Towards Zero-Shot Code-Switched Speech Recognition

TOWARDS ZERO-SHOT CODE-SWITCHED SPEECH RECOGNITION

Brian Yan[1], *Matthew Wiesner*[2], *Ondřej Klejch*[3], *Preethi Jyothi*[4], *Shinji Watanabe*[1,2]

[1]Carnegie Mellon University, US, [2]Johns Hopkins University, US,
[3]University of Edinburgh, UK, [4]Indian Institute of Technology Bombay, IN

ABSTRACT

In this work, we seek to build effective code-switched (CS) automatic speech recognition systems (ASR) under the zero-shot setting where no transcribed CS speech data is available for training. Previously proposed frameworks which conditionally factorize the bilingual task into its constituent monolingual parts are a promising starting point for leveraging monolingual data efficiently. However, these methods require the monolingual modules to perform *language segmentation*. That is, each monolingual module has to simultaneously detect CS points and transcribe speech segments of one language while ignoring those of other languages – not a trivial task. We propose to simplify each monolingual module by allowing them to transcribe all speech segments indiscriminately with a monolingual script (i.e. *transliteration*). This simple modification passes the responsibility of CS point detection to subsequent bilingual modules which determine the final output by considering multiple monolingual transliterations along with external language model information. We apply this transliteration-based approach in an end-to-end differentiable neural network and demonstrate its efficacy for zero-shot CS ASR on Mandarin-English SEAME test sets.

Index Terms— code-switched ASR, zero-shot ASR, CTC

1. INTRODUCTION

In order to build multilingual automatic speech recognition (ASR) systems that are robust to code-switching (CS), practitioners must tackle both the long-tail of possible language pairs [1] and the relative infrequency of intra-sententially CS examples within collected training corpora [2]. Therefore, a preeminent challenge in the CS ASR field is to build effective systems under the zero-shot setting where no CS ASR training data is available. Recent advancements in multilingual speech recognition have demonstrated the impressive scale of cross-lingual sharing in neural network approaches [3–10], and these works have shown that jointly modeling ASR with language identity (LID) grants some intra-sentential CS ability [9–11]. However, most of these large scale models skew towards high-resourced languages [7] and do not seek to directly optimize for intra-sentential CS ASR between particular language pairs.

A more promising direction towards zero-shot CS ASR can be found in prior works which seek to incorporate monolingual data directly to improve CS performance [12–22]. In particular, there are several works which achieve joint modeling of CS and monolingual ASR by conditionally factorizing the overall bilingual task into monolingual parts [23–25]. By using label-to-frame synchronization, this *conditionally factorized* approach can make a CS prediction given only the predictions of the monolingual parts [23] – theoretically these conditionally factorized models can model CS ASR without any CS data, but this has not been previously confirmed.

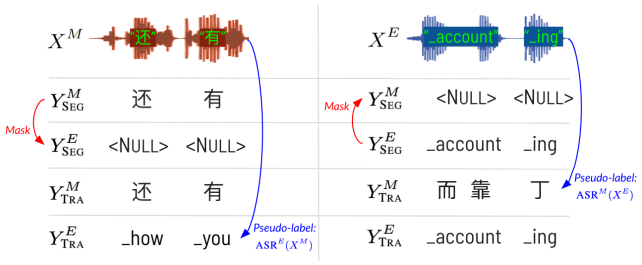

Fig. 1. Examples showing the difference between language segmentation targets $Y^{M/E}_{\text{SEG}}$ obtained via masking (§2.2) vs. transliteration targets $Y^{M/E}_{\text{TRA}}$ obtained via cross-lingual pseudo-labeling (§3.1).

In this work, we seek to build CS ASR systems under two zero-shot data conditions: 1) monolingual speech and CS text data are available, 2) only monolingual speech and text data are available. In particular, we are interested in exploring the zero-shot capability of conditionally factorized joint CS and monolingual ASR models.

