INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Towards Context-Aware End-to-End Code-Switching Speech Recognition

Zimeng Qiu[†], *Yiyuan Li*[‡], *Xinjian Li*[‡], *Florian Metze*[‡], *William M. Campbell*[†]

† Amazon Alexa AI
‡ Carnegie Mellon University
{zimengqi, cmpw}@amazon.com {yiyuanli, xinjianl, fmetze}@andrew.cmu.edu

Abstract

Code-switching (CS) speech recognition is drawing increasing attention in recent years as it is a common situation in speech where speakers alternate between languages in the context of a single utterance or discourse. In this work, we propose Hierarchical Attention-based Recurrent Decoder (HARD) to build a context-aware end-to-end code-switching speech recognition system. HARD is an attention-based decoder model which employs a hierarchical recurrent network to enhance model's awareness of previous generated historical sequence (sub-sequence) at decoding. This architecture has two LSTMs to model encoder hidden states at both the character level and sub-sequence level, therefore enables us to generate utterances that switch between languages more precisely from speech. We also employ language identification (LID) as an auxiliary task in multi-task learning (MTL) to boost speech recognition performance. We evaluate the effectiveness of our model on the SEAME dataset, results show that our multi-task learning HARD (MTL-HARD) model improves over the baseline Listen, Attend and Spell (LAS) model by reducing character error rate (CER) from 29.91% to 26.56% and mixed error rate (MER) from 38.99% to 34.50%, and case study shows MTL-HARD can carry historical information in the sub-sequences.
Index Terms: speech recognition, code-switching

1. Introduction

Code-switching (CS) is a phenomenon when speakers alternate between two or more languages in the context of a single utterance or discourse. Previous code-switching works focus on the interplay between multiple languages by modelling the linguistic structure of code-switched utterances [1, 2, 3]. As machine learning approaches become increasingly popular, statistical methods are also applied to code-switching tasks [4, 5]. While DNN-HMM based ASR models are widely applied to code-switching speech recognition [6, 7], its weakness in great model complexity and being unable to be optimized end-to-end motivate researchers to explore End-to-End (E2E) frameworks.

Similar E2E strategies are pursued to resolve Mandarin-English code-switching speech recognition in [8, 9]. They both adopt hybrid CTC and attention-based networks. Unlike DNN-HMM based approaches, they don't require efforts in lexicon modeling and the entire system comprises compactly connected neural networks that can be jointly learned from scratch. They also employ a multi-task learning (MTL) [10] approach that enhances the E2E ASR system with language identification (LID) as the auxiliary task to boost the performance of ASR.

However, these approaches fail to incorporate historical information in the decoding phase. Input-feeding [11, 12] is applied to basic attention-based speech models to alleviate this issue [13]. Specifically, they investigate performances of Listen, Attend and Spell (LAS) model [14] with input-feeding and

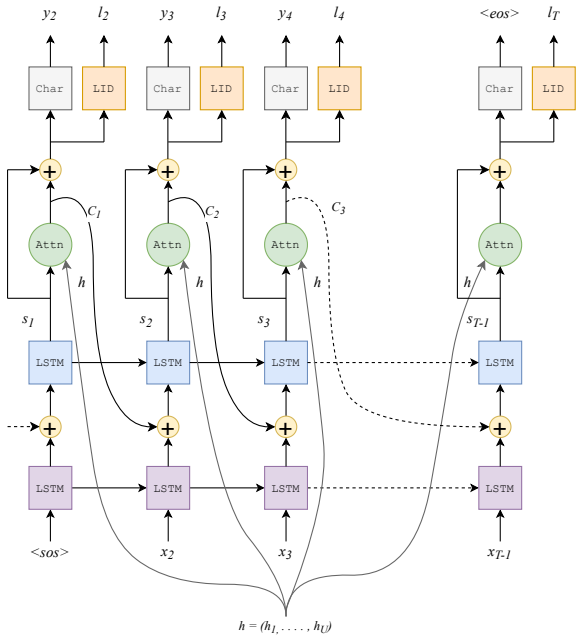

Figure 1: *Architecture of MTL-HARD decoder model. The first LSTM encodes the sub-sequence history and captures longer context, the second LSTM takes the character-level information and combines it with first context.*

proves that input-feeding can outperform baseline in terms of CER on code-switching corpora. Though the input-feeding keeps information about previous alignment decisions between hidden states of the encoder and decoder, it does not overcome errors caused by short memory of previous generated sub-sequence. This is more crucial when it comes to code-switching scenarios because the distribution of tokens varies between code-switched and monolingual generated history.

