applied sciences

MDPI

Article

Improving Transformer Based End-to-End Code-Switching Speech Recognition Using Language Identification

Zheying Huang [1,2], Pei Wang [3], Jian Wang [1], Haoran Miao [1,2], Ji Xu [1,2] and Pengyuan Zhang [1,2,*]

[1] Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; huangzheying@hccl.ioa.ac.cn (Z.H.); wangjian@hccl.ioa.ac.cn (J.W.); miaohaoran@hccl.ioa.ac.cn (H.M.); xuji@hccl.ioa.ac.cn (J.X.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] National Internet Emergency Center, Beijing 100094, China; wangpei@cert.org.cn
* Correspondence: zhangpengyuan@hccl.ioa.ac.cn

Abstract: A Recurrent Neural Networks (RNN) based attention model has been used in code-switching speech recognition (CSSR). However, due to the sequential computation constraint of RNN, there are stronger short-range dependencies and weaker long-range dependencies, which makes it hard to immediately switch languages in CSSR. Firstly, to deal with this problem, we introduce the CTC-Transformer, relying entirely on a self-attention mechanism to draw global dependencies and adopting connectionist temporal classification (CTC) as an auxiliary task for better convergence. Secondly, we proposed two multi-task learning recipes, where a language identification (LID) auxiliary task is learned in addition to the CTC-Transformer automatic speech recognition (ASR) task. Thirdly, we study a decoding strategy to combine the LID into an ASR task. Experiments on the SEAME corpus demonstrate the effects of the proposed methods, achieving a mixed error rate (MER) of 30.95%. It obtains up to 19.35% relative MER reduction compared to the baseline RNN-based CTC-Attention system, and 8.86% relative MER reduction compared to the baseline CTC-Transformer system.

Keywords: speech recognition; code-switching; Transformer; multi-task learning; language identification

Citation: Huang, Z.; Wang, P.; Wang, J.; Miao, H.; Xu, J.; Zhang, P. Improving Transformer Based End-to-End Code-Switching Speech Recognition Using Language Identification. *Appl. Sci.* **2021**, *11*, 9106. https://doi.org/10.3390/app11199106

Academic Editor: José A. González-López

Received: 16 August 2021
Accepted: 24 September 2021
Published: 30 September 2021

1. Introduction

Code-switching (CS) speech is defined as speech which contains more than one language within an utterance [1]. With the development of globalization, this multilingual phenomenon has become increasingly common in real life, so the research on this phenomenon has attracted growing attention. Traditionally, research on a Gaussian mixture model based hidden Markov models (GMM-HMM) and deep neural network based hidden Markov model (DNN-HMM) framework for code-switching speech recognition (CSSR) [2,3] focuses on two challenges: lack of language model training data at CS points, co-articulation effects between phonemes at CS points. Therefore, it is difficult to reliably estimate the probability of word sequences where CS appears and to model the phonemes at CS points. To address the former challenge, statistical machine translation is utilized to manufacture artificial CS training text [4]. Several methods are proposed to improve the performance of language modeling to CS speech: recurrent neural network language models and factored language models with the integration of part-of-speech tag, language information, or syntactic and semantic features [5–7]. To address the latter challenge, speaker adaptation, phone sharing and phone merging are applied [4]. Recently, an End-to-End (E2E) approach for the CSSR task has attracted increasing interest [8–11]. By predicting graphemes or characters directly from acoustic information without predefined alignment, the E2E system can considerably reduce the effort of building automatic speech recognition (ASR) systems. In the mean time, the need of expert linguistic knowledge is also eliminated,