We first re-formulate the initial monolingual stage of these conditionally factorized models in terms of their *language segmentation* burden, showing that prior works expect each monolingual module to perform CS point detection and transcription in tandem. Any errors in CS point detection are thus propagated downstream to the final bilingual stage which attempts to stitch multiple monolingual predictions into an output which may or may not be CS. To improve model robustness towards zero-shot CS ASR, we propose an alternative formulation of the monolingual stage such that each module is an indiscriminate *transliterator*, transcribing all speech using a monolingual script without any regard for potential CS points. As a result we delay CS point detection until the final bilingual stage, allowing our models to condition this critical decision on multiple monolingual inputs and incorporate additional information from external language models. Our transliteration-based method yielded 5 absolute error-rate reduction in our zero-shot CS ASR experiments.

2. BACKGROUND AND MOTIVATION

In this section, we examine the language segmentation role of the monolingual modules in previously proposed conditionally factorized models [23], motivating our transliteration-based approach (§3).

2.1. Joint Modeling of Code-Switched and Monolingual ASR

Let us take the Mandarin-English bilingual pair as an example for the following formulations. Bilingual ASR, where speech may or may not be CS, is a sequence mapping from a T-length speech feature sequence, $X = \{\mathbf{x}_t \in \mathbb{R}^D | t = 1, ..., T\}$, to an L-length label

sequence, $Y = \{y_l \in (\mathcal{V}^M \cup \mathcal{V}^E) | l = 1, ..., L\}$ consisting of Mandarin $\mathcal{V}^M$ and English $\mathcal{V}^E$. The conditionally factorized framework [23] decomposes this bilingual task into three sub-tasks: 1) recognizing Mandarin, 2) recognizing English, and 3) composing recognized monolingual segments into a bilingual sequence.

The basis of this approach is to model the label-to-frame alignments. For each T-length observation sequence X and L-length bilingual label sequence Y there are a number of possible T-length label-to-frame sequences $Z = \{z_t \in \mathcal{V}^M \cup \mathcal{V}^E \cup \{\varnothing\} | t = 1 \dots T\}$, where $\varnothing$ denotes a blank symbol as in Connectionist Temporal Classification (CTC) [26] or RNN-T [27]. Further consider that for each bilingual Z there are two corresponding monolingual label-to-frame sequences $Z^M = \{z_t^M \in \mathcal{V}^M \cup \{\varnothing\} | t = 1 \dots T\}$ and $Z^E = \{z_t^E \in \mathcal{V}^E \cup \{\varnothing\} | t = 1 \dots T\}$. The label posterior, $p(Y|X)$, can thus be represented in terms of bilingual, $p(Z|X)$, and monolingual, $p(Z^M|X)$ and $p(Z^E|X)$, label-to-frame posteriors as follows:

$$p(Y|X) = \sum_{Z \in \mathcal{Z}} \sum_{Z^M \in \mathcal{Z}^M} \sum_{Z^E \in \mathcal{Z}^E} p(Z, Z^M, Z^E | X) \quad (1)$$

where $\mathcal{Z}$ and $\mathcal{Z}^{M/E}$ denote sets of all possible bilingual and monolingual label-to-frame alignments for a given Y. Eq. (1) is the exact *joint* bilingual and monolingual ASR likelihood which can be further factorized using independence assumptions to obtain the form:

$$p(Y|X) \approx \underbrace{\sum_{Z} p(Z|Z^M, Z^E)}_{\text{Bilingual Posterior}} \underbrace{\sum_{Z^M} p(Z^M|X) \sum_{Z^E} p(Z^E|X)}_{\text{Monolingual Posteriors}} \quad (2)$$

From Eq. (1) to Eq. (2), the first assumption is that given Z^M and Z^E, no other information from the observation X is required to determine Z, allowing for conditional modeling of the bilingual posterior $p(Z|Z^M, Z^E, \cancel{X})$ given only monolingual information. The second assumption is that given X, Z^M and Z^E are independent, allowing for separate modeling of monolingual posteriors $p(Z^M|\cancel{Z^E}, X)$ and $p(Z^E|\cancel{Z^M}, X)$. Note we abbreviate this pair of separate monolingual modules as $p(Z^{M/E}|X)$ in future sections.