To tackle this issue, we propose Hierarchical Attention-based Recurrent Decoder (HARD), an attention-based decoder model which employs a hierarchical recurrent network to enhance model's ability of being aware of previous generated history sequence at decoding. This architecture has two LSTMs to model encoder hidden states as shown in Figure.1: *Character LSTM* is to encode character level information and the *Sub-sequence LSTM* is to encode sub-sequence level information and is expected to carry the history context within each utterance. Those two-level LSTMs enable our model to generate utterances that switch between languages more precisely from speech. Additionally, in order to help the model better decide which language to predict for next character, we also employ language identification (LID) as an auxiliary task in multi-task

http://dx.doi.org/10.21437/Interspeech.2020-1980

learning and jointly train the character predictor and LID predictor. By sharing previous layers between tasks of character prediction and LID prediction, the character predictor can benefit from having more information of languages that previous generated characters belong to.

We evaluate the effectiveness of our model on the SEAME dataset [15]. The character error rate (CER) is improved from 29.91% to 26.56% and mixed error rate (MER, Chinese character and English word respectively) from 38.99% to 34.50% compared to the Listen, Attend and Spell (LAS) model [14].

Our contributions are mainly the following: we model each generated character as a signal of new sub-sequence, and employ a hierarchical recurrent network [16] in attention-based decoder to capture longer history and code-switching information at decoding, which alleviates the issue that the decoder being confused where to switch language and hard to generate reasonable results when code-switching occurs frequently within a single utterance. Additionally, we also employ language identification (LID) as an auxiliary task in multi-task learning and jointly train the character predictor and LID predictor with previous layers shared.

The rest of paper are organized as follows: Section 2 illustrates our proposed models; Section 3 describes the experimental setup. We share experiment results and code-switching intentions mining in Section 4. Section 5 concludes the work and addresses the future work.

2. Models

2.1. LAS Baseline

We use the state-of-the-art end-to-end speech recognition model *Listen, Attend and Spell* (LAS) model [14] as our baseline. The LAS model is comprised of three parts: the LISTENER, which encodes the speech features into a high-level representation, the SPELLER, which generates a sequence of characters based on the encoded representation, and an attention module between Listener and Speller. Specifically, the Listener is a pyramidal bidirectional LSTM (pBLSTM), i.e. a multi-layer LSTM that halves its sequence length each layer, which outputs a short sequence of hidden states $\mathbf{h} = (h_1, \dots, h_U)$. The hidden state of the i^{th} time step for the j^{th} layer is given by $h_i^j = f(h_{i-1}^j, h_i^*)$, where f denotes the pBLSTM function and h_i^* equals the concatenation of h_{2i}^{j-1} and h_{2i+1}^{j-1}.

$$h_i^j = \text{PBLSTM}(h_{i-1}^j, [h_{2i}^{j-1}, h_{2i+1}^{j-1}]) \tag{1}$$

The decoder part consists of the attention module and the Speller and is defined as ATTENDANDSPELL function. The function is computed using an attention-based LSTM transducer. The distribution for output character y_i is a function of the decoder state s_i and context c_i. The decoder state s_i is a function of the previous state s_{i-1}, the previously emitted character y_{i-1} and context c_{i-1}. The context vector c_i is produced by an attention mechanism. Specifically,

$$c_i = \text{ATTENTIONCONTEXT}(s_i, \mathbf{h}) \tag{2}$$

$$s_i = \text{RNN}(s_{i-1}, y_{i-1}, c_{i-1}) \tag{3}$$

where RNN is a two-layer LSTM. At each time step i, the ATTENTIONCONTEXT generates a context vector c_i that encapsulates the information in the acoustic signal needed to generate the next character.