which makes it an attractive choice for CSSR. Previous works mainly adopted two types of E2E methods in the CSSR task: connectionist temporal classification (CTC) [12] and the RNN-based attention method [13–15]. The CTC objective function simplifies acoustic modeling into learning a RNN over pairs of speech and context-independent (CI) label sequences, without requiring a frame-level alignment of the target labels for a training utterance [16]. The RNN-based attention method consists of an RNN encoder and an attention-based RNN decoder, which maps acoustic speech into a high-level representation and recognizes symbols conditioned on previous predictions, respectively [13–15]. A joint CTC-Attention multi-task learning model is presented to combine the benefit of both types of systems [17,18]. However, RNN remains as the sequential computation constraint. Therefore, stronger short-range dependencies and weaker long-range dependencies exist in encoder outputs and decoder outputs, which makes it hard to immediately switch languages in CSSR. Recently, the Transformer [19,20] has achieved state-of-the-art performances in many monolingual ASRs [21]. It transduces sequential data with its self-attention mechanism, which replaces the RNN in previous works [15,18]. Since self-attention mechanisms utilize global context—that is, all frames learn time dependency inside the input sequence to achieve sequence transduction in parallel—information transmission is the same for each location to draw global dependencies, which makes it possible to switch more freely at CS points. Therefore, in this paper, we apply a joint CTC-Transformer framework for CSSR. Then, we study different multi-task learning recipes, where a language identification (LID) auxiliary task is learned in addition to the ASR task. Lastly, we study a decoding strategy to combine the LID information into ASR. All of our experiments are conducted on the SEAME corpus.

The paper is organized as follows. Related works are presented in Section 2. The multi-task learning recipes and LID joint decoding are studied in Section 3. Experimental setups and results analysis are described in Section 4. Some conclusions are drawn in Section 5.

2. Related Work

2.1. Transformer Based E2E Architecture

The Transformer contains an Encoder network and a Decoder network [21–23]. Both the Encoder and the Decoder consist of several layers stacked, as shown in Figure 1a,b, respectively. The Encoder transforms the input features $X = [x_1, \ldots, x_T]$ into a sequence of encoded features $H_e = [h_{e,1}, \ldots, h_{e,T}]$, as follows:

$$\begin{aligned} H_0 &= CNN(X) + PE \\ H_i^{'} &= H_i + MHA_i(N(H_i), N(H_i), N(H_i)) \\ H_{i+1} &= H_i^{'} + FF_i(N(H_i^{'})), \end{aligned} \tag{1}$$

where $i = 0, \ldots, e-1$, e is the number of encoder layers, $CNN(\cdot)$ is a convolution network, PE is positional encoding, $MHA(\cdot)$ is a multi-head self-attention mechanism, and FF_i is a positionwise fully connected feed-forward network. In this work, layer normalization $N(\cdot)$ is employed before each sub-layer . The Decoder receives the encoded features H_e and the label sequence $Y[0 : l-1]$ to emit the probabilities of the Decoder output units set $\mathcal{Y}[l] = [y_l^1, \ldots, y_l^{d^{units}}]$ of the l-th step, and then determine the subsequent $[\hat{y}_1, \ldots, \hat{y}_l]$, as Equations (2)–(4):

$$E_{dec} = Embed(Y[0 : l-1]). \tag{2}$$

$Embed(\cdot)$ is an embedding layer that transforms a sequence of labels $Y[0 : l-1]$ into a sequence of learnable vectors $E_{dec} \in \mathbb{R}^{l \times d^{att}}$, and d^{att} is the dimension of attentions.

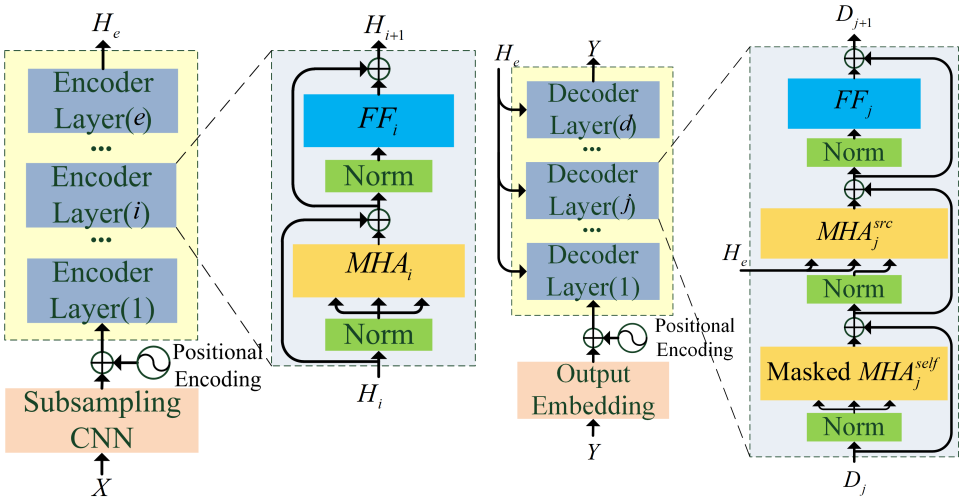
$$
\begin{aligned}
D_0 &= E_{dec} + PE \\
D_j^{'} &= D_j + MHA_j^{self}(N(D_j), N(D_j), N(D_j)) \\
D_j^{''} &= D_j^{'} + MHA_j^{src}(N(D_j^{'}), H_e, H_e) \\
D_{j+1} &= D_j^{''} + FF_j(N(D_j^{''})),
\end{aligned}
\tag{3}
$$

