INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

The Tag-Team Approach: Leveraging CLS and Language Tagging for Enhancing Multilingual ASR

Kaousheik Jayakumar, Vrunda N. Sukhadia, A Arunkumar, S. Umesh

Speech Lab, Indian Institute of Technology Madras, India
kaousheik@gmail.com, ee20s008@smail.iitm.ac.in, arunkumaras10@gmail.com, umesh@ee.iitm.ac.in

Abstract

Building a multilingual Automated Speech Recognition (ASR) system in a linguistically diverse country like India can be a challenging task due to the differences in scripts and the limited availability of speech data. This problem can be solved by exploiting the fact that many of these languages are phonetically similar. These languages can be converted into a Common Label Set (CLS) by mapping similar sounds to common labels. In this paper, new approaches are explored and compared to improve the performance of CLS based multilingual ASR model. Specific language information is infused in the ASR model by giving Language ID or using CLS to Native script converter on top of the CLS Multilingual model. These methods give a significant improvement in Word Error Rate (WER) compared to the CLS baseline. These methods are further tried on out-of-distribution data to check their robustness.

Index Terms: Automatic Speech Recognition, Multilingual, Common Label Set, Language ID

1. Introduction

A major problem in building good ASR systems for many languages is the lack of transcribed data. To tackle the data insufficiency problem, most works have focused on pooling data from different languages to build multilingual systems. Multilingual ASR systems trained with data from many languages not only have an inherent advantage of recognizing speech corresponding to different languages used in training but also solve the data insufficiency problem by pooling. [1] explored a common acoustic model for multiple languages based on a global phone set. Later, [2] explored different training approaches using tandem and bottleneck features for multilingual Multi-layer perceptron models. In [3], a hybrid attention-CTC model was employed to perform grapheme-based speech recognition of 10 different languages. A similar idea of using a single multilingual end-to-end model based on attention was shown to improve performance over monolingual models for various Indian languages with the grapheme target being a union of all the characters in the languages considered [4]. The application of adapter modules was explored to handle imbalanced datasets in a multilingual scenario in [5].

In multilingual systems, as the number of languages increases, the number of target units to be modeled also increases. The concept of using a common phone set, which reduces the number of target units, was earlier used in English-Hindi code switched ASR [6, 7] and also in multilingual speech synthesizers across four Indian languages [8]. Previous work in multilingual ASR [9] proposed a Common Labeled Set (CLS) to map characters with similar sounds from different Indian languages to one common representation using their similar characteristics of Unicode representation across Indian languages. Though the CLS approach gives an improvement in performance, in order to render the decodes in their native script we require prior knowledge of the language.

The idea of adding a Language Identity (LID) token also has been investigated in many works in different contexts. Various methods have been proposed based on how the LID is incorporated in the models. [10] proposed the use of Language ID tokens and language embedding for the low-resource multilingual ASR model to incorporate the language information in the model. [11, 12] explore having a LID predictor before feeding the identified chunks to the respective monolingual models to handle code-switch utterances. [13] talks about learning and predicting the LID parallelly through which the best decode is selected from different monolingual systems using the respective confidence scores. Further, integrating LID and multilingual ASR under a common framework is discussed in [3].

In this work, we have focused on exploring methods to improve the performance of CLS models by incorporating language information. CLS models often struggle to accurately capture the linguistic aspects of different languages, which can lead to lower accuracy. To address this challenge, we have investigated two different approaches: (i) CLS Model with CLS-to-Native Script converter [CLS2NS] and (ii) CLS model with LID token. Using CLS2NS on top of CLS-ASR model helps take care of the linguistic aspects of that language. Using CLS model with LID tokens is another way to provide language information to the model. To evaluate the efficacy of these approaches, we compared them with a conventional CLS model. Our experiments demonstrate that both the CLS model with LID and CLS model with CLS2NS approaches outperform the conventional CLS ASR model in terms of WER (word error rate).

The rest of the paper is organized as follows. Section 2 briefly describes the dataset and architecture details used in this work. Section 3 discusses the Baseline Experiments. Section 4 describes the Common Label Set. Section 5 details approaches we followed to use LID to build a multilingual ASR system. Section 6 discusses the results of various experiments performed in this work. Section 7 draws important conclusions based on the results observed.