2.2. Modeling $p(Z^{M/E}|X)$ with Language Segmentation

What should be the behavior of the monolingual Mandarin module $p(Z^M|X)$ when encountering a segment of English speech and vice versa? Monolingual modules in prior works [23–25] determine each label-to-frame alignment $z_t^{M/E}$ by first determining the language identity of each speech frame $\text{LID}(\mathbf{x}_t)$ [28]. If the speech frame $\mathbf{x}_t$ is from a foreign language then the module will ignore it by emitting a special <NULL> token, otherwise it will transcribe using its monolingual vocabulary. This monolingual *language segmentation* decision is defined as follows (shown for Mandarin):

$$z_t^M = \begin{cases} \underset{m \in \mathcal{V}^M \cup \{\varnothing\}}{\text{argmax}} \; p(z_t^M = m | X, z_{1:t-1}^M) & \text{if LID}(\mathbf{x}_t) \text{ is } M \\ \underset{m \in \{\text{<NULL>}, \varnothing\}}{\text{argmax}} \; p(z_t^M = m | X, z_{1:t-1}^M) & \text{if LID}(\mathbf{x}_t) \text{ is } E \end{cases} \quad (3)$$

Note that the frame-wise $\text{LID}(\mathbf{x}_t)$ is not a separate module, but rather an implicit decision within the posterior maximization over the <NULL> augmented monolingual label-to-frame alignments $Z^{M/E} = \{z_t^{M/E} \in \mathcal{V}^{M/E} \cup \{\varnothing, \text{<NULL>}\} | t = 1 \dots T\}$. This language segmentation behavior is learned by optimizing likelihoods of <NULL> masked label targets Y_{SEG}^M and Y_{SEG}^E (e.g. in Figure 1).

It follows that the bilingual $p(Z|Z^M, Z^E)$ (Eq. (2)) behaves as:

$$z_t = \begin{cases} m & \text{if } m \in \mathcal{V}^M \wedge e = \text{<NULL>} \\ e & \text{if } e \in \mathcal{V}^E \wedge m = \text{<NULL>} \\ b & \text{otherwise} \end{cases} \quad (4)$$

where m and e are the arguments maximizing $p(z_t^M | X, z_{1:t-1}^M)$ and $p(z_t^E | X, z_{1:t-1}^E)$ respectively and b is the argument maximizing $p(z_t | Z^M, Z^E, z_{1:t-1})$. If either monolingual module predicts a CS point by emitting <NULL> then the bilingual module defaults to the prediction of the other monolingual module – in other words, the first two cases of Eq. (4) expect that the language segmentation in Eq. (3) is mistake-free. The third fall-back case is considered for ambiguous language segmentation, such as if m and e are both <NULL> or both non <NULL>. This case-by-case bilingual decision is an adverse design for our zero-shot objective – models are likely to become over-reliant on the first two cases during training. Language segmentation while training on purely monolingual utterances boils down to an over-simplified *utterance-level* language identification task which may not generalize to *intra-sententially* CS test utterances. If CS point detection is expected to be tricky, then a more robust strategy should *always* expect ambiguous monolingual inputs to the final bilingual decision as in the third case of Eq. (4).

3. PROPOSED FRAMEWORK

In this section, we propose to completely remove language segmentation from monolingual modules using a transliteration-based formulation of $p(Z^{M/E}|X)$. We then present a neural model of our modified conditionally factorized approach for zero-shot CS ASR.

3.1. Modeling $p(Z^{M/E}|X)$ with Transliteration

Rather than detecting CS points at the monolingual stage in order to know which speech segments to transcribe vs. which to ignore, we propose to simply allow each monolingual module to transcribe everything. This means that for speech of a foreign language the monolingual modules are producing *transliterations*, mapping sounds to phonetically similar units within their monolingual vocabularies $\mathcal{V}^M$ and $\mathcal{V}^E$. In other words, the monolingual modules simplify from Eq. (3) to the following form (shown for Mandarin):

$$z_t^M = \underset{m \in \mathcal{V}^M \cup \{\varnothing\}}{\text{argmax}} \; p(z_t^M = m | X, z_{1:t-1}^M) \quad (5)$$

where the speech X may contain any language. This form completely removes any sense of frame-wise language identity $\text{LID}(\mathbf{x}_t)$.