where $j = 0, \ldots, d-1$, d is the number of decoder layers.

$$
\begin{aligned}
&[p_{att}(\mathcal{Y}[1]), \ldots, p_{att}(\mathcal{Y}[l])] = softmax(D_d W^o) \\
&\hat{y}_{l'} = \underset{y_{l'} \in \mathcal{Y}}{\arg\max}\, p_{att}(\mathcal{Y}[l']) \qquad l' \in 1, \ldots, l
\end{aligned}
\tag{4}
$$

where learnable weight matrices $W^o \in \mathbb{R}^{d^{att} \times d^{units}}$ belong to the output linear layer, and d^{units} is the number of output units.

In this work, in the training stage, ground truth sequence ($Y[0:l-1] = [y_0, \ldots, y_{l-1}]$) is adopted as the input of the embedding layer, while in the decoding stage, predicted sequence ($Y[0:l-1] = [\hat{y}_0, \ldots, \hat{y}_{l-1}]$) is adopted.



(a) Encoder **(b)** Decoder

Figure 1. Encoder and Decoder.

2.2. Self-Attention

Scaled Dot-Product Attention is commonly used as an attention function for the self-attention mechanism [21]. The input consists of queries(Q) and keys(K) of dimension d_k, and values(V) of dimension d_v. Scaled Dot-Product Attention is computed as Equation (5):

$$
Att(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{5}
$$

To allow the model to jointly attend to information from different representation sub-spaces at different positions, [21] extends Equation (5) to multi-head attention Equation (6):

$$
\begin{aligned}
MHA(Q, K, V) &= Concat(hd_1, \ldots, hd_h)W \\
hd_i &= Att(QW_i^Q, KW_i^K, VW_i^V),
\end{aligned}
\tag{6}
$$

where h is the number of attention heads, $W_i^Q, W_i^K \in \mathbb{R}^{d_{att} \times d_k}, W_i^V \in \mathbb{R}^{d_{att} \times d_v}$ and $W \in \mathbb{R}^{(h \times d_v) \times d_{att}}$ are learnable weight matrices.

3. Methods

3.1. CTC-Transformer Based CSSR Baseline System

We adopted a Transformer framework to build the CSSR system. However, the Transformer takes many more epochs for the monolingual ASR task to converge, let alone the CSSR task, which has many more model units. Inspired by [20], we added a CTC objective function to train the encoder of the Transformer. CTC helps the Transformer to converge with the forward–backward algorithm, enforcing a monotonic alignment between input features and output labels. The architecture of the CTC-Transformer baseline system is indicated in Figure 2.

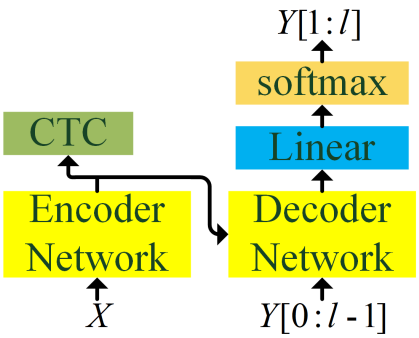


Figure 2. CTC-Transformer baseline architecture.

