15, 31}. For the LID branch, the encoder consists of 3 blocks with numbers of attention heads for each block are set to {2, 4, 2}, feed forward dimensions are set to {256,256,256} and convolution kernel sizes are set to {31,15,31}. In each attention head for both encoders, the query dimension and value dimension are set to 32 and 12, respectively. The stateless prediction network implemented using a single 256-dim Conv1D layer with kernel size of 2. Our training configuration utilizes ScaledAdam optimizer [36] with a learning rate of 0.045 warmed up for 5K iterations, and the interpolation weight α_{lid} in Eq. (8), that provided best performance, is 0.3. The model size without the LID branch is 30M parameters and with LID branch is 35M parameters. All models were trained for 25 epochs using 4 Titan RTX GPUs with batch size of 500 seconds.
Evaluation: During decoding, we employ a beam search with beam of size 10. Evaluation is performed on SEAME test sets, measuring mixed error-rate (MER) that considers word-level English and character-level Mandarin. We also report WER on monolingual English and CER on monolingual Mandarin subsets. We test the significance in the WER improvements using Matched-Pair Sentence Segment Word Error (MAPSSWE) introduced by [37], with a significance level of $p = 5\%$. In addition, for LID performance we report the F1 score.

4. Results

4.1. Multilingual ASR

In this part, we examine the impact of multitask training on ASR with an auxiliary LID task where the models are trained on multilingual conversational data from Table 2. Our HAT implementation with Zipformer is based on the pruned transducer, described in Section (2.2), therefore, we first explore the impact of using HAT blank factorization. It can be observed from the first two rows that, apart from Bolt, which experiences a relative improvement of 8.3% in CER with HAT, the results are consistent. Notably, introducing the auxiliary LID task alongside blank synchronization does not compromise ASR performance, instead offering a relative improvement of about 3.6% across both Fisher code-switching (CS) and Bolt datasets. We found that using representations from the third ASR encoder block as input to the LID encoder is most effective. Observing that the majority of this improvement is seen in the code-switching subset (Fisher-CS), we conduct further investigations on the standard CS SEAME dataset, as detailed in Section (4.2).

[2]https://github.com/k2-fsa/icefall

Table 2: *Comparative analysis of WER/CER results for models trained on multilingual data. †: denotes a statistically significant difference ($p < 0.05$) compared to the HAT baseline. The number following HAT specifies the layer from which ASR representations are provided as input to LID.*

Model	Tunisian	Fisher	Fisher-CS	Callhome	Bolt
	WER	WER	WER	WER	CER
RNN-T	42.1	18.4	**27.7**	29.9	25.3
HAT(Baseline)	**42.0**	18.4	27.8	**29.7**	**23.2**†
HAT(1) + LID	41.8	18.4	26.8	30.0	23.0
HAT(6) + LID	42.0	18.5	27.7	30.1	23.2
HAT(3) + LID	**41.7**	**18.3**	**26.8**†	**29.9**	**22.4**†

4.2. Ablations of multitask approach components

To better understand the effectiveness of our technique, we conducted analysis across different languages as shown in Table 3. We begin by assessing the impact of adding an auxiliary branch with transducer loss to Zipformer (**HAT-Seg**), tasked with predicting blank/non-blank states, effectively serving as an alignment predictor. This approach resulted in up to a 6.7% relative improvement in the MER, underscoring the importance of alignment prediction as an auxiliary task. Subsequently, we explored the impact of auxiliary branch tasked with predicting LIDs using Connectionist Temporal Classification (**HAT-LID CTC**) and HAT objectives (**HAT-LID HAT**) (see rows 3 and 4). Our analysis indicate that multitask learning with both CTC and HAT losses leads to improvements compared to the HAT baseline. Notably, LID with HAT surpasses LID with CTC, achieving a relative improvement of up to 2.8% in MER. Additionally, enriching ASR joiner with LID representations (**HAT-LID+R**) yields further relative improvements reaching up to 2% in MER. The overall relative improvement of our proposed **HAT-LID+R** approach compared to the **HAT** baseline,

