IMPROVING THE PERFORMANCE OF TRANSFORMER BASED LOW RESOURCE SPEECH RECOGNITION FOR INDIAN LANGUAGES

Vishwas M. Shetty, Metilda Sagaya Mary N J, S. Umesh

Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, India

ABSTRACT

The recent success of the Transformer based sequence-to-sequence framework for various Natural Language Processing tasks has motivated its application to Automatic Speech Recognition. In this work, we explore the application of Transformers on low resource Indian languages in a multilingual framework. We explore various methods to incorporate language information into a multilingual Transformer, i.e., (i) at the decoder, (ii) at the encoder. These methods include using language identity tokens or providing language information to the acoustic vectors. Language information to the acoustic vectors can be given in the form of one hot vector or by learning a language embedding. From our experiments, we observed that providing language identity always improved performance. The language embedding learned from our proposed approach, when added to the acoustic feature vector, gave the best result. The proposed approach with retraining gave 6% - 11% relative improvements in character error rates over the monolingual baseline.

Index Terms— Transformer, Automatic Speech Recognition, Multilingual, Low Resource

1. INTRODUCTION

India majorly comprises of a polyglot populace. In addition to accent and dialectal variations, informal conversations are a blend of multiple languages. Given the vast market that India is, building an efficient speech recognizer for each of these languages is of paramount importance. Apart from Hindi and Indian English, there is a dearth of speech corpus in other Indian languages. Hence, building a separate speech recognizer for each of these languages is tricky.

In the conventional ASR approach, for low resource languages, a deep neural network (DNN) is trained using all the languages. This network can be used as a feature extractor for a language-specific task, or its initial layers shared among multiple languages [1, 2, 3]. The conventional ASR approaches also involve multiple components like an acoustic model, a pronunciation dictionary, and a language model. These are significant bottlenecks in the case of low resource languages.

The end-to-end (E2E) approach, on the other hand, integrates acoustic, pronunciation, and language models into one single framework. In [4], a single E2E speech recognizer was trained based on Listen, Attend and Spell (LAS) [5]. In [6], a universal speech recognizer for ten different languages was built based on a hybrid CTC/attention criterion. In [4, 6], training and testing conditioned on the language identity (LID) improved the performance as well as reduced confusion among different languages. [7] discussed various retraining approaches for a multilingual E2E model. The approaches discussed in [4, 6, 7] were in an RNN based E2E framework.

Transformer [8] is another sequence to sequence approach, solely based on attention. Its success in natural language processing tasks has motivated its application to ASR tasks as well [9, 10, 11, 12]. The Transformer performed better than the conventional RNN based encoder-decoder for a multilingual ASR task [13]. It was implemented for multilingual speech recognition in a low resource scenario in [14].

In our paper, we have worked on low resource Indian languages in the Transformer framework. Maintaining the essence of E2E models, we have not made use of pronunciation dictionary and language models in our experiments. Each utterance in our dataset belongs to one of the languages. There is no switching of languages within an utterance. From our experiments, we have observed that the Transformer significantly outperforms conventional RNN based models.

2. PROPOSED METHODS

In our work, we explore ways to include language information at the decoder (LID as an output token) and the encoder (LID at the feature level). Our approach tries to address two problems in low resource E2E speech recognition, each of these is explained in detail in further sections.

- Building a language-independent, i.e., universal speech recognizer. Such a model would not require a separate language identification module. In a country like India where the urban crowd converses in more than one language, deploying a universal recognizer would be preferred.
- Given prior knowledge of the target language, we explore methods to improve the performance of the speech models over the baseline monolingual systems.

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: MINCYT. Downloaded on July 16,2020 at 21:05:52 UTC from IEEE Xplore. Restrictions apply.

3. DATA SET AND EXPERIMENTAL DETAILS

We report results on the data set released by Microsoft and SpeechOcean.com as part of a special session on "Low resource speech recognition challenge on Indian languages in INTERSPEECH 2018". It includes data in three Indian languages - Gujarati, Tamil, and Telugu. The data consists of wave files sampled at 16kHz along with corresponding text transcriptions in UTF-8. Details of the data set are given in Table 1.

Language		Train	Dev	Eval
Gujarati	Duration(hrs)	40	5	5
	# Utterances	22807	3075	3419
Tamil	Duration(hrs)	40	5	5
	# Utterances	39131	3081	2609
Telugu	Duration(hrs)	40	5	5
	# Utterances	44882	3040	2549

Table 1: Indian languages data set statistics

Our models are trained using 80-dimensional filter bank features along with three-dimensional pitch features. The experiments are based on the Hybrid CTC/attention approach [15]. Given input acoustic feature vectors X and the corresponding output character sequence C, the network is trained within the Multi-Objective Learning (MOL) framework. The training objective function, L_{MOL} for the network is a logarithmic linear combination of attention and CTC objectives, i.e.

$$L_{MOL} = \lambda log p_{ctc}(C|X) + (1 - \lambda) log p^{*}_{att}(C|X)$$

where λ, the multitask learning coefficient, is a tunable parameter that decides the contribution from CTC and attention components. λ satisfies $0 \leq \lambda \leq 1$. The experiments are run on a GTX 1080 GPU card using Kaldi [16] and Espnet [17] tool kits.

