Adapting the adapters for code-switching in multilingual ASR

Atharva Kulkarni, Ajinkya Kulkarni, Miguel Couceiro, Hanan Aldarmaki

ADAPTING THE ADAPTERS FOR CODE-SWITCHING IN MULTILINGUAL ASR

Atharva Kulkarni[*,‡], *Ajinkya Kulkarni*[*], *Miguel Couceiro*[†], *Hanan Aldarmaki*[*]

[*] MBZUAI, UAE, [†] Université de Lorraine, CNRS, LORIA, France, [‡]Erisha Labs, India

ABSTRACT

Recently, large pre-trained multilingual speech models have shown potential in scaling Automatic Speech Recognition (ASR) to many low-resource languages. Some of these models employ language adapters in their formulation, which helps to improve monolingual performance and avoids some of the drawbacks of multi-lingual modeling on resource-rich languages. However, this formulation restricts the usability of these models on code-switched speech, where two languages are mixed together in the same utterance. In this work, we propose ways to effectively fine-tune such models on code-switched speech, by assimilating information from both language adapters at each language adaptation point in the network. We also model code-switching as a sequence of latent binary sequences that can be used to guide the flow of information from each language adapter at the frame level. The proposed approaches are evaluated on three code-switched datasets encompassing Arabic, Mandarin, and Hindi languages paired with English, showing consistent improvements in code-switching performance with at least 10% absolute reduction in CER across all test sets.

Index Terms— Speech, ASR, code-switching, multilingual, adapters

1. INTRODUCTION

Large pre-trained multi-lingual speech models have been proliferating in recent years, including various self-supervised models, as well as supervised ASR models, such as the recently released Massively Multilingual Speech (MMS) project[1], which expanded the multilingualism in speech processing applications to 1,406 languages.To account for the variability in these languages, language adapters were employed to learn language-specific features at some parts of the networks, while sharing most of the overall parameters to enable cross-lingual transfer. While this formulation leads to better management of the vast multi-lingual use cases, it restricts the use of the model on code-switched speech, where a single speech utterance contains two or more languages. Code-switching (CS) is generally challenging for ASR models, which usually result in degraded ASR performances [1]. This decline in performance in ASR tasks while handling CS may lead to error propagation across various modules (both text and speech processing) in conversation AI systems. CS can be categorized into two types: inter-sentential CS, where language switches occurs at sentence boundaries, and intra-sentential CS, where the switching takes place within the same sentence [2]. The *matrix* language is the main language used as a basis in most utterances, and the *embedded* language is the secondary language in code-switching. For example, many Arabic or Hindi speakers often use English as an embedded language in casual conversations.

Previous works that attempt to address CS in bilingual or multilingual ASR [3, 4, 5] typically work by explicitly modeling language information either in acoustic ASR models [6, 7, 8, 9] or in language modeling [10, 11, 12]. In contrast to previous works that train models for specific language pairs from scratch, we focus on improving CS recognition in large pre-trained multilingual models that rely on language adapters, using MMS as our model of choice. MMS is based on the Wav2Vec 2.0 architecture and has been trained both using self-supervised learning as well as supervised pre-training for ASR using language adapters that can be loaded and exchanged at inference time as needed. The model consists of multiple Transformer blocks, each appended with a language-specific adapter, in addition to language-specific softmax layers for prediction.

As intra-sentential CS speech contains more than one language within utterances, the pre-specification of a language adapter results in degradation of ASR performance since the model is not optimized to identify and predict the embedded language. In this work, we introduce two approaches to regulate the information flow through each transformer block using two language adapters. First, we propose a *Post-Adapter Switcher* (PAS) network, which integrates the information from two adapters and learns to modulate the hidden output information. Second, we propose a *Transformer Code Switcher* (TCS), where we explicitly estimate the binarized sequence of CS points using a transformer block, which enable adapter switching at the frame level. The latent binary code sequence is assimilated in the hidden outputs from both target language adapters in each block of transformer in ASR, thus enabling the MMS system to gain finer control over boundaries of CS in a given speech utterance. The PAS and TCS network are first fine-tuned using