2. Dataset and Architecture Details

2.1. Dataset

This paper utilizes publicly available labeled speech data obtained from the Universal Language Contribution API (ULCA)[1]

[1] https://github.com/Open-Speech-EkStep/ULCA-asr-dataset-corpus

10.21437/Interspeech.2023-2406

for five Indian languages: Hindi, Gujarati, Marathi, Bengali, and Odia, which are all Indo-Aryan languages. A random subset of 200 Hours is taken as the training data for all languages. All audio files in datasets are sampled at 16kHz. The duration of training, validation, and test speech data from each language is given in Table 1

	Train	Valid	Test
Hindi	206	15	14
Marathi	201	15	15
Gujarati	213	17	17
Bengali	207	13	13
Odia	211	16	16
Total	1038	76	75

Table 1: *No. of hours of data in each language*

2.2. Architecture Details

2.2.1. ASR system

In this paper, all ASR models are based on the Transformer framework [14] with joint Connectionist Temporal Classification (CTC)/attention multi-task learning with a CTC weight of 0.3 using ESPNet[15]. The standard configurations for all the systems are given in Table 2. All ASR models in this paper are trained on one A100 GPU for 50 epochs.

Hyperparameters	Values
Feature vector dimension	512
Number of encoder layers	12
Encoder units	2048
Number of decoder layers	6
Decoder units	2048
Attention heads	8
CTC weight	0.3

Table 2: *Transformer model configurations*

3. Baseline

3.1. Monolingual Model

In this setup, each language is individually modeled using a transformer model that is specifically trained on 200 hours of data from the respective language. 750 byte pair units are utilized for each individual model training. Notably, no language model is used to test these models. All other architecture details are consistent with those presented in Section 2.2.1. The results obtained from these Monolingual Models are presented in Table 5.

3.2. Multilingual ASR Model

To build the Multilingual ASR model, data from all the languages are combined and used to train a single transformer model, which follows the same architecture details as presented in Section 2.2.1. Note that in this case the text is in native script of each language. The total training data amounts to 1000 hours, with 200 hours collected from each of the five languages. As language information is not explicitly provided to the model during training, the Multilingual ASR model is designed to be language-independent. The model utilizes 2500 Byte Pair Encodings (BPE) during training and does not rely on any External Language Model during testing.

4. Common Label Set Multilingual ASR Models

4.1. Common Label Set

Since graphemes and phonemes in Indian languages have a strong correlation, grapheme-to-phoneme conversion can be simple and rule-based. The possibility of a reduced and universal set of target labels is afforded by the acoustic similarity between corresponding graphemes in various languages. A standard set of labels are given to phonetically similar speech sounds in Indian languages by the CLS. This work uses the CLS proposed in [16]. Equivalent sounds corresponding to different grapheme from different languages are given the same label. Each CLS label is made up of a string of alphabet and Roman numbers. For instance, the CLS label "aa" stands in for the sound represented by the International Phonetic Alphabet (IPA) /a:/. Table 3 provides examples of CLS mapping with a few Indian language scripts. Unified Parser [17] is used to create CLS scripts from the corresponding Native Script. The language is determined by the Unicode range, and the language-specific rules are then applied to generate the CLS-based representation. Table 4 demonstrates that the similar-sounding phones are mapped to the same CLS labels even though the scripts are different. It is important to note that reconstructing native script text from CLS text is not straightforward due to special cases such as schwa deletion, geminate correction, and syllable parsing.

Label	IPA	Hindi	Marathi
a	/a/	अ	अ
aa	/a:/	आ	आ
au	/aΩ/	-	औ
nx	/ɳ/	ण	ण
m	/m/	म	म

Table 3: *Examples of CLS Mapping with Native Script Characters*

Using the CLS approach in Section 4.1, the native script text is converted to a CLS text format for all training data from different languages. We then trained an end-to-end ASR model to predict CLS text based on speech input. However, since the eventual output needs to be in the native script, a CLS-to-Native converter model is built to convert the CLS text back to the native script text which is discussed in the subsequent sections.

4.2. Multilingual CLS Model

To create the multilingual CLS model, a transformer model is utilized with 12 Encoder layers, 6 Decoder layers, and 3000 BPE units. This approach allows the model to effectively learn and identify the common sound patterns shared across multiple languages, providing a more efficient and effective means of speech recognition. By leveraging the benefits of the CLS, this model is capable of producing highly accurate and contextually relevant speech output across multiple Indian languages in the CLS target space.

4.3. CLS-to-Native Script Converter

The CLS-to-Native Script converter [CLS2NS] is a text-to-text transformer model as shown in Figure 2. In this paper, a CLS2NS is built for each language. The input to CLS2NS is CLS text and the target is the native script. We need excellent transliteration models to exploit the gains realized by the CLS

system. Otherwise, the CLS2NS model might introduce errors that will affect the system's overall performance. A transformer model with 6 encoder layers, 6 decoder layers, and 4 attention heads are used to train this model for each language. The CLS2NS models are trained using the *Fairseq* toolkit [18]. All CLS2NS models are trained using only the text corresponding to each language's 200 hours of training data. Note that while training we use ground truth CLS and native script. As we will see, using CLS2NS provides additional gains over performance in CLS space. Additionally, the CLS2NS decoder also serves to rectify any errors that may occur in the CLS space. One such example is shown in Figure 1. In the next section, we investigate the explicit use of language information to improve performance.