2.2. Hierarchical Attention-based Recurrent Decoder

The output character distribution at time step t of the Speller in the LAS model is conditioned on all previously characters by using a two-layer LSTM, where the historical information is stored in its hidden states. However, the input of LSTM cell is merely the previous character, which is not capable of carrying generated historical information in long dependencies The results from LAS on SEAME dataset show that most generated English tokens are not valid English words. For example, *four three two one* is mistaken as *forr tweee two one*, which are similar in pronunciations but vary in spellings. In order to alleviate this issue, we introduce Hierarchical Attention-based Recurrent Decoder (HARD) to the ATTENDANDSPELL function while remaining the LISTENER in LAS.

At each decoding time step i, as mentioned previously, attention context c_i is calculated using both encoder hidden states $\mathbf{h}$ and decoder states s_i. Instead of vanilla RNN, we employ a hierarchical recurrent network as HRED architecture [16] to further encode historical and sub-sequence code-switching information before attention calculation. Namely, we model each generated character as a signal of new sub-sequence. Thus, within an utterance, we will have sequences in character level and sub-sequence level, corresponding to word level and turn level in [16]. The HRED in our model has one *Character LSTM* LSTM_1 for character level, and a top level *Sub-sequence LSTM* LSTM_2 for sub-sequence level with hidden states s_i' and s_i respectively. Given an input character sequence $\mathbf{x} = (x_1, x_2, \cdots, x_T)$,

$$s_i' = \text{LSTM}_1(s_{i-1}', x_{i-1}) \tag{4}$$

$$s_i = \text{LSTM}_2(s_{i-1}, s_i', c_{i-1}) \tag{5}$$

The character LSTM_1 hidden states s_i' depend on both hidden states of previous time step s_{i-1}' and previous generated character x_{i-1} by time step i. The sub-sequence LSTM_2 hidden states s_i depend on three inputs: previous time step hidden states s_{i-1}, attention context c_{i-1} and output hidden states s_i' from LSTM_1 at current time step.

The sub-sequence LSTM_2 captures history within each sub-sequence and therefore has a better view of intra-language information, since some sub-sequences are not code-switched. This intra-language information is helpful to predict more reasonable token sequences for each language. While the character LSTM_1 are seeing code-switching transitions and memorizing information in hidden states, which is then fed to the sub-sequence.

After the attention module, the character distribution function CHARACTERDISTRIBUTION is employed to predict character sequence,

$$P(y_i|\mathbf{x}, y_{<i}) = \text{CHARACTERDISTRIBUTION}(s_i, c_i) \tag{6}$$

where CHARACTERDISTRIBUTION is an MLP followed by sampling outputs over the entire character set with Gumbel-softmax [17]. Namely,

$$\hat{y}_i = \text{SOFTMAX}(P(y_i|\mathbf{x}, y_{<i}) + z) \tag{7}$$

where $z \sim \text{GUMBEL}(0, 1)$, i.e. $p(z) = e^{-(z+e^{-z})}$. The Gumbel softmax samples outputs according to the current learned distribution P and is differentiable.

2.3. Multi-task Learning: Language Identification

We observe that LAS model generates more code-switched utterances and less pure English and Mandarin utterances compared to the ground truth. Therefore, we make an assumption that if the model knows when to better switch languages, which leads to the distribution of code-switched, pure English and pure Mandarin utterances closer to the ground truth. Based on this assumption, we design the multi-task learning approach, which jointly learn the character sequence and language id sequence, where each character has its own language id indicating the language it belongs.

We modify the output layer in decoder part, after the AttendAndSpell function, in addition to using an MLP with gumbel softmax to predict the character at each time step (CharacterDistribution in Eq. 6), we also employ the LanguageIdentifier function, which is another linear layer to predict language id for the character generated at the same time step simultaneously. Specifically, the language id l at time step i is given by

$$P(l_i|\mathbf{x}, l_{<i}) = \text{LanguageIdentifier}(s_i, c_i) \quad (8)$$

Moreover, the loss function becomes

$$\mathcal{L} = \lambda\mathcal{L}_{char} + (1-\lambda)\mathcal{L}_{LID} \quad (9)$$

where $\mathcal{L}_{char}$ is the loss of character sequence prediction task, $\mathcal{L}_{LID}$ is the loss of language identification task and λ is a hyper-parameter. Architecture of our Multi-Task Learning Hierarchical Attention-based Recurrent Network (MTL-HARD) is shown as Figure 1.