[1] https://ai.meta.com/blog/multilingual-model-speech-recognition/

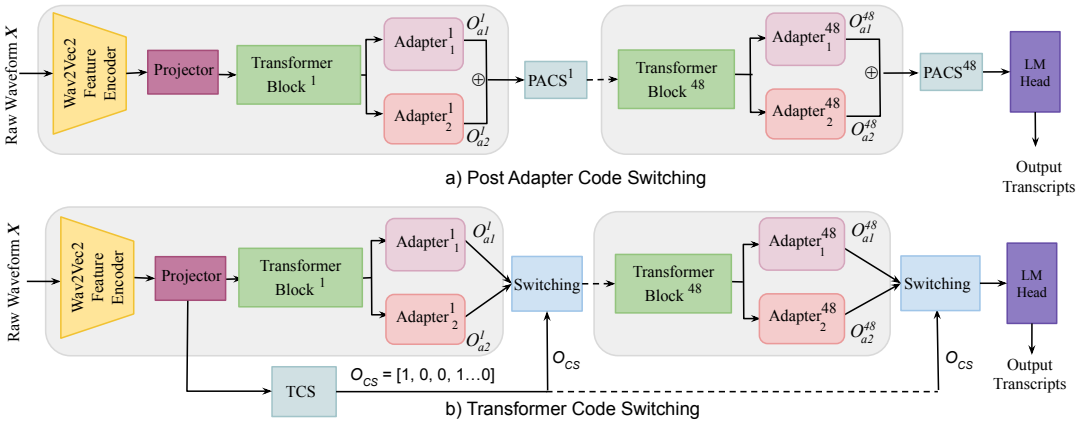

Fig. 1: Framework of proposed approaches with MMS, **a) Post Adapter Switching Approach b) Transformer Code Switching.** Transformer blocks range from 1 to 48 and the grey color indicates frozen model parameters during training.

relatively small amounts of CS training data in the target language pair, while keeping all other parameters frozen. To illustrate the robustness of the approach, we evaluate it on three different CS datasets, for which English is the embedded language: ASCEND (Mandarin-English) [13], ESCWA (Arabic-English) [**?**], and MUCS (Hindi-English) [14]. We released the pre-trained models along with inference code on a Github repository[2].

The **contributions** of this paper are as follows, **1.** Presenting a novel adapter switching network for handling CS in MMS and similar systems. **2.** Proposing a novel mechanism for regulating information flow from language-specific adapters at the frame level. **3.** Demonstrating improved performance on CS datasets compared to the original MMS model.

2. PROPOSED APPROACHES

In this section, we present the proposed framework for adapting the adapters in the context of code-switched ASR using the MMS system. First, we briefly describe the general framework of MMS ASR system and the role of adapters in scaling it to multiple languages. Thereafter, we provide the details of the proposed approaches to utilize two language adapters for recognizing CS speech: *Post Adapter Code Switching* (PACS) and *Transformer Code Switching* (TCS).

2.1. The MMS model

The MMS framework is built upon the Wav2Vec2 architecture, which serves as the feature encoder network. This encoder learns to convert raw speech waveforms into latent representations using 1D temporal convolutions. In the ASR task, these latent representations are input into transformer blocks to generate contextual representations. The final contextual representations are processed by a language-specific feed-forward/softmax layer, called LM head, to predict the distribution of tokens within the language vocabulary. The transformer block follows the BERT architecture [15] with 48 blocks. MMS training involves a two-stage process. First, it undergoes self-supervised learning by solving a contrastive task using masked feature encoder outputs from training data representing 1,406 languages. After this pre-training phase, the base model weights, including the feature encoder, feature projector, and the 48 transformer blocks, are frozen. Only the adapter modules and the LM head are further trained on labeled data using the CTC loss for ASR. During inference, MMS only requires specification of the language as an additional argument along with the speech utterance to load the corresponding language adapters and LM head. For more comprehensive details on MMS and the ASR training setup, please refer to [16].

To support ASR for over 1,100 languages, language-specific adapter modules [17] are incorporated at the end of each transformer block. These adapter modules consist of two feedforward layers with a LayerNorm layer and a linear projection to 16 dimensions with ReLU activation. Additionally, language-specific LM heads are selectively trained because token characters can vary among languages. This selective approach is especially important in cases like CS ASR, where the system needs to produce token characters from multiple languages, making a single language-specific adapter inadequate.

2.2. Post Adapter Code Switching (PACS)

This section outlines the Post Adapter Code Switching (PACS) approach, which involves incorporating information from two pre-trained language adapters, denoted as A_1^n and A_2^n corresponding to the matrix and embedded languages, where subscript n denotes the n^{th} transformer block. As depicted in Fig 1a, the hidden output of each transformer block is fed into both A_1^n and A_2^n. The resulting outputs, O_{a1}^n

[2] https://github.com/Atharva7K/MMS-Code-Switching

and O^n_{a2} from A^n_1 and A^n_2, are both 1280-dimensional. These outputs are concatenated and provided as input to the n^{th} PACS block, which is designed with the same architecture as the adapters and has an input dimension of 2560, allowing it to accommodate the concatenated outputs from O^n_{a1} and O^n_{a2}. During the training phase, MMS is trained alongside PACS on a labeled code-switching (CS) speech dataset. This training inherently enables PACS to learn how to handle CS without the need for additional auxiliary loss. The weights of the Wav2Vec2 encoder and transformer blocks remain fixed. The output dimension of the PACS module is set to 1280 to fit the input dimension for the subsequent transformer block. This design facilitates the seamless integration of PACS into the MMS framework without modifying the underlying MMS components.