The RNN based E2E model has a four-layered encoder network. Each encoder layer comprises of 320 bidirectional long short-term memory (BLSTM) units with sub-sampling. "Location" aware attention is used. One layer decoder network with 300 long short-term memory (LSTM) units is implemented. The multi-task learning co-efficient λ is set to 0.5 while training, giving equal importance to both CTC and attention. While decoding, λ is set to 0.3.

Transformer based encoder-decoder models have a 12 layer encoder network with 2048 units. A single layer decoder with 1024 units is used. Each layer has four attention heads, with each head having an attention dimension of 64. The outputs of the attention heads are concatenated to give a 256 dimension attention vector. A batch size of 32 is used with gradient being accumulated after every four iterations. "noam" optimizer (Section 5.3 in [8]) with a learning rate of 10 is used. Number of warmup steps is fixed at 25000. All our Transformer experiments are run for 150 epochs. The average of the models from the last ten epochs is used while decoding. Multitask learning parameter λ is set to 0.3 both during training and decoding.

4. BUILDING A UNIVERSAL SPEECH RECOGNIZER

We pooled in data from all the three languages and built a universal speech recognizer. The token set used for this model was a union of tokens from the three languages [6]. In our case, there were 64, 48, and 64 language tokens (excluding punctuation marks, <blank> and <space>) in Gujarati, Tamil, and Telugu, respectively. There was no overlap among the tokens from the three languages. Our universal recognizer had 181 unique tokens including punctuation marks, <blank>, <space> and <eos>. The universal recognizer's performance was comparable to that of the corresponding monolingual speech recognizer. The results from our experiments are shown in Tables 2 and 3. Here ***Mono*** and ***Multi*** refer to monolingual and multilingual models, respectively. ***RNN*** and ***Trans*** indicate RNN and Transformer based E2E frameworks, respectively. As seen in the tables, merely using a Transformer gave a substantial boost to the model performance over using the RNN based E2E framework. As also observed in [6, 18], we noted that the multilingual model performance slightly degraded for a few languages. The models discussed so far were given no information on the language either during training or decoding.

	Gujarati		Tamil		Telugu	
	Dev	Eval	Dev	Eval	Dev	Eval
Mono RNN	45.3	50.2	39.6	38.8	43.9	45.6
Multi RNN	40.0	48.8	39.6	38.7	42.9	43.7
Mono Trans	31.9	40.6	34.1	33.4	36.0	36.4
Multi Trans	30.9	39.8	34.7	34.0	36.4	37.0
LID Trans	30.0	39.4	33.9	33.2	35.4	35.9

Table 2: Universal model performance(% WER)

	Gujarati		Tamil		Telugu	
	Dev	Eval	Dev	Eval	Dev	Eval
Mono RNN	16.1	20.0	9.7	9.2	12.0	12.3
Multi RNN	13.5	20.7	9.5	9.1	11.4	11.6
Mono Trans	9.6	14.2	7.7	7.3	8.6	8.8
Multi Trans	9.1	13.7	7.7	7.3	8.5	8.8
LID Trans	8.9	13.5	7.5	7.2	8.3	8.6

Table 3: Universal model performance(% CER)

The performance of ***Multi Trans*** can be improved by providing language information while training. We refer to this model as ***LID Trans***. The model *LID Trans* was trained with

Authorized licensed use limited to: MINCYT. Downloaded on July 16,2020 at 21:05:52 UTC from IEEE Xplore. Restrictions apply.

Language	Example
Gujarati	<guj_beg> આતંકી હુમાંલો હુમલો <guj_end>
Tamil	<tam_beg> கண்டிப்பா கண்டிப்பா <tam_end>
Telugu	<tlg_beg> అనంతరం <tlg_end>

Table 4: Examples of LID in the target sequence

language identity tokens in the beginning and end of the target sequence of each utterance. A few examples are given in Table 4. Since the LID tokens here are part of the target sequence, they are also taken into account while performing self-attention (Section 3.2 of [8]) in the decoder.

The results for *LID Trans* are given in Tables 2 and 3. *LID Trans* is a "Universal speech recognizer," in a sense that no language identity is provided during decode. Despite the absence of LID while decoding, *LID Trans* almost always picked the right language. It was noted that even if occasionally a wrong language was picked, the decoded output was a transliterated version of the reference text. The proposed model *LID Trans* outperformed the baseline monolingual model *Mono Trans* and multilingual model *Multi Trans*.

5. IMPROVING THE PERFORMANCE OF THE MONOLINGUAL SYSTEMS

Next, we explored methods to improve the recognition accuracy of the recognizer given prior knowledge of the target language. One way to do this is to force decode from a model like *LID trans*, to start with the target LID. This broadly implies conditioning the decoder with the LID. However, conditioning the encoder (as opposed to the decoder) with LID information [4] was found to give maximum benefit in RNN based encoder-decoder framework. Motivated by this, we propose two methods to condition the encoder with the language information in the transformer framework.