3. Experimental Setup

3.1. Dataset

We evaluate our model as well as the baseline model on SEAME dataset, a Mandarin-English code-switching speech corpus collected from South-East Asia. We randomly divide SEAME dataset into train, dev and test set according to a ratio of 8:1:1 in terms of number of utterance. Statistics are shown in Table 1. Performances are measured by CER and MER.

Table 1: *Statistics of SEAME dataset.*

	Train	Dev	Test
# Utterance	129,217	16,156	16,152
# Token	1,879,778	232,360	237,487
# EN Token	583,210	73,390	73,470
# CN Token	1,296,568	158,970	164,017

3.2. Parameters Setting

We construct our context-aware end-to-end code-switching ASR system with the same encoder from [14] and our decoder as described in Section 2.2 and Section 2.3. The hidden dimension of encoder and decoder LSTMs are 256 and 512 respectively, dimensions of attention key, query and value are 128. We train and test models in a batch size of 32, using ADAM as the optimizer with initial learning rate set to 0.001 and weight decay coefficient to 0.0001. In order to avoid explosion of lower layer gradients, we clip the norm of the gradients by scaling the gradients down by the ratio of 0.25 divided by max norm (if this ratio is less than 1) to reduce gradients norm. We also performs early stopping with patience set to 10 and dropout with probability of 0.5 to prevent models from overfitting. For quick and efficient training, we employ teacher forcing that use the ground truth from a prior time step as input and set the rate to 0.9. In all our experiments, HARD and LAS baseline share the same parameters as shown above.

4. Results and Analysis

4.1. Model Comparison

Comparison between performances of baseline model (LAS) and our models in CER and MER on test set is shown in Table 2. Apart from MER and CER on all utterances (All), we also report CER on code-switching utterances (CS), pure Mandarin utterances (CN) and pure English utterances (EN). Note that HARD+LID refers to our MTL-HARD model.

Table 2: *Model performance comparison. HARD can improve CER and MER over LAS, and MTL boosts the performance further. Therefore, MTL-HARD gives the best performance.*

Model	MER (%)	CER (%)			
		All	CS	CN	EN
LAS	38.99	29.91	28.31	34.34	30.29
HARD	35.42	27.23	25.65	32.39	24.90
LAS + LID	34.93	27.06	25.82	31.68	25.22
HARD + LID	**34.50**	**26.56**	**25.54**	**30.44**	**24.78**

The experimental results demonstrate a significant improvement in our hierarchical attention-based recurrent decoder model over the LAS model. Both MER and CER show consistent trends: while HARD outperforms baseline without multi-tasking, the language identification task brings additional boost in performance, and our MTL-HARD model achieves best in both MER and CER. In terms of fine-grained language specific CER, HARD with language identification performs best in code-switching, pure Mandarin and pure English sentences.

The hyperparameter λ in Equation 9 introduces a balance between speech recognition task and language identification task. Mandarin has a larger vocabulary size than English alphabets and dominates the SEAME dataset in terms of number of tokens. Therefore, a small λ does not count such lexicon imbalance between two languages, and the language identification task dominates when λ is large. Additional to the single task setting, we conduct a grid search from 0.1 to 0.9, and our model performs best at $\lambda = 0.5$, shown in Table 3.

Table 3: *Performances of MTL-HARD with different λ values. λ = 0.5 gives the best performance.*

λ	CER (%)			
	All	CS	CN	EN
0.1	27.60	26.34	32.23	25.91
0.2	27.81	26.48	32.04	27.87
0.3	27.29	26.22	30.93	26.70
0.4	26.80	25.55	31.21	25.58
0.5	**26.56**	**25.54**	**30.44**	**24.78**
0.6	27.23	25.88	32.13	25.53
0.7	27.90	26.69	32.27	26.50
0.8	28.84	27.84	32.58	27.35
0.9	30.18	28.73	35.15	29.24

Table 4: *Examples for different models, where HARD brings improvement by incorporating sub-sequence context. Multi-tasking in language identification leads to better recognition by converting subword code-switchings into full tokens in English and Mandarin. And with MTL-HARD, information learned in the history contributes to the recognition of following information, such as from 'kid' to 'children' in the first example, and from 'image processing' to its second mention and 'paper' in the second example.*