Furthermore, to consider character tokens from both languages in CS, we concatenate the pretrained weights of the LM head layers from both languages. For example, if the English and Arabic LM Head layers have output dimensions of 154 and 121 respectively, we created a new feedforward layer with a dimension of 275 (154+121). We then copy the first 154 weights from the English LM head and the remaining 121 from the Arabic LM head. This approach allows us to utilize pre-trained LM head without the need to train them from scratch. Additionally, it's worth noting that each LM head contains lower-case English alphabets along with punctuation and digits. This presents a challenge when merging the pre-trained LM head weights of the English and matrix languages. To address this, we construct a binary mask that assigns a probability of 0.0 to punctuation and English alphabet character tokens for the matrix language LM head. The concatenated LM head is fine-tuned along with the code-switching network.

2.3. Transformer Code Switching (TCS)

In this section, we describe Transformer Code Switcher (TCS), which is designed to capture the latent binary sequence representing CS points in the utterance. The TCS network takes the output of the projector block as input to estimate switching points from input features, as depicted in Fig 1b. The estimated binary code sequence, denoted as $O_{CS} = [1, 1, ..., 0, 1, ...0]$, is combined with the outputs of the two adapters, namely O_{a1} and O_{a2}, at each transformer block, according to Equation 1.

$$O^n_{Switch} = (1 - O_{CS}) * O^n_{a1} + O_{CS} * O^n_{a2} \quad (1)$$

For implementing TCS, we chose a transformer architecture with an output sigmoid activation function. This choice allows us to leverage the self-attention mechanism, which assists in identifying CS boundaries at the frame level. Therefore O_{CS} regulates the flow of information from two adapters to enable the network to handle CS speech by dynamically masking out one of the languages based on the input features. We applied a threshold value of 0.5 to the output of the sigmoid activation to create binarized latent codes for CS. Subsequently, the outputs of both adapters are processed through Equation 1 as O^n_{Switch} and passed to the next transformer block. Thus, the switching operation enables finer control over CS. Since the output dimension of adapters is of 1280 dimension, we repeated O_{CS} across the output dimension to ensure the mask is applied across all features in the given frame.

3. DATA PREPARATION

We use 3 code-switched datasets to evaluate our models. **ASCEND** [13]: is a Mandarin-English code-switched corpus with spontaneous conversational speech spanning over 10.62 hours comprising of $\tilde{1}$2.3K utterances in total. The train-validation-test splits contain 9.8K, 1.1K, and 1.3K utterances corresponding to 8.78, 0.92, and 0.92 hours of speech, respectively. **MUCS** [14]: is a family of multilingual and code-switched ASR datasets for Indic languages introduced in the MUCS challenge. We use only the Hindi-English code-switched data from MUCS with a train-val split of 52.8K, 3.1K utterances and duration 89.86, and 5.18 hours of speech, respectively. For evaluation, we use the validation set as the test set as no explicit evaluation set was provided. **QASR** [18]: is a large-scale multi-dialectal speech corpus for Arabic spanning over 2000 hours with some utterances containing Arabic-English CS. We selectively use only the code-switched samples from the entire corpus for training. Specifically, we select transcripts that contain complete English words and not only numbers or dates. We were able to extract 5.8K utterances over 4.8 hours for the training. **ESCWA**: [**?**] It consists of 845 utterances spanning 2.8 hours with Arabic-English code-switching. We used ESCWA as the test set to evaluate models trained on the QASR dataset.