Table 3: *Comparative analysis of CER/WER/MER results for models trained on SEAME data. **HAT-LID+R**: Multitask HAT with LID representation passed to ASR. **HAT-Seg**: auxiliary branch used to predict alignments with single label. † and ††: Denotes statistically significant difference ($p < 0.05$) compared to HAT and compared to CTC respectively.*

Model	Dev-Man			Dev-Sge		
	CER-MAN	WER-EN	MER	CER-MAN	WER-EN	MER
HAT (baseline)	18.5	36.5	22.0	34.7	**28.5**	29.9
HAT-Seg	**17.8**†	**34.5**†	**20.9**†	**28.1**†	32.5	**27.9**†
HAT-LID (CTC)	18.0	34.8	21.0	28.4	32.2	27.6
HAT-LID (HAT)	16.9††	34.4	20.4††	27.4††	32.1	27.2
HAT-LID+R	**16.7**††	**34.0**††	**20.0**††	**26.8**††	**32.0**	**26.9**††

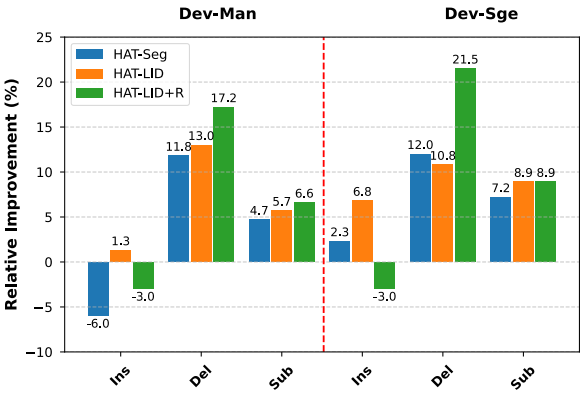

Figure 2: *Relative change in insertions, deletions and substitutions on SEAME dataset (Dev-Man, Dev-Sge).* ***HAT-Seg****: LID branch with single label to predict alignments.* ***HAT-LID+R****: proposed multitask architecture with LID encoder representation passed to ASR.*

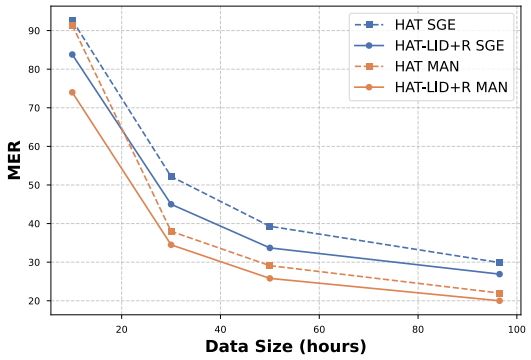

Figure 3: *MER with different amount of hours of code switching SEAME data.* ***HAT-LID+R****: proposed multitask HAT with LID encoder representation passed to ASR.*

are 9% and 10% in MER for Dev-Man and Dev-Sge, respectively. A closer inspection across individual languages reveals that the most gains predominantly emanate from the embedded languages (English in Dev-Man and Mandarin in Dev-Sge). To better understand where our model improves, we analyze the relative change in various error types (insertions, deletions and substitutions) compared to the HAT baseline as illustrated in Figure 2. We observe a consistent trend, with the most significant relative improvements occurring in deletions, followed by substitutions. Notably, the inclusion of LID labels (**HAT-LID**) improves substitutions, owing in part to more accurate language prediction. This hypothesis is explored further in Section 4.3. Moreover, leveraging the auxiliary LID encoder representation (**HAT-LID+R**) yields further improvements in both deletions and substitutions.