To see why this modification is advantageous for zero-shot CS ASR, consider the corresponding change to the bilingual module:

$$z_t = \underset{b \in \mathcal{V}^M \cup \mathcal{V}^E \cup \{\varnothing\}}{\text{argmax}} \; p(z_t = b | Z^M, Z^E, z_{1:t-1}) \quad (6)$$

Note that this new bilingual form in Eq. (6) never defaults to the prediction of one monolingual module as in the first two cases of the previously proposed bilingual form in Eq. (4), reducing the risk of propagating errors made in the monolingual stage. In other words, the bilingual decision now determines each z_t by directly considering the conditional likelihood $p(z_t | Z^M, Z^E, z_{1:t-1})$ (Eq. (2)). This modification effectively delays CS point detection from the monolingual stage (where we would have to simultaneously transcribe and

perform frame-wise language identification per §2.2), to the bilingual stage (where transcription information is already given).

To train monolingual modules to transliterate speech segments of a foreign language, we obtain transliteration targets Y^M_{TRA} and Y^E_{TRA} using cross-lingual pseudo-labeling.[1] For instance, we pass monolingual English speech X^M to a monolingual Mandarin ASR model $\text{ASR}^M(\cdot)$ for inference and vice versa as follows:

$$Y^M_{\text{TRA}} \leftarrow \text{ASR}^M(X^E) \quad (7)$$

$$Y^E_{\text{TRA}} \leftarrow \text{ASR}^E(X^M) \quad (8)$$

where $\text{ASR}^{M/E}(\cdot)$ denote generic label-to-frame models – if we use the same architecture for pseudo-labeling as we do for our monolingual modules then these transliteration targets are cross-lingual semi-supervisions [30–33].[2] Swapping the language segmentation targets Y^M_{SEG} and Y^E_{SEG} (§2.2) for these transliteration targets Y^M_{TRA} and Y^E_{TRA} is the *only* modification required to realize our desired monolingual and bilingual module behaviors in Eq. (5) and (6).

3.2. Conditional CTC with External LM Architecture

Finally, let us consider how to construct a neural architecture for our modified conditionally factorized framework. Monolingual and bilingual label-to-frame posteriors (§2.1) may be modeled using CTC or RNN-T networks as demonstrated by prior works [23–25]. However for zero-shot CS ASR, the conditional independence assumption of CTC vs. the internal language modeling of RNN-T is a critical difference. A RNN-T based model may require internal language model (LM) adaptation [27, 35, 36] to alleviate monolingual biases while a CTC based model can be directly applied to CS test sets with optional shallow external LM fusion [37].

We therefore model monolingual, $p(Z^M|X)$ and $p(Z^E|X)$, and bilingual likelihoods, $p(Z|Z^M, Z^E)$, using CTC networks, $P_{\text{M\_CTC}}(\cdot)$, $P_{\text{E\_CTC}}(\cdot)$, and $P_{\text{B\_CTC}}(\cdot)$, as follows:

$$P_{\text{M\_CTC}}(z^M_t|X, \cancel{z^M_{1:t-1}}) = \text{SOFTMAXOUT}^M(\mathbf{h}^M_t) \quad (9)$$

$$P_{\text{E\_CTC}}(z^E_t|X, \cancel{z^E_{1:t-1}}) = \text{SOFTMAXOUT}^E(\mathbf{h}^E_t) \quad (10)$$

$$P_{\text{B\_CTC}}(z_t|\mathbf{h}^M, \mathbf{h}^E, \cancel{z_{1:t-1}}) = \text{SOFTMAXOUT}^B(\mathbf{h}^M_t + \mathbf{h}^E_t) \quad (11)$$

where speech encoders, ENCODER^M and ENCODER^E, map the speech signal, X, to latent monolingual representations, $\mathbf{h}^M = \{\mathbf{h}^M_t \in \mathbb{R}^D | t = 1, ..., T\}$ and $\mathbf{h}^E = \{\mathbf{h}^E_t \in \mathbb{R}^D | t = 1, ..., T\}$ followed by softmax normalized linear projections to monolingual or bilingual vocabularies. Then addition fusion yields a bilingual latent representation which is finally fed to the bilingual CTC. These three CTC networks are jointly optimized with an interpolated multi-task objective: $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{B\_CTC}} + (1 - \lambda_1)(\mathcal{L}_{\text{M\_CTC}} + \mathcal{L}_{\text{E\_CTC}})/2$.