Example #1	
Ref:	like 你给我这些baby bonus 我都会like 只是一点点而已所以我like people still wouldn't have enough children ah (*like the baby bonus you give me, I like it just a little bit, therefore I like people still wouldn't have enough children ah.*)
LAS:	like 你因为因后就些bamibodnass 我都会like 只是一点点而里又有like builds 丢等样then 你就久灯这样
HARD:	like 你第我是这些beyb boaus h 都会like 只是一点而已所以我like peopleswtull yollde't have anough jhiodren ah
LAS $_{+\text{LID:}}$	like 你给我这些baby pouus 我都都会like 只是一点点而已也以ailike perple wtill 我ouldn't have enough 久hildren ah
HARD $_{+\text{LID:}}$	like 你把我这些baby 我oyes 我都like 只是一点点而已所以有like people ttull 我inldn t have enought children ah
Example #2	
Ref:	我有medical 因为那时候我有image processing 的base then 其他朋友没有image processing 的base 没有拿过paper (*I have medical because at that time I had a base of image processing then other friends didn't have bases of image processing and didn't have any paper.*)
LAS:	我回medical 因为有那时候我有image first aie singapbe d. then 其它朋里咯image proseici 应该就是你了老别pater
HARD:	我有medical 因为有时后我有image plocessingl的base 妹hen 其他朋友他有image brocesi因ng 了地ase 是有过ppaper
LAS $_{+\text{LID:}}$	我有medical 因为有时候有有image prrsessnng 的base then 去他朋友m有image proc.ssnng a品pasi 我有拿过paper
HARD $_{+\text{LID:}}$	我用medical 因为有时侯我有inage processing 的base hen 其它朋友没有image processing 的base 没有过paper

We also conduct study on how MTL-HARD incorporates sub-sequence history in recognition, and show some examples in Table 4. The first example has a consistent topic about *child*, since keywords *baby* and *children* are indicated in the reference sentence. The LAS model performs poorly and fails to recognize both keywords. Adding LID to LAS model improves the model and could recognize *baby* and still fails to generate children as LAS model is not good at capture long context as mentioned above. In contrast, our MTL-HARD model overcomes this issue as shown in the last sentence: the context of *baby* are carried along the sequence helps the recognition of *children*. Similarly, in the second example, it helps in the second mentions of *image processing* and *paper*. Additionally, multi-tasking helps convert subword code-switching cases into full English/Mandarin tokens.

4.2. BPE vs Character

Applying byte-pair encoding (BPE) on English shows improvement of word error rate (WER) over purely character level models [18, 19, 20]. However, our experiments reveal that BPE does not improve code-switching speech recognition performance in terms of CER and MER. In detail, our BPE model applies pre-trained subword tokenizer mentioned in [21] to English, while keeping it in character level for Mandarin. We compare BPE model in subword corpus size of 1000, 3000 and 10000, named as BPE-1k, BPE-3k and BPE-10k respectively. Using our MTL-HARD model and set $\lambda = 0.5$, the results are shown in Table 5.

Table 5: *BPE versus character based model in CER and MER.*

Unit	**MER (%)**	**CER (%)**			
		All	CS	CN	EN
BPE-10k	36.47	40.23	41.23	35.45	44.87
BPE-3k	38.92	41.30	42.25	37.58	43.27
BPE-1k	39.18	38.86	39.54	36.10	40.53
Chars	**34.50**	**26.56**	**25.54**	**30.44**	**24.78**

We can infer from the results that small subword corpus tends to yield better CER numbers - character model can be regarded as a subword model whose corpus contains only 26 units for English. Thus, these three models share the same Mandarin corpus while vary in English subword corpus, which is consistent with the fact that they achieve relatively approaching CER on Mandarin, while more significantly differ in English and code-switching utterances.

Regarding MER, though character model still achieved best MER, BPE-1k performs worse than BPE-3k and BPE-10k and the differences among three models are smaller compared with CER numbers. One possible explanation for BPE-10k beating BPE-1k is that subwords in larger corpus are more closer to full words - a smaller vocabulary size will yield a segmentation into many subwords, while a large vocabulary size may leave frequent words unsplit. For example, the word "hollywood" is splitted as '*h*', '*ol*', '*ly*', '*w*', '*ood*'; '*hol*', '*ly*', '*wood*' and '*hollywood*' with BPE-1k, BPE-3k, and BPE-10k tokenizations respectively. Another reason which can lead to BPE being outperformed by character based model in our experiments is the distribution of Mandarin and English tokens. From Table 1 we can observe the domination of Mandarin tokens over English tokens, thus when calculating MER, error rate on Mandarin tokens weighs more than it on English tokens.