इसलिए हम दोनों: Ground truth in Native script
isaliee ham donoq: Ground truth in CLS script

isaliE ham dono: ASR CLS output
इसलिए हम दोनों: CLS2NS converter output

Figure 1: *CLS2NS rectifies errors caused by the CLS ASR model*

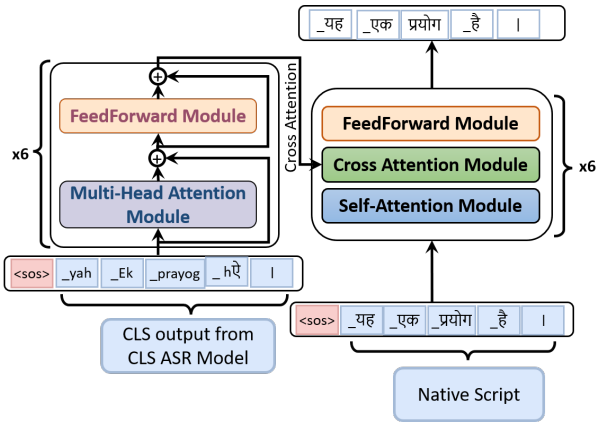

Figure 2: *A Sequence-to-Sequence Transformer model is trained for CLS2NS to convert CLS text to Native Script text*

4.4. Monolingual model with Language model

One may argue that the decoder of the CLS2NS acts like a language model for that particular language. To justify the use of the CLS2NS over simply adding an external language model in ASR model, we compared the results of the Monolingual model with the external language model and CLS ASR model with CLS2NS. In this setup training data text is only used to train the language model for each language. All Language models are based on the Transformer framework with the configuration as follows: attention units 1024, encoder units 2048, encoder layers 16, attention head 8, batchbins 350000, and learning rate 0.001 with warumplr learning rate scheduler.

5. Multilingual ASR Models using Language ID Token

5.1. Native Script with Language ID Multilingual ASR Models

A different way to introduce language-specific information into an ASR model is to include *language identification* (LID) tokens in the target text. This approach involves training the ASR model with the LID token, which allows the model to better distinguish between each language's unique linguistic features and sound patterns. For instance, in the LID tags, <**gujarati** refers to the tag being appended at the beginning and corresponds to the language Gujarati. The experiment setup for this experiment is as shown in Figure 3 and embedding for LID are learnt during training. The method of using language id is shown in Table 4.

Language	Native Script	CLS	CLS + LID
Hindi	इस अध्याय	is aधyAy	<hindi is aधyAy
Gujarati	સમાચાર અમિતા	samAcAr amitA	<gujarati samAcAr amitA
Marathi	वरच्या खड्यात	waracyA khaडyAt	<marathi waracyA khaडyAt
Bengali	প্রচুর পাঠক	pracur pAঠak	<bengali pracur pAঠak
Odia	ଯିବା କଟକୁ	yibA kaटku	<odia yibA kaटku

Table 4: *In multilingual the reference transcription is provided in their native scripts while in CLS they are in the common script. LID provides the language information while training*

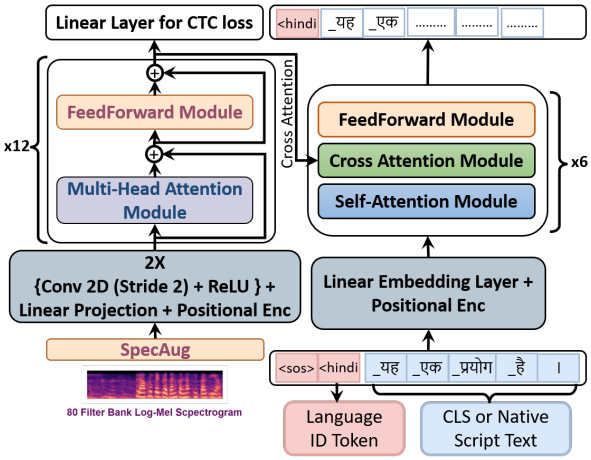

Figure 3: *Experimental setup for the Multilingual ASR model with LID or CLS multilingual ASR with LID*

5.2. Common Label Set with Language ID Multilingual ASR model

A major limitation of the baseline CLS-based ASR model is that it does not have any language information while decoding the test utterance. In this study, we have investigated the use of Language Identification (LID) tokens in conjunction with CLS text during the training process. To achieve this, the transcription text was modified as illustrated in Table 4. Specifically, the CLS+LID column shows samples from the six Indian languages with corresponding LID tokens added to the beginning of the transcription text. The use of LID tokens in the decoder may be interpreted as providing means for the model to learn language aspects of that language.