During decoding, we first merge all CTC likelihoods, $P_{\text{M\_CTC}}(\cdot)$, $P_{\text{E\_CTC}}(\cdot)$, and $P_{\text{B\_CTC}}(\cdot)$, following the interpolation procedure described in Eq. (6) of [25]; we denote this merged CTC likelihood as $P_{\text{CTC}}(Z|X)$. We then jointly decode $P_{\text{CTC}}(\cdot)$ with an external bilingual LM, $P_{\text{B\_LM}}(Y)$, using the time-synchronous beam search described in [37], which approximates the following decision:

$$\underset{Y \in \{\mathcal{V}^M \cup \mathcal{V}^E\}*}{\text{argmax}} \lambda_2 (\prod_{Z \in \mathcal{Z}} \log P_{\text{CTC}}(\cdot)) + (1 - \lambda_2) \log P_{\text{B\_LM}}(\cdot) \quad (12)$$

[1] Unlike text-based transliteration [29], pseudo-labeling relies solely on the resources presumed to be available in our zero-shot CS ASR settings.

[2] We can apply transliteration to CS speech by stitching predictions corresponding to forced aligned [34] foreign segments between true native targets.

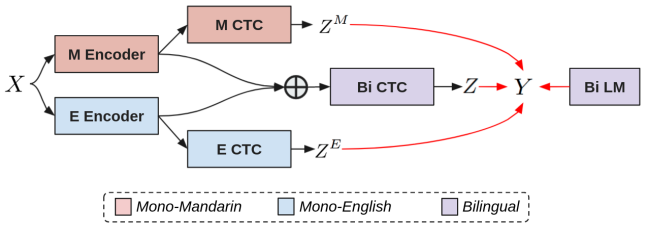

Fig. 2. Conditional CTC architecture consisting of monolingual and bilingual CTC's plus an external bilingual LM. Red lines indicate joint decoding via time-synchronous beam search.

where $\{\mathcal{V}^M \cup \mathcal{V}^E\}*$ denotes the set of all possible bilingual outputs.[3] This architecture, which we refer to as Conditional CTC, is depicted by the block-diagram in Figure 2. The monolingual modules of these Conditional CTC models can perform either language segmentation (§2.2) or transliteration (§3.1) depending on which set of monolingual targets (e.g. Figure 1) is used during training. For transliteration, we obtain Y^M_{TRA} and Y^E_{TRA} (Eq. (7) and (8)) by greedily decoding monolingual CTC models (Eq. (9) and (10)) and then applying repeat and blank removal.

4. DATA AND EXPERIMENTAL SETUP

Data: We split SEAME [38] training data into CS and monolingual (Mandarin + English) parts to create two zero-shot settings. The first setting allows 204h of monolingual labeled speech data (for ASR training) and 89k lines of unpaired CS or monolingual text data (for LM training). The second fully zero-shot setting removes the CS unpaired text data, leaving 49k lines of unpaired monolingual text data. Monolingual CTC's trained on the English and Mandarin only SEAME splits were used for cross-lingual pseudo-labeling §3.1.
Models: Models are trained using ESPnet [39]. We apply speed perturbations to up-sample training data by 3x. We combine 4000 Mandarin characters with 4000 English BPE [40] units to form the output vocabulary. Conditional CTC models have two conformer encoders [41, 42] with 12 blocks, 4 heads, 15 kernel size, 2048 feed-forward dim, 256 and attention dim. Vanilla CTC baselines with only one encoder use 512 attention dim, so all models have about 80M parameters. All models are initialized with encoder(s) pre-trained on 150h of Mandarin AISHELL-1 [43] and/or 118h of English TED-LIUM-v1 [44]. We set $\lambda_1 = 0.7$ (§3.2) during training for 40 epochs. We set $\lambda_2 = 0.8$ (§3.2) during decoding with beam size 10. We use RNN-LMs with 4 layers and 2048 dim trained for 20 epochs.
Evaluation: Systems are evaluated on the full SEAME test sets (devman and devsge) and also scored individually on the CS and monolingual portions of these sets. We measure mixed error-rate (MER) that considers word-level English and character-level Mandarin.