4.3. LID results on CS data

In this section, we analyze the improvements resulting from more accurate language identification predictions. We examine the improvements in the F1 scores on SEAME dataset, as detailed in Table 4. Our analysis reveals negligible improvements in languages identification when employing an auxiliary branch to predict blank vs non-blanks (**HAT-Seg**) compared to baseline HAT. Notably, employing both CTC and HAT as auxiliary tasks to predict language labels alongside blanks (see rows 3 and 4)

Table 4: *Comparative analysis of LID F1 scores for models trained on SEAME data.* ***HAT-Seg****: LID branch with single label to predict alignments.* ***HAT-LID+R****: Multitask HAT with LID encoder representation passed from LID branch to ASR.*

Model	DevMan			DevSge		
	F1-MAN	F1-EN	F1	F1-MAN	F1-EN	F1
HAT (baseline)	0.946	0.876	0.926	0.902	0.909	0.905
HAT-Seg	0.948	0.878	0.927	0.901	0.910	0.905
HAT-LID (CTC)	0.948	0.879	0.928	0.898	0.907	0.904
HAT-LID (HAT)	0.950	0.885	0.930	0.905	0.910	0.907
HAT-LID+R	**0.952**	**0.888**	**0.932**	**0.908**	**0.912**	**0.909**

results in improvements in LID performance compared to the baseline with HAT surpassing CTC. This indicates that the integration of acoustic and linguistic features effectively improves LID performance. Finally, leveraging the auxiliary LID encoder representation (**HAT-LID+R**) yields further gains in language identification performance. This results in overall relative reduction of 8.1% and 4.2% in complement of F1 scores for Dev-Man and Dev-Sge, respectively, when compared to the HAT baseline.

4.4. Data size effect on performance

In this section, we investigate the impact of the amount of code-switched (CS) training data on the performance of our proposed multitask ASR system with LID, compared to the baseline HAT. Figure 3 presents the WER for different volumes of training data in hours (10 hours, 30 hours, 50 hours, 96 hours). Interestingly, we observe significant improvement of up to 19% in relative MER with 10 hours, which then gradually decreases to $\approx$ 10% in relative MER as the volume of training data increases to its full size. This pattern suggests that our approach is more robust than the baseline when CS data is scarce, but the magnitude of improvement decreases as the amount of data grows.

5. Conclusion

In this paper, we introduce a multitask learning approach that integrates multilingual ASR with language identification (LID), based on a neural transducer, suitable for real-time interactions. The proposed approach synchronizes token-level predictions between between ASR and LID through blank sharing. This method significantly improves ASR performance on code-switching data without compromising the accuracy on multilingual data with single-language utterances. Our analysis highlights that the inclusion of an auxiliary task in the ASR primarily enhances deletion corrections, which is closely linked to speech activity detection, followed by improvement in substitutions. Further investigation shows that the error reduction in substitutions are partly due to improvements in LID accuracy. Moreover, we show that combining acoustic and linguistic features boosts LID performance. In the future, we would like to investigate the potential of this approach in processing long-form content taking into account historical context to further refine our model's capabilities.

Acknowledgment. This work was partially funded by NSF CCRI Grant No 2120435.

6. References

[1] S. Sitaram *et al.*, "A survey of code-switched speech and language processing," *ArXiv preprint*, vol. abs/1904.00784, 2019.

[2] Z. Zeng *et al.*, "On the End-to-End Solution to Mandarin-English Code-Switching Speech Recognition," in *Proc. Interspeech*, 2019, pp. 2165–2169.

[3] J. Li *et al.*, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1,

[4] K. Li *et al.*, "Towards code-switching ASR for end-to-end ctc models," in *Proc. ICASSP*, 2019, pp. 6076–6080.

[5] B. Li *et al.*, "Multi-dialect speech recognition with a single sequence-to-sequence model," in *Proc. ICASSP*, 2018, pp. 4749–4753.

[6] A. Waters *et al.*, "Leveraging language id in multilingual end-to-end speech recognition," in *Proc. ASRU*, 2019, pp. 928–935.

[7] J. Zhang *et al.*, "E2E-based multi-task learning approach to joint speech and accent recognition," in *Proc. Interspeech*, 2021, pp. 1519–1523.

[8] L. Zhou *et al.*, "A configurable multilingual model is all you need to recognize all languages," in *Proc. ICASSP*, 2022, pp. 6422–6426.

[9] S. Zhang *et al.*, "Rnn-transducer with language bias for end-to-end mandarin-english code-switching speech recognition," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2021, pp. 1–5.

[10] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *Proc. ASRU*, 2017, pp. 265–271.