5.2.1. CLS2NS for CLS+LID

To convert the output of the CLS+LID model from CLS text to Native script text, mono CLS2NS or unified CLS2NS can be used. Mono CLS2NS is the same as described in section 4.3. The Unified CLS2NS approach entails training the CLS2NS model using combined text data from all languages. This results in a single, comprehensive model that can be employed to seamlessly convert text from the CLS space to the native script of any language.

Experiment	Hindi	Gujarati	Marathi	Bengali	Odia
Monolingual	17	26.3	52.2	23	32.7
Monolingual + LM	16.7	25.7	51.5	22.3	32.2
Multilungual Native Script	16.8	28.4	52.2	23.1	31.1
Multilingual CLS	16	27.1	50.9	22	29.5
+ CLS2NS	16	24.61	44.74	18.46	23.57
Multilingual Native script with LID	15.4	24.5	45.2	20.7	29
Multilingual CLS with LID	**14.2**	**22.8**	43.9	19.5	27
+ CLS2NS	15.54	23.4	**43.43**	**18.1**	**23.45**
Multilingual CLS with LID + Unified CLS2NS	15.94	25.34	44.76	18.27	23.74

Table 5: *WER comparison of Baseline and various Multilingual ASR model [No external LM is used except for the Monolingual + LM experiment]*

6. Results and Discussion

Results for all experiments are shown in the Table 5. We begin by comparing the multilingual NS model and the multilingual CLS model with the monolingual models. We can see that the multilingual NS model is slightly better or competitive to the monolingual models. This is mainly because the encoder of the multilingual model sees more data than that of the monolingual model. But, the decoder of the multilingual model has to learn to output the text in native script of various languages and the search space of the decoder becomes large. This is the reason why it is sometimes inferior to the monolingual model. On the other hand, the multilingual CLS model, as given in row 5 of Table 5, is consistently better than the monolingual models as well as the multilingual NS model. But, these results are evaluated for the CLS text as targets.

In practice, it is more useful to evaluate the performance in native script. To convert the CLS text to NS text, we train a CLS2NS as described in Section 4.3 for each language separately. Row 6 of Table 5 shows the performance of multilingual CLS model whose outputs are converted to native script by the use of CLS2NS models and it consistently outperforms the aforementioned models. We argue that the jump in performance after applying the CLS2NS is due to its language modeling capability to an extent. More specifically, we hypothesize that the decoder of the CLS2NS acts as a language model.

Past works have suggested that the use of LID tokens in multilingual systems is beneficial. We wanted to know if we can get similar benefits in multilingual CLS models by adding LID tokens to the CLS text. As we can see from Table 5, adding LID tokens improves the performance of both the multilingual systems as expected. Even though the multilingual models trained with LID tokens can predict the LID during inference, for the results presented in this work, we always provide the ground truth LID since we know the language being inferenced. It is observed that these results are better than the multilingual native script with LID model. When we convert the CLS text to native script, the performance improved for three languages and degraded a little for the other two languages. In cases where the performance has improved, the improvement is not as large as compared to the non-LID counterpart. The reason for this is that adding LID to the target text already improves the language modeling power of the decoder for a given language. On top of that, the CLS2NS does not have much to offer in terms of the language model.

Finally, we wanted to see if we can unify the CLS2NS of various languages into a single model. So, we trained a unified CLS2NS which can convert the CLS text of all languages to its corresponding native script. We can see that its performance is competitive to the individual CLS2NS with little degradation. But the advantage is that we can replace the individual models with a single unified model.

Experiment	Hindi	Gujarati	Marathi
Monolingual	34.9	43.1	57.8
Multilingual CLS+LID +CLS2NS	31.82	32.04	37.7

Table 6: *WER comparison for FLUERS test data [out-of-distribution]*

Additionally, we evaluated our monolingual system on an out-of-distribution dataset. We evaluated three languages from FLEURS dataset [19]. The results are shown in Table 6. It can be clearly seen that same pattern emerges for an out-of-distribution test data also. Multilingual CLS and Multilingual CLS+LID are more robust than the monolingual systems and show up to 20.1 % absolute improvement in WER.