4. EXPERIMENTAL SETUP

All the experiments are conducted by modifying the HuggingFace implementation[3]. We used Adam optimizer with a learning rate of 1e-6 in the case of QASR, ASCEND, and 1e-5 for MUCS to optimize our models using CTC loss. We experimented with learning rates of 1e-3, 1e-5, and 1e-6 that led to best performances. We used a warm-up of 1K steps and decaying thereafter according to a polynomial lr scheduling. Following the experimentation with batch sizes of 8, 16, and 32, we found that a batch size of 32 is optimal for QASR and ASCEND, while a batch size of 16 is most effective for MUCS. All the experiments were carried out on a single Nvidia A100 40GB GPU for up to 100 epochs, by selecting the best-performing model on the validation set where applicable. For the CTC loss function, we set

[3] https://github.com/huggingface/transformers/tree/main/src/transformers/models/wav2vec2

Model	ASCEND		ESCWA		MUCS		# params (Millions)
	MER	CER	WER	CER	WER	CER	
MMS with single language adapter:							
English	98.02	87.85	92.73	71.14	101.72	74.02	965
Matrix-language	71.98	66.76	75.98	46.38	58.05	49.20	965
Proposed models for fine-tuning:							
Matrix-language-FT	45.97	44.13	77.47	37.69	66.19	41.10	965
PACS	44.41	40.24	75.50	46.69	63.32	42.66	970
TCS	41.07	37.89	74.42	**35.54**	**57.95**	**38.26**	980
Whisper-large, with language id prompt:							
Matrix language id	23.31	23.49	59.30	39.66	104.4	87.48	1550
Concatenated prompt [19]	**21.13**	**21.30**	**58.70**	38.86	101.62	91.59	1550

Table 1: Evaluation of proposed approaches on three CS datasets (ASCEND, ECSWA, and MUCS). Except for Whisper and Whisper-prompt, all other systems indicate the adapter approach with MMS; FT denotes fine-tuned.

the `ctc_zero_infinity` parameter to True for managing optimization issues.

5. RESULTS AND DISCUSSION

We report performance in terms of ASR character error rate (CER) and mixed error rate (MER) to evaluate performance on Mandarin-English code-switched ASCEND, which accounts for the fact that individual Chinese characters represent words. To compute CER and MER for ASCEND, we used the implementation from the official repository [4].In case of ESCWA and MUCS datasets, we use Word Error Rate (WER) and CER for evaluation using HuggingFace evaluate library[5].

Table 1 shows the performance of all of our approaches along with the baselines on the three datasets. We can see that plain MMS performs poorly on code-switched datasets. In the case of plain MMS, the performance is better when we use language adapters corresponding to the matrix language, *i.e*, Mandarin in case of ASCEND, Hindi in the case of MUCS, and Arabic in the case of ESCWA/QASR as expected. We also directly fine-tune the matrix language adapter as another baseline, termed Matrix-language-FT in the Table 1. Fine-tuning matrix-language adapters significantly improves the CER in case of ESCWA and MUCS, but it worsens the WER. Except ASCEND, fine-tuning actually degrades WER; this indicates that directly fine-tuning one language adapter on CS speech is not a feasible option, as the pre-trained adapters are largely biased towards the specified languages. Both proposed frameworks, PACS and TCS improve performance compared with the original model and direct fine-tuning of the matrix language adapter. Among MMS-based variants, TCS results in the best overall performance in all metrics and test sets.

We also show the performance of the pre-trained Whisper-large model as a point of reference since the two models are not directly comparable. We do not fine-tune the Whisper model, but we use the prompt approach described in [19], which concatenates the language ids in the prompt to improve performance on CS speech. One interesting observation from these results is that Whisper model, without fine-tuning, results in much better performance than MMS raw performance (over 40% absolute CER difference) in the ASCEND dataset. While our best adaptation method improves performance by $\sim$30% absolute CER, it remains significantly higher the Whisper's raw performance. This is likely a direct result of the size of the labeled dataset used in the supervised pre-training. While exact numbers of training hours are not directly reported for MMS, we can estimate it based on the component datasets. In our estimate, MMS ASR model was pre-trained on $\sim$1K hours of Mandarin speech, while Whisper was pre-trained using over 23K hours.

6. CONCLUSION

We presented a framework for adapting adapter-based multilingual ASR models for code-switched speech recognition. We conducted experiments using the MMS model, which was pre-trained on a wide range of languages, including English, Mandarin, Arabic, and Hindi, matching the languages in our CS test sets. Since the MMS framework involves loading language adapters, the original model performs poorly in our test sets. The proposed models involve loading the language adapters for the matrix and embedded language, and mixing their predictions to produce better code-switched transcriptions. With a small increase in model size, our proposed method improves the CS output across all test sets, with more than 10% absolute reduction in CER. Note however that we did not employ language model rescoring in the current empirical study. Thus our models could be potentially improved by using an external LM trained with synthetic CS text or by controlling the adapter choice by reinforcement learning, which we leave for future work.

[4] `https://github.com/HLTCHKUST/ASCEND/blob/main/eval.py`

[5] `https://huggingface.co/docs/evaluate`

7. REFERENCES

[1] M. B. Mustafa, M. A. Yusoof, H. K. Khalaf, Ahmad A. R. M. Abushariah, Miss L. M. Kiah, H. N. Ting, and S. Muthaiyah, "Code-switching in automatic speech recognition: The issues and future directions," *Applied Sciences*, 2022.