5. RESULTS

Table 1 presents results in three horizontal partitions where 1) all SEAME training data is allowed 2) CS speech data is removed and 3) CS speech and text data are removed; the latter two settings emulate practical zero-shot scenarios. When CS speech data is available, language segmentation is reliable and thus the transliteration-based method is not necessary (A2 vs. A3). However, once CS speech data is removed the language segmentation approach degrades 13

[3] For language segmentation variants of Conditional CTC, we do not expand hypotheses with the special <NULL> token to avoid corrupt outputs.

Table 1. Results comparing Conditional CTC models with *transliteration*-based monolingual modules to their *language segmentation* counterparts and Vanilla CTC baselines. The 1st horizontal partition shows top-line results when CS ASR training data is available. The 2nd and 3rd partitions show zero-shot results when only monolingual ASR training data is available. Performances on the full, CS only, and monolingual only splits of the SEAME test sets are measured by % mixed error rate (MER ↓). All models use CTC + LM decoding.

			ASR	LM	DEVMAN			DEVSGE		
ID	Model	Monolingual Behavior	Data	Data	Full	CS	M	Full	CS	M
A1	Vanilla CTC [37]	*No Monolingual Modules*	CS + M	CS + M	18.8	18.2	21.5	26.2	23.7	29.8
A2	Conditional CTC [24, 25]	Language Segmentation	CS + M	CS + M	**17.1**	**16.5**	19.9	**23.5**	**21.4**	**26.5**
A3	Conditional CTC (Ours)	Transliteration	CS + M	CS + M	17.3	16.9	**19.1**	24.0	22.1	26.7
B1	Vanilla CTC [37]	*No Monolingual Modules*	M	CS + M	36.6	38.9	27.0	42.5	47.0	36.1
B2	Conditional CTC [24, 25]	Language Segmentation	M	CS + M	30.1	32.0	22.0	35.7	39.7	**30.1**
B3	Conditional CTC (Ours)	Transliteration	M	CS + M	**25.2**	**26.0**	**21.9**	**31.0**	**31.5**	30.2
C1	Vanilla CTC [37]	*No Monolingual Modules*	M	M	39.1	41.6	28.4	44.8	50.0	37.3
C2	Conditional CTC [24, 25]	Language Segmentation	M	M	32.2	34.4	23.0	37.8	42.6	31.1
C3	Conditional CTC (Ours)	Transliteration	M	M	**27.3**	**28.5**	**22.6**	**32.7**	**34.0**	**30.8**

Table 2. Ablation study examining the relative importance of monolingual CTC, bilingual CTC, and bilingual LM modules during decoding as measured by % mixed error rate (MER ↓) on the devman test set. Bilingual modules are shown in blue and the most severely degraded combination (with no bilingual modules) is **bolded**.

#	Model	Decoding Likelihoods	MER(↓)
1	Cond. CTC w/ Trans.	$P_{\text{M\_CTC}}$, $P_{\text{E\_CTC}}$, $P_{\text{B\_CTC}}$, $P_{\text{B\_LM}}$	25.2
2	– Bilingual LM	$P_{\text{M\_CTC}}$, $P_{\text{E\_CTC}}$, $P_{\text{B\_CTC}}$	27.4
3	– Monolingual CTCs	$P_{\text{B\_CTC}}$, $P_{\text{B\_LM}}$	25.7
4	– Bilingual LM	$P_{\text{B\_CTC}}$	27.9
5	– Bilingual CTC	$P_{\text{M\_CTC}}$, $P_{\text{E\_CTC}}$, $P_{\text{B\_LM}}$	26.0
6	– Bilingual LM	$P_{\text{M\_CTC}}$, $P_{\text{E\_CTC}}$	**48.1**

absolute MER on both full test sets; as a result the transliteration approach outperforms by 5 absolute MER, a wide margin, owing primarily to superior performance on CS utterances (B2 vs. B3). When CS text data is also removed both variants of Conditional CTC degrade only by an additional 2 absolute MER and the gap between remains (C2 vs. C3). In all three data settings both Conditional CTC models outperform Vanilla CTC baselines.