- Append one hot vector to the feature vectors.
- Learn a language embedding for each of the languages and add it to the feature vectors.

In a conventional Transformer along with the attention within the encoder and decoder blocks, i.e., self-attention, attention weights are also computed using the encoder block outputs and the decoder block character embedding vectors as shown in Figures 1 and 2. This is like cross attention. Hence, any language-specific information provided at the encoder propagates to the decoder via cross attention.

5.1. Appending one hot vector

By appending LID vectors to the acoustic features [19], language information can be provided at the feature level. Since we were dealing with three languages, we made use of three dimensional one-hot vectors. Feature vectors from each of the three languages were appended with a one-hot vector, such that features from the same language got the same vector, as shown in Figure 1. This model is referred to as ***Onehot Trans*** in this paper. This approach gave considerable improvements in performance over the baseline model *Mono Trans* as shown in Tables 5 and 6.

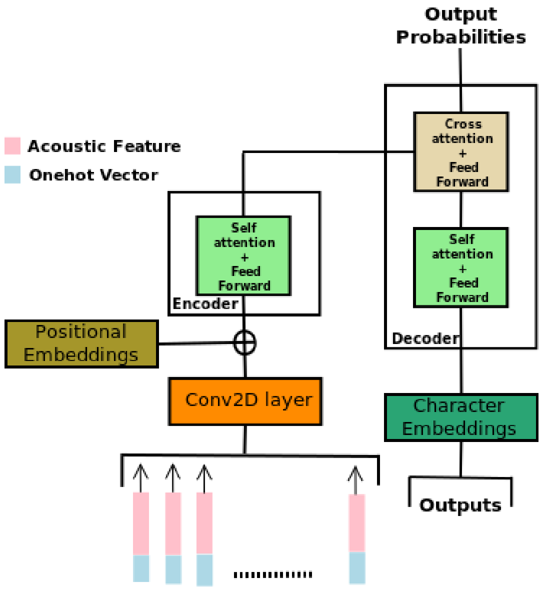

Fig. 1: Appending onehot vector to acoustic feature frames

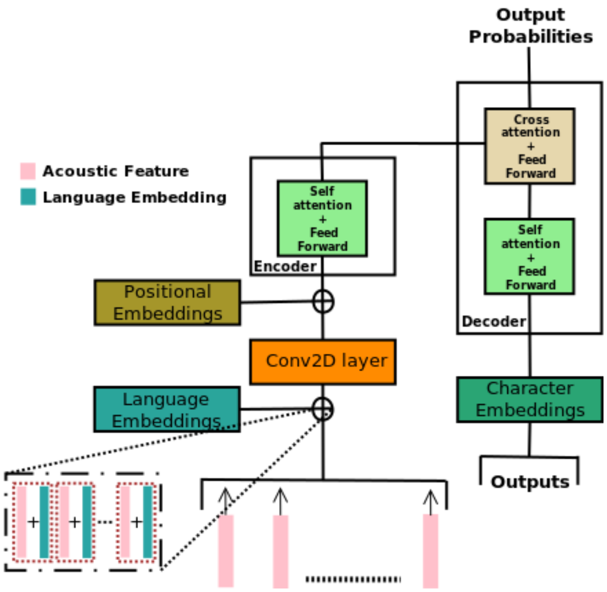

Fig. 2: Adding learnt language embedding vector to acoustic feature frames

5.2. Learning a language embedding

In the second approach, we learn a language embedding for each of the languages. In a conventional Transformer, at the

Authorized licensed use limited to: MINCYT. Downloaded on July 16,2020 at 21:05:52 UTC from IEEE Xplore. Restrictions apply.

	Gujarati		Tamil		Telugu	
	Dev	Eval	Dev	Eval	Dev	Eval
Mono Trans	31.9	40.6	34.1	33.4	36.0	36.4
Onehot Trans	29.4	38.3	33.9	32.8	35.2	35.9
Lang_embed Trans	29.2	37.7	33.2	32.4	34.8	35.3
Lang_embed Trans + LID	28.8	37.4	33.4	32.4	34.5	35.1

Table 5: Language specific model performance (% WER)

	Gujarati		Tamil		Telugu	
	Dev	Eval	Dev	Eval	Dev	Eval
Mono Trans	9.6	14.2	7.7	7.3	8.6	8.8
Onehot Trans	8.6	13.0	7.4	7.1	8.2	8.5
Lang_embed Trans	8.6	12.8	7.3	7.0	8.1	8.3
Lang_embed Trans + LID	8.3	12.7	7.3	7.0	8.1	8.3

Table 6: Language specific model performance (% CER)

decoder end, a token/character embedding matrix, that generates a fixed dimension vector for each of the target tokens is learned. Similarly, in our proposed method, at the encoder side, we learn a language embedding matrix, that would generate a language embedding for each of the languages. As mentioned earlier, each utterance comprises of only one language. Hence, for a given utterance, the desired targets would be tokens from only one of the languages. We have made use of this fact to learn the language embedding matrix. The