[2] G Winata, A F Aji, Z X Yong, and T Solorio, "The decades progress on code-switching research in NLP: A systematic survey on trends and challenges," in *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada, July 2023, pp. 2936–2978, Association for Computational Linguistics.

[3] E Yilmaz, H van den Heuvel, and D. A. van Leeuwen, "Code-switching detection using multilingual dnns," *2016 IEEE Spoken Language Technology Workshop (SLT)*, pp. 610–616, 2016.

[4] X Yue, G Lee, E Yilmaz, F Deng, and H Li, "End-to-end code-switching asr for low-resourced language pairs," *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 972–979, 2019.

[5] S Nakayama, A Tjandra, S Sakti, and S Nakamura, "Speech chain for semi-supervised learning of japanese-english code-switching asr and tts," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 182–189, 2018.

[6] H. S. Chadha, P Shah, A Dhuriya, N Chhimwal, A Gupta, and V Raghavan, "Code switched and code mixed speech recognition for indic languages," *ArXiv*, vol. abs/2203.16578, 2022.

[7] L-H Tseng, Y-K Fu, H Chang, and H yi Lee, "Mandarin-english code-switching speech recognition with self-supervised speech representation models," *AAAI Workshop on Self-supervised Learning for Audio and Speech Processing (SAS) 2022*, 2021.

[8] A Biswas, E Yilmaz, E van der Westhuizen, F de Wet, and T. R. Niesler, "Code-switched automatic speech recognition in five south african languages," *Comput. Speech Lang.*, vol. 71, pp. 101262, 2018.

[9] H Seki, S Watanabe, T Hori, J. L. Roux, and J. R. Hershey, "An end-to-end language-tracking speech recognizer for mixed-language speech," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4919–4923, 2018.

[10] E Yilmaz, A Biswas, E van der Westhuizen, F de Wet, and T. R. Niesler, "Building a unified code-switching asr system for south african languages," *ArXiv*, vol. abs/1807.10949, 2018.

[11] H Adel, N. T. Vu, F Kraus, T Schlippe, H Li, and T Schultz, "Recurrent neural network language modeling for code switching conversational speech," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8411–8415, 2013.

[12] A. M. Ali, Shammur A. Chowdhury, A. Hussein, and Y Hifny, "Arabic code-switching speech recognition using monolingual data," *Interspeech*, 2021.

[13] H Lovenia, S Cahyawijaya, G. I. Winata, P Xu, X Yan, Z Liu, R Frieske, T Yu, W Dai, E. J. Barezi, and P Fung, "Ascend: A spontaneous chinese-english dataset for code-switching in multi-turn conversation," in *International Conference on Language Resources and Evaluation*, 2021.

[14] A Diwan, R Vaideeswaran, S Shah, A Singh, S Raghavan, S Khare, V Unni, S Vyas, A Rajpuria, C Yarra, A Mittal, P. K. Ghosh, P Jyothi, K Bali, V Seshadri, S Sitaram, S Bharadwaj, J Nanavati, R Nanavati, and K Sankaranarayanan, "MUCS 2021: Multilingual and Code-Switching ASR Challenges for Low Resource Indian Languages," in *Proc. Interspeech 2021*, 2021, pp. 2446–2450.

[15] Jacob D., Ming-Wei C., Kenton L., and Kristina T., "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, June 2019, pp. 4171–4186, Association for Computational Linguistics.

[16] V Pratap, A Tjandra, B Shi, P Tomasello, A Babu, S Kundu, A Mamdouh E, Z Ni, A Vyas, M. Fazel-Zarandi, A Baevski, Y Adi, X Zhang, Wei-Ning Hsu, A Conneau, and M Auli, "Scaling speech technology to 1, 000+ languages," *ArXiv*, vol. abs/2305.13516, 2023.

[17] N Houlsby, A Giurgiu, S Jastrzebski, B Morrone, Quentin De L, A Gesmundo, M Attariyan, and S Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.

[18] H Mubarak, A. Hussein, S. A. Chowdhury, and A. M. Ali, "Qasr: Qcri aljazeera speech resource a large scale annotated arabic speech corpus," in *Annual Meeting of the Association for Computational Linguistics*, 2021.

[19] Puyuan P., Brian Y., W. Shinji, and David H., "Prompting the hidden talent of web-scale speech models for zero-shot task generalization," *arXiv preprint arXiv:2305.11095*, 2023.