Specifically, let X be the input acoustic sequence, Y be the output label sequence comprising Mandarin modeling units or English modeling units, let $L_{ctc}(Y|X)$ be the CTC objective loss [16] and $L_{att}(Y|X)$ be the attention-based objective loss. The $L_{att}(Y|X)$ loss is the cross entropy of predicted $\hat{Y}$ and ground truth Y. The combination of $L_{ctc}(Y|X)$ and $L_{att}(Y|X)$ is adopted for the ASR task:

$$L_{ASR}(Y|X) = (1-\alpha)L_{att}(Y|X) + \alpha L_{ctc}(Y|X), \tag{7}$$

where α is a hyperparameter.

We chose a Chinese character as the Mandarin acoustic modeling unit, as it is the most common choice for E2E Mandarin ASR and it has shown a state-of-the-art performance in Mandarin ASR [24,25]. As for English, we chose the subword as the English unit. We adopted Byte Pair Encoding (BPE) [26] to generate subword units.

3.2. CSSR Multi-Task Learning with LID

In CSSR, modeling units belonging to different languages but with similar pronunciation are easy to confuse. Meanwhile, the language information was not used explicitly during training. LID is a process by which a computer analyzes and processes speech to determine which language it belongs to. So, we believe that adopting LID prediction as an auxiliary task can improve the CSSR performance. We sent the feature output from the encoder to the decoder, and the decoder output to the LID sequence corresponding to the feature sequence. The LID task and the ASR task share the same encoder, therefore we call it multitask learning. The objective loss $L_{LID}(Z|X)$ was added to extend the multi-task learning (MTL) objective loss:

$$L_{MTL} = (1-\beta)L_{ASR}(Y|X) + \beta L_{LID}(Z|X). \tag{8}$$

The $L_{LID}(Z|X)$ loss is the cross entropy of the predicted LID label sequence $\hat{Z}$ and the ground truth LID label sequence Z. In this work, each LID label z_l corresponds to an

ASR label y_l. So the length of the LID label sequence Z is the same as that of the ASR label sequence Y. An example is shown in Figure 3.

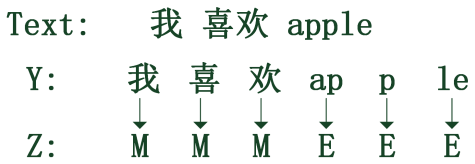


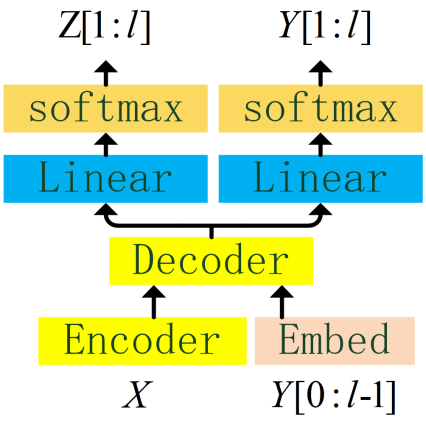
Figure 3. An example: Y label sequence corresponding to Z label sequence.

We used label '*E*' for English, and label '*M*' for Mandarin. We used label '*N*' for nonverbal, such as noise, laugh and so on, but we do not mention '*N*' below, as it is not important in this study. Because the length of the Z sequence is the same as that of the Y sequence, and is inspired by the decoding method of the Y sequence, we used a similar structure to predict the Z sequence. In the training stage, the Decoder of the ASR task used H_e and the information of the label sequence $(y_0, \ldots, y_{l-1})$ to predict $(y_1, \ldots, y_l)$. We studied what label sequence should participate to predict the next LID label. Specifically, we propose two strategies to implement the Z label prediction task.

- LID label sequence (LLS): as indicated in Figure 4a, the LID predictor receives the encoded features H_e and the LID label sequence $(z_0, \ldots, z_{l-1})$ to predict $(z_1, \ldots, z_l)$. The LID predictor does not share the embedding layer with ASR predictor, and it has its own embedding layer:

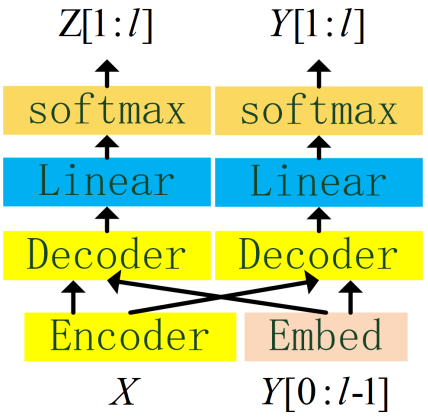
$$E_Z = Embed_{LID}(Z[0 : l-1]). \tag{9}$$