5.1. Ablations on the Conditional CTC Model

Our Conditional CTC models consist of three types of modules: monolingual CTC's ($P_{\text{M\_CTC}}$ and $P_{\text{E\_CTC}}$), bilingual CTC ($P_{\text{B\_CTC}}$), and bilingual LM ($P_{\text{B\_LM}}$). In Table 2, we examine the relative contributions of these modules by removing each from model B3 of Table 1 during joint decoding (described in §3.2). Removing the bilingual LM (line 2) degrades performance more than removing the bilingual CTC (line 5), showing the importance of utilizing CS textual data when available. Further, note that monolingual CTCs do contribute (line 3), but are insufficient on their own (line 6). Finally, the fact performance is still reasonable without the bilingual CTC (line 5) suggests that separately trained monolingual CTCs may be directly applied to CS ASR if a CS LM is available – this direction may offer a high degree of scalability towards the long-tail of possible CS pairs and towards CS between three or more languages.

5.2. Relaxing the Zero-Shot Setting

How much CS ASR training data do we need for the originally proposed language segmentation method (§2.2) to be sufficient? The answer depends on the proximity of the particular language pair and characteristics of the dataset being used, but in our experimental setup we find that the answer is 2h of CS speech data (see Figure 3). The decreasing effectiveness of our transliteration method for increasing amounts of CS ASR training data suggests that the cross-lingual pseudo-labels are noisy to a degree. Future investigations into improving pseudo-labeling quality (e.g. via constrained decoding) may benefit this work and other related techniques which employ cross-lingual semi-supervision [30–33].

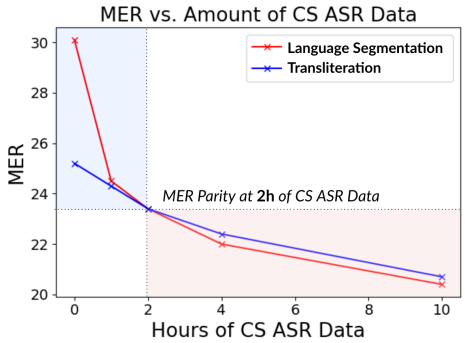

Fig. 3. Analysis on the amount of CS ASR training data required for conditional CTC with language segmentation to outperform conditional CTC with transliteration. MER(↓) on devman is shown.

6. CONCLUSION

We identify that the promising conditionally factorized joint CS and monolingual ASR framework has an acute weakness which limits its applicability to zero-shot CS ASR; the original formulation expects that each monolingual module can cleanly transcribe native speech while ignoring foreign speech. We propose a simple modification via cross-lingual pseudo-labeling to allow the monolingual modules to instead produce transliterations of foreign speech, thereby avoiding error propagation of frame-wise LID decisions. We demonstrate the effectiveness of our transliteration-based method using Conditional CTC models deployed for zero-shot Mandarin-English CS ASR. In future work, we will extend to other languages, scale beyond bilingualism, and refine our pseudo-labeling technique.

This work was supported by JSALT 2022 at JHU via Amazon, Microsoft and Google. Brian and Shinji are also supported by the HLTCOE at JHU.

7. REFERENCES

[1] M. P. Lewis, *Ethnologue: Languages of the world*. SIL international, 2009.

[2] B. Gambäck and A. Das, "Comparing the level of code-switching in corpora," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016.

[3] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 265–271.

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

[5] Y. Lu *et al.*, "Language adaptive cross-lingual speech representation learning with sparse sharing sub-networks," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 6882–6886.

[6] A. Bapna *et al.*, "Mslam: Massively multilingual joint pre-training for speech and text," *arXiv preprint arXiv:2202.01374*, 2022.

[7] X. Li *et al.*, "Asr2k: Speech recognition for around 2000 languages without audio," *INTERPSPEECH 2022*, 2022.

[8] J. Bai *et al.*, "Joint unsupervised and supervised training for multilingual asr," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[9] L. Zhou *et al.*, "A configurable multilingual model is all you need to recognize all languages," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 6422–6426.

[10] C. Zhang *et al.*, "Streaming end-to-end multilingual speech recognition with joint language identification," *INTERSPEECH 2022*, 2022.

[11] H. Seki *et al.*, "End-to-end language-tracking speech recognizer for mixed-language speech," in *ICASSP*, 2018.