Overall, we observed up to 5.55% absolute improvement in WER compared to the multilingual with LID baseline and up to 9.25% absolute improvement in WER compared to the monolingual baseline.

7. Conclusion

This paper discusses the methods in which one can leverage the phonetic similarities of different languages (like Indian languages) and build a Multilingual ASR model. The data from all the languages are pooled together and phonetically similar speech sounds are assigned to Common Label Set. This reduces the number of target units for ASR. A CLS2NS is built for each language that decodes CLS text to native script text. Experiments performed show that CLS-ASR model gives significant improvement over the baseline monolinguistic and multilingual ASR models without CLS. Further, another method to include language-specific information to the ASR model is discussed by means of Language Identification tokens. The LID tokens are passed to the model for training in two ways: with Native Script and with CLS. To convert the output of the CLS+LID model into native script text, mono and unified CLS2NS converters are used. The experimental results show that CLS+LID model performs best among all the experiments. The Multilingual CLS and Multilingual CLS+LID models are giving the best performance also on out-of-distribution test dataset FLEURS.

8. Acknowledgements

We would like to thank the Ministry of Electronics and Information Technology (MeitY), Government of India, for providing us with the compute resources as a part of the "Bhashini" project.

9. References

[1] T. Schultz and A. Waibel, "Fast bootstrapping of lvcsr systems with multilingual phoneme sets," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[2] S. Thomas, S. Ganapathy, and H. Hermansky, "Multilingual mlp features for low-resource lvcsr systems," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4269–4272.

[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)*. IEEE, 2017, pp. 265–271.

[4] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4904–4908.

[5] A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, Z. Chen, and S. Lee, "Large-scale multilingual speech recognition with a streaming end-to-end model," *arXiv preprint arXiv:1909.05330*, 2019.

[6] S. Sivasankaran, B. M. L. Srivastava, S. Sitaram, K. Bali, and M. Choudhury, "Phone merging for code-switched speech recognition," in *Third Workshop on Computational Approaches to Linguistic Code-switching*, 2018.

[7] K. Dhawan, G. Sreeram, K. Priyadarshi, and R. Sinha, "Investigating target set reduction for end-to-end speech recognition of hindi-english code-switching data," in *2020 National Conference on Communications (NCC)*. IEEE, 2020, pp. 1–5.

[8] A. Prakash, A. L. Thomas, S. Umesh, and H. A. Murthy, "Building multilingual end-to-end speech synthesisers for indian languages," in *Proc. of 10th ISCA Speech Synthesis Workshop (SSW'10)*, 2019, pp. 194–199.

[9] V. M. Shetty, M. Sagaya Mary N J, and S. Umesh, "Exploring the use of common label set to improve speech recognition of low resource indian languages," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[10] V. M. Shetty and M. Sagaya Mary N.J., "Improving the performance of transformer based low resource speech recognition for indian languages," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8279–8283.

[11] D.-C. Lyu and R.-Y. Lyu, "Language identification on code-switching utterances using multiple cues," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[12] K. R. Mabokela and M. J. Manamela, "An integrated language identification for code-switched speech using decoded-phonemes and support vector machine," in *2013 7th Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. IEEE, 2013, pp. 1–6.

[13] J. Gonzalez-Dominguez, D. Eustis, I. Lopez-Moreno, A. Senior, F. Beaufays, and P. J. Moreno, "A real-time end-to-end multilingual speech recognition architecture," *IEEE Journal of selected topics in signal processing*, vol. 9, no. 4, pp. 749–759, 2014.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[16] B. Ramani, S. L. Christina, G. A. Rachel, V. S. Solomi, M. K. Nandwana, A. Prakash, S. A. Shanmugam, R. Krishnan, S. K. Prahalad, K. Samudravijaya, P. Vijayalakshmi, T. Nagarajan, and H. A. Murthy, "A common attribute based unified HTS framework for speech synthesis in Indian languages," in *Proc. 8th ISCA Workshop on Speech Synthesis (SSW 8)*, 2013, pp. 291–296.

[17] A. Baby, N. N.L., A. L. Thomas, and H. A. Murthy, "A unified parser for developing indian language text to speech synthesizers," in *Text, Speech, and Dialogue*, P. Sojka, A. Horák, I. Kopeček, and K. Pala, Eds. Cham: Springer International Publishing, 2016, pp. 514–521.

[18] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[19] A. Conneau, M. Ma, S. Khanuja, Y. Zhang, V. Axelrod, S. Dalmia, J. Riesa, C. Rivera, and A. Bapna, "Fleurs: Few-shot learning evaluation of universal representations of speech," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 798–805.