  $Embed_{LID}(\cdot)$ transforms a sequence of labels $Z[0 : l-1]$ into a sequence of learnable vectors $E_Z \in \mathbb{R}^{l \times d^{att}}$.
- ASR label sequence (ALS): just like LLS, except that the LID predictor does not receive the LID label sequence, but the ASR label sequence $(y_0, \ldots, y_{l-1})$, and in fact, the LID predictor shares the embedding layer with the ASR predictor. ALS can be implemented in two structures, one is the LID task sharing the decoder with the ASR task (ALS-share), the other is not sharing (ALS-indep), as indicated in Figure 4b,c, respectively.

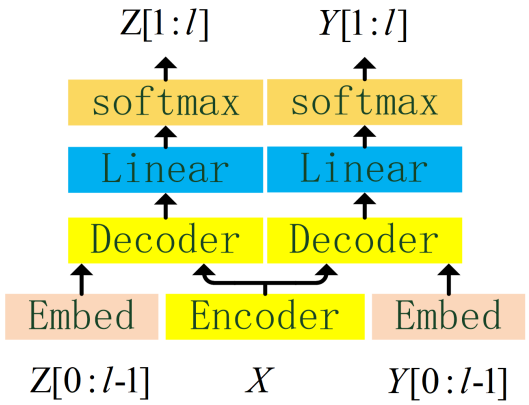


(**a**) ALS-share

Figure 4. *Cont.*



(**b**) ALS-indep



(**c**) LLS

Figure 4. Frameworks for different MTL strategies.

On this basis, inspired by [20], we also operate another set of experiments adding joint CTC training for the LID task.

3.3. CSSR Joint Decoding with LID

There are similar pronunciation units across two languages; therefore, units of one language may be incorrectly identified as units with similar pronunciation to that of another language. Figure 5 shows an example in our experiment.

Figure 5. An example of units in one language being incorrectly identified as units in another language.

This problem may be related to the fact that LID is not used in decoding. As indicated in Figure 6, we integrate LID into the decoding process by conditionally modifying the ASR output probabilities $p_{att}(y_l^k)$ with the LID output probabilities $p_{att}(z_l^{m_k})$:

- Firstly, ASR branch decoding and LID branch decoding are carried out simultaneously to obtain the ASR label $\hat{y}_l$ and the LID label $\hat{z}_l$ of the l-th step;
- $\hat{y}_l$ can be uniquely mapped to $\hat{z}_l^{'}$, since there is no intersection between the Chinese modeling unit set and the English modeling unit set;
- If $\hat{z}_l^{'}$ is not in $\{'E','M'\}$ or $\hat{z}_l$ is not in $\{'E','M'\}$, $p_{att}(y_l^k)$ will not be modified.

- If both $\hat{z}_l^{'}$ and $\hat{z}_l$ are in $\{'E','M'\}$, and $\hat{z}_l^{'}$ is different from $\hat{z}_l$, then, modification and normalization will be added to $p_{att}(y_l^k)$.

$$p_{att}^{'}(y_l^k) = \frac{p_{att}(y_l^k) \times p_{att}(z_l^{m_k})}{\sum_k p_{att}(y_l^k) p_{att}(z_l^{m_k})}$$
$$p_{att}(y_l^k) = p_{att}^{'}(y_l^k), \tag{10}$$

where $k = 1, \ldots, d^{units}$, d^{units} is the number of ASR decoder output units, and y_l^k is the k-th output unit of l-th step, and m_k is the k-th ASR task output units mapping to the LID task output units; therefore, $z_l^{m_k}$ is the m_k-th output unit of l-th step of the LID task, $p_{att}(y_l^k)$ is the probability of the k-thASR output unit of the l-th step, $p_{att}(z_l^{m_k})$ is the probability of the m_k-th LID output unit of the l-th step.