[11] B. Li *et al.*, "Scaling end-to-end models for large-scale multilingual ASR," *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1011–1018, 2021.

[12] S. Toshniwal *et al.*, "Multilingual speech recognition with a single end-to-end model," in *Proc. ICASSP*, 2018, pp. 4904–4908.

[13] C. Shan *et al.*, "Investigating end-to-end speech recognition for mandarin-english code-switching," in *Proc. ICASSP*, 2019, pp. 6056–6060.

[14] S. Punjabi *et al.*, "Joint ASR and language identification using rnn-t: An efficient approach to dynamic language switching," in *Proc. ICASSP*, 2021, pp. 7218–7222.

[15] S. Dalmia *et al.*, "Transformer-transducers for code-switched speech recognition," in *Proc. ICASSP*, 2021, pp. 5859–5863.

[16] B. Yan *et al.*, "Towards zero-shot code-switched speech recognition," in *Proc. ICASSP*, 2023, pp. 1–5.

[17] W. Chen *et al.*, "Improving massively multilingual ASR with auxiliary ctc objectives," in *Proc. ICASSP*, 2023, pp. 1–5.

[18] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[19] C. Zhang *et al.*, "Streaming End-to-End Multilingual Speech Recognition with Joint Language Identification," in *Proc. Interspeech*, 2022, pp. 3223–3227.

[20] B. Yan *et al.*, "Joint modeling of code-switched and monolingual ASR via conditional factorization," in *Proc. ICASSP*, 2022, pp. 6412–6416.

[21] S.-Y. Chang *et al.*, "Turn-Taking Prediction for Natural Conversational Speech," in *Proc. Interspeech*, 2022, pp. 1821–1825.

[22] S.-Y. Chang *et al.*, "Streaming Intended Query Detection using E2E Modeling for Continued Conversation," in *Proc. Interspeech*, 2022, pp. 1826–1830.

[23] E. Variani *et al.*, "Hybrid autoregressive transducer (HAT)," in *Proc. ICASSP*, 2020, pp. 6139–6143.

[24] W. Wang *et al.*, "Multi-output RNN-T joint networks for multi-task learning of ASR and auxiliary tasks," in *Proc. ICASSP*, 2023, pp. 1–5.

[25] Y. Huang *et al.*, "Towards word-level end-to-end neural speaker diarization with auxiliary network," *ArXiv preprint*, vol. abs/2309.08489, 2023.

[26] D. Raj *et al.*, "On speaker attribution with SURT," in *Speaker Odyssey*, 2024.

[27] C. Chandak *et al.*, "Streaming language identification using combination of acoustic representations and ASR hypotheses," *ArXiv preprint*, vol. abs/2006.00703, 2020.

[28] F. Kuang *et al.*, "Pruned RNN-T for fast, memory-efficient ASR training," in *Proc. Interspeech*, 2022.

[29] D.-C. Lyu *et al.*, "Seame: A mandarin-english code-switching speech corpus in south-east asia," in *Proc. Interspeech*, 2010.

[30] M. Post *et al.*, "Improved speech-to-text translation with the fisher and callhome Spanish-English speech translation corpus," in *Proceedings of the 10th International Workshop on Spoken Language Translation: Papers*, 2013.

[31] E. Ansari *et al.*, "Findings of the IWSLT 2020 evaluation campaign," 2020, pp. 1–34.

[32] Z. Song *et al.*, "Collecting natural SMS and chat conversations in multiple languages: The BOLT phase 2 corpus," 2014, pp. 1699–1704.

[33] O. Weller *et al.*, "End-to-end speech translation for code switched speech," in *Proc. ACL*, 2022, pp. 1435–1448.

[34] P. Żelasko *et al.*, "Lhotse: A speech data representation library for the modern deep learning ecosystem," in *NeurIPS Data-Centric AI Workshop*, 2021.

[35] D. S. Park *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," 2019, pp. 2613–2617.

[36] Z. Yao *et al.*, "Zipformer: A faster and better encoder for automatic speech recognition," in *ICLR*, 2024.

[37] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. ICASSP*, 1989, pp. 532–535.