5. Conclusion and Future Work

In this work, we propose Hierarchical Attention-based Recurrent Decoder, an attention-based model which employs a hierarchical recurrent network to increase model's awareness of previous generated context sequence at decoding. Our model improves CER and MER significantly on SEAME dataset over LAS model. We also find that jointly learning language id and speech recognition boosts model performance. Although reported to be able to improve WER in recent works, BPE does not outperform character based model in terms of CER and MER.

Moreover, we plan to conduct experiments on pre-training the model on monolingual English and Mandarin speech dataset to boost the performance of English tokens recognition. And we would like to incorporate an advanced code-switching language model for re-scoring and vocabulary expansion to resolve the issue that ASR model often fails to generate reasonable English tokens.

6. References

[1] A. K. Joshi, "Processing of sentences with intra-sentential code-switching," in *Coling 1982: Proceedings of the Ninth International Conference on Computational Linguistics*, 1982.

[2] D. Sankoff, "The production of code-mixed discourse," in *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*. Montreal, Quebec, Canada: Association for Computational Linguistics, Aug. 1998.

[3] P. Gardner-Chloros and M. Edwards, "Assumptions behind grammatical approaches to code-switching: when the blueprint is a red herring," *Transactions of the Philological Society*, vol. 102, no. 1, pp. 103–129, 2004.

[4] D. Lyu, R. Lyu, Y. Chiang, and C. Hsu, "Speech recognition on code-switching among the chinese dialects," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2006, Toulouse, France, May 14-19, 2006*, 2006, pp. 1105–1108.

[5] B. H. A. Ahmed and T. Tan, "Automatic speech recognition of code switching speech using 1-best rescoring," in *2012 International Conference on Asian Language Processing, Hanoi, Vietnam, November 13-15, 2012*, 2012, pp. 137–140.

[6] E. Yilmaz, H. van den Heuvel, and D. A. van Leeuwen, "Acoustic and textual data augmentation for improved ASR of code-switching speech," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana, Ed. ISCA, 2018, pp. 1933–1937.

[7] P. Guo, H. Xu, L. Xie, and C. E. Siong, "Study of semi-supervised approaches to improving english-mandarin code-switching speech recognition," in *INTERSPEECH*, 2018.

[8] Z. Zeng, Y. Khassanov, V. T. Pham, H. Xu, C. E. Siong, and H. Li, "On the end-to-end solution to mandarin-english code-switching speech recognition," in *INTERSPEECH*, 2019.

[9] K. Li, J. Li, G. Ye, R. Zhao, and Y. Gong, "Towards code-switching asr for end-to-end ctc models," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6076–6080.

[10] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[11] C. Weng, J. Cui, G. Wang, J. Wang, C. Yu, D. Su, and D. Yu, "Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana, Ed. ISCA, 2018, pp. 761–765.

[12] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015.

[13] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, "Investigating end-to-end speech recognition for mandarin-english code-switching," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6056–6060.

[14] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[15] D. Lyu, T. P. Tan, E. Chng, and H. Li, "SEAME: a mandarin-english code-switching speech corpus in south-east asia," in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. ISCA, 2010, pp. 1986–1989.

[16] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J.-Y. Nie, "A hierarchical recurrent encoder-decoder for generative context-aware query suggestion," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 553–562.

[17] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[18] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana, Ed. ISCA, 2018, pp. 7–11.

[19] T. Zenkel, R. Sanabria, F. Metze, and A. Waibel, "Subword and crossword units for CTC acoustic models," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana, Ed. ISCA, 2018, pp. 396–400.

[20] Z. Xiao, Z. Ou, W. Chu, and H. Lin, "Hybrid ctc-attention based end-to-end speech recognition using subword units," in *11th International Symposium on Chinese Spoken Language Processing, ISCSLP 2018, Taipei City, Taiwan, November 26-29, 2018*. IEEE, 2018, pp. 146–150.

[21] B. Heinzerling and M. Strube, "BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. Miyazaki, Japan: European Language Resources Association (ELRA), May 7-12, 2018 2018.