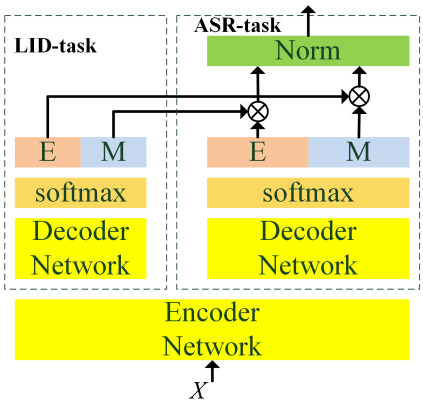


Figure 6. Frameworks for joint LID and ASR decoding.

If $z_l^{m_k}$ is different to the corresponding language of y_l^k, then, after normalization, the value of $p_{att}(y_l^k)$ will decrease, which can reduce the probability of selecting the current error unit.

4. Experiments and Results

4.1. Data

We conduct experiments on the SEAME(South East Asia Mandarin–English) corpus [27], which was developed for spontaneous Mandarin–English code-switching research. We divide the SEAME corpus into three sets (train, development and test) by proportionally sampling speakers. The detailed statistics of the corpus division are presented in Table 1.

Table 1. Statistics of the SEAME corpus.

	Train Set	Dev Set	Test Set	Total
speakers	145	2	8	155
Duration (h)	107	1.86	6.38	115.23
utterances	100802	1596	6276	108674

4.2. Baseline Setup

Firstly, we replicate two other baseline systems based on different frameworks—GMM-HMM [4] and RNN-based CTC-Attention [17]. Since the partition of training/development/test sets is not identical, there is a slight gap between our results and the original, but within the allowable range. A "big model" is commonly suggested for the Transformer [19,20] in monolingual ASR, but we find it unsuitable for CSSR due to insufficient CS data obtained and CSSR output units that are too large. In this work, we chose a "smaller model" for the Transformer($d^{att} = 256$, $e = 6$, $d = 3$). The input speech is represented as a sequence

of a 40-dim filterbank feature. The filterbank feature is firstly subsampled by a two layer time-axis convolutional neural network with *ReLU* activation (stride size is 2, kernel size is 3, the number of channels is 256). The loss weight α for the CTC joint training is set to 0.3. To prevent training from overfitting to the training set, label smoothing [28] with a penalty of 0.1 is applied. For Mandarin modeling units, a set of 2639 Chinese characters is used, covering the Chinese characters that appear in the training text. For English, a set of 200 subwords is used, which is trained on English segments of the training set using the BPE method [26].

As shown in Table 2, the CTC-Transformer baseline had a mixed error rate (MER) of 33.96%, better than that of the RNN-based CTC-Attention baseline (38.38%) and the GMM-HMM baseline (39.6%).

Table 2. MER(%) of different framework CSSR baseline systems.

Baseline System	MER(%)
GMM-HMM	39.6
RNN-based CTC-Attention	38.38
CTC-Transformer	**33.96**

4.3. Effects of Different MTL Strategies for Combination of LID and ASR

We conducted experiments using a different choice of MTL weight β, and we chose the best weight $\beta = 0.1$ in the following experiments. As shown in Table 3, LLS and ALS-share achieve an equivalent effect, both better than the CTC-Transformer baseline, consistent with our expectations. However, the effect of ALS-indep is worse, therefore we did not use ALS-indep for subsequent experiments.

Table 3. MER(%) of ASR task corresponding to three MTL implementations: LLS, ALS-share and ALS-indep.

MTL Strategy	LID-CTC	MER(%)
no LID task	no	33.96
LLS	no	32.24
	yes	**31.37**
ALS-share	no	32.79
	yes	31.84
ALS-indep	no	34.4

To add joint CTC training for LID, the LID CTC weight is set to 0.3, the same as the ASR CTC weight (α = 0.3). As shown in Table 3, the LID auxiliary task with joint CTC training can better assist the ASR main task, because CTC can learn to align the speech feature and the LID label sequence explicitly.