[12] H. Gonen and Y. Goldberg, "Language modeling for code-switching: Evaluation, integration of monolingual data, and discriminative training," *Proc. EMNLP*, 2018.

[13] S. Zhang *et al.*, "Decoupling pronunciation and language for end-to-end code-switching asr," in *Proc. ICASSP*, 2021.

[14] G. Liu and L. Cao, "Code-switch speech rescoring with monolingual data," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[15] K. Li *et al.*, "Towards code-switching asr for end-to-end ctc models," in *Proc. ICASSP*, 2019.

[16] C. Shan *et al.*, "Component fusion: Learning replaceable language model component for end-to-end speech recognition system," in *Proc. ICASSP*, 2019.

[17] K. Taneja *et al.*, "Exploiting monolingual speech corpora for code-mixed speech recognition.," in *Interspeech*, 2019.

[18] S. Shah *et al.*, "Learning to recognize code-switched speech without forgetting monolingual speech recognition," *arXiv preprint arXiv:2006.00782*, 2020.

[19] Y. Lu *et al.*, "Bi-encoder transformer network for mandarin-english code-switching speech recognition using mixture of experts.," in *Proc. Interspeech*, 2020.

[20] X. Zhou *et al.*, "Multi-encoder-decoder transformer for code-switching speech recognition," *Interspeech*, 2020.

[21] S.-P. Chuang, T.-W. Sung, and H.-y. Lee, "Training code-switching language model with monolingual data," in *Proc. ICASSP*, 2020.

[22] S. Dalmia *et al.*, "Transformer-transducers for code-switched speech recognition," in *Proc. ICASSP*, 2021.

[23] B. Yan *et al.*, "Joint modeling of code-switched and monolingual asr via conditional factorization," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6412–6416.

[24] J. Tian *et al.*, "Lae: Language-aware encoder for monolingual and multilingual asr," *INTERSPEECH 2022*, 2022.

[25] T. Song *et al.*, "Language-specific characteristic assistance for code-switching speech recognition," *INTERSPEECH 2022*, 2022.

[26] A. Graves *et al.*, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006.

[27] A. Graves, "Sequence Transduction with Recurrent Neural Networks," in *Proc. ICML*, 2012.

[28] H. Liu *et al.*, "End-to-end language diarization for bilingual code-switching speech.," in *Interspeech*, 2021.

[29] K. Knight and J. Graehl, "Machine transliteration," *Computational Linguistics*, vol. 24, no. 4, pp. 599–612, 1998.

[30] P. Jyothi and M. Hasegawa-Johnson, "Transcribing continuous speech using mismatched crowdsourcing," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[31] S. Thomas, K. Audhkhasi, and B. Kingsbury, "Transliteration based data augmentation for training multilingual asr acoustic models in low resource settings.," in *INTERSPEECH*, 2020, pp. 4736–4740.

[32] J. Billa, "Leveraging Non-Target Language Resources to Improve ASR Performance in a Target Language," in *Proc. Interspeech 2021*, 2021, pp. 2581–2585.

[33] L. Lugosch *et al.*, "Pseudo-labeling for massively multilingual speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 7687–7691.

[34] L. Kürzinger *et al.*, "Ctc-segmentation of large corpora for german end-to-end speech recognition," in *International Conference on Speech and Computer*, Springer, 2020, pp. 267–278.

[35] Z. Meng *et al.*, "Internal language model training for domain-adaptive end-to-end speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 7338–7342.

[36] W. Zhou *et al.*, "On language model integration for rnn transducer based speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 8407–8411.

[37] A. Y. Hannun *et al.*, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns," *arXiv preprint arXiv:1408.2873*, 2014.

[38] D.-C. Lyu *et al.*, "Seame: A mandarin-english code-switching speech corpus in south-east asia," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[39] S. Watanabe *et al.*, "Espnet: End-to-end speech processing toolkit," *Proc. Interspeech*, 2018.

[40] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *Proc. ACL*, 2015.

[41] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Interspeech*, 2020.

[42] P. Guo *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *Proc. ICASSP*, 2021.

[43] H. Bu *et al.*, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Proc. O-COCOSDA*, IEEE, 2017.

[44] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: An automatic speech recognition dedicated corpus," in *Proc. LREC*, 2012.