4.4. Effects of Joint Decoding with LID

As shown in Table 4, joint LID decoding improves LLS, but has no effect on ALS-share. For ALS-share, the LID task and the ASR task use the same decoder, which makes the result of the LID task more closely related to that of the ASR task. In contrast, as for LLS, the LID task is relatively independent of the ASR task. Therefore, LID decoder training in LLS is more capable of correcting language errors in the ASR task. Under the configuration of the LLS method, CTC joint training for the LID task and LID joint decoding, the final system achieves an MER of (30.95%), obtaining up to a (19.35%) and (8.86%) relative MER reduction compared to the RNN-based CTC-Attention baseline system (38.38%) and the CTC-Transformer baseline system (33.96%) respectively.

Table 4. MER(%) of ASR task (using CTC-Transformer architecture), corresponding to joint decoding with LID or not.

MTL Strategy	LID-CTC	LID Joint Decoding	MER(%)
LLS	yes	no	31.37
LLS	yes	yes	**30.95**
ALS-share	yes	no	31.84
ALS-share	yes	yes	31.84

5. Conclusions

In this work, we introduce a CTC-Transformer based E2E model for Mandarin–English CSSR, which outperforms most of the traditional systems on the SEAME corpus. As for the inclusion of LID, we propose two LID multi-task learning strategies: LLS and ALS (ALS-share and ALS-indep). LLS and ALS-share have a comparable promotion effect. Furthermore, we study a decoding strategy to combine the LID information into the ASR task and it slightly improves the performance in the case of LLS. The final system with the proposed methods achieved an MER of 30.95%, obtaining up to a 19.35% and 8.86% relative MER reduction compared to the RNN-based CTC-Attention baseline system (38.38%) and the CTC-Transformer baseline system (33.96%), respectively.

Author Contributions: Conceptualization, Z.H. and J.X.; methodology, Z.H.; software, Z.H. and H.M.; validation, Z.H., J.X. and P.W.; formal analysis, Z.H., J.X. and H.M.; investigation, Z.H. and J.W.; resources, Z.H. and H.M.; data curation, Z.H.; writing—original draft preparation, Z.H.; writing—review and editing, J.X.; visualization, Z.H. and H.M.; supervision, P.Z.; project administration, P.Z.; Funding acquisition, P.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially founded by the National Key Research and Development Program (No. 2019QY1805), the National Natural Science Foundation of China (No. 61901466), and the High Tech Project (No. 31513070501).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The SEAME dataset was analyzed in this study. These data can be found: (https://catalog.ldc.upenn.edu/LDC2015S04, accessed on 15 April 2015).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Auer, P. *Code-Switching in Conversation: Language, Interaction and Identity*; Routledge: London, UK, 1998.
2. Yilmaz, E.; Heuvel, H.V.D.; van Leeuwen, D.A. *Acoustic and Textual Data Augmentation for Improved ASR of Code-Switching Speech*; INTERSPEECH: Hyderabad, India, 2–6 September 2018; pp. 1933–1937.
3. Guo, P.; Xu, H.; Lei, X.; Chng, E.S. *Study of Semi-Supervised Approaches to Improving English-Mandarin Code-Switching Speech Recognition*; INTERSPEECH: Hyderabad, India, 2–6 September 2018.
4. Vu, N.T.; Lyu, D.C.; Weiner, J.; Telaar, D.; Schlippe, T.; Blaicher, F.; Chng, E.S.; Schultz, T.; Li, H. A first speech recognition system for Mandarin-English code-switch conversational speech. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan, 25–30 March 2012; pp. 4889–4892.
5. Adel, H.; Vu, N.T.; Kirchhoff, K.; Telaar, D.; Schultz, T. Syntactic and Semantic Features For Code-Switching Factored Language Models. *IEEE Trans. Audio Speech Lang. Process.* **2015**, *23*, 431–440. [CrossRef]
6. Adel, H.; Vu, N.T.; Schultz, T. Combination of Recurrent Neural Networks and Factored Language Models for Code-Switching Language Modeling. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–10 August 2013; Volume 2, pp. 206–211.
7. Adel, H.; Kirchhoff, K.; Telaar, D.; Vu, N.T.; Schlippe, T.; Schultz, T. Features for factored language models for code-Switching speech. In Proceedings of the Spoken Language Technologies for Under-Resourced Languages, St. Petersburg, Russia, 14–16 May 2014; pp. 32–38.
8. Lu, Y.; Huang, M.; Li, H.; Jiaqi, G.; Qian, Y. *Bi-encoder Transformer Network for Mandarin-English Code-Switching Speech Recognition Using Mixture of Experts*; INTERSPEECH: Shanghai, China, 25–29 October 2020; pp. 4766–4770.

9. Metilda Sagaya Mary, N.J.; Shetty, V.M.; Umesh, S. Investigation of methods to improve the recognition performance of tamil-english code-switched data in transformer framework. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Virtual Conference, 4–9 May 2020; pp. 7889–7893.
10. Dalmia, S.; Liu, Y.; Ronanki, S.; Kirchhoff, K. Transformer-transducers for code-switched speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, ON, Canada, 6–11 June 2021; pp. 5859–5863.
11. Tan, Z.; Fan, X.; Zhu, H.; Lin, E. Addressing accent mismatch In Mandarin-English code-switching speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, ON, Canada, 6–11 June 2021; pp. 8259–8263.
12. Kim, S.; Hori, T.; Watanabe, S. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, 5–9 March 2017; pp. 4835–4839.
13. Prabhavalkar, R.; Rao, K.; Sainath, T.N.; Bo, L.; Johnson, L.; Jaitly, N. *A Comparison of Sequence-to-Sequence Models for Speech Recognition*; INTERSPEECH: Hyderabad, India, 2017; pp. 939–943.
14. Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-Based Models for Speech Recognition. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 577–585.
15. Chan, W.; Lane, I. *On Online Attention-Based Speech Recognition and Joint Mandarin Character-Pinyin Training*; INTERSPEECH: San Francisco, CA, USA, 8–12 September 2016; pp. 3404–3408.
16. Graves, A.; Fernández, S.; Gomez, F.J.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the Machine Learning, Proceedings of the 23rd International Conference, Pittsburgh, PA, USA, 25–29 June 2006; Volume 148, pp. 369–376.
17. Luo, N.; Jiang, D.; Zhao, S.; Gong, C.; Zou, W.; Li, X. Towards End-to-End Code-Switching Speech Recognition. *arXiv* **2018**, arXiv:1810.13091.
18. Zeng, Z.; Khassanov, Y.; Pham, V.T.; Xu, H.; Chng, E.S.; Li, H. On the End-to-End Solution to Mandarin-English Code-switching Speech Recognition. *arXiv* **2018**, arXiv:1811.00241.
19. Dong, L.; Xu, S.; Xu, B. Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, AB, Canada, 15–20 April 2018; pp. 5884–5888.
20. Karita, S.; Soplin, N.E.Y.; Watanabe, S.; Delcroix, M.; Ogawa, A.; Nakatani, T. *Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration*; INTERSPEECH: Graz, Austria, 15–19 September 2019; pp. 1408–1412.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 3–9 December 2017; pp. 5998–6008.
22. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N. End-to-end object detection with transformers. *arXiv* **2020**, arXiv:2005.12872.
23. Choromanski, K.; Likhosherstov, V.; Dohan, D.; Song, X.; Davis, J.; Sarlos, T.; Belanger, D.; Colwell, L.; Weller, A. Masked language modeling for proteins via linearly scalable long-context transformers. *arXiv* **2020**, arXiv:2006.03555.
24. Zou, W.; Jiang, D.; Zhao, S.; Li, X. A comparable study of modeling units for end-to-end Mandarin speech recognition. *arXiv* **2018**, arXiv:1805.03832.
25. Zhou, S.; Dong, L.; Xu, S.; Xu, B. A Comparison of Modeling Units in Sequence-to-Sequence Speech Recognition with the Transformer on Mandarin Chinese. *arXiv* **2018**, arXiv:1805.06239.
26. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. *arXiv* **2015**, arXiv:1508.07909.
27. Lyu, D.; Tan, T.P.; Chng, E.; Li, H. *SEAME: A Mandarin-English Code-Switching Speech Corpus in South-East Asia*; INTERSPEECH: Makuhari, Japan, 26–30 September 2010; pp. 1986–1989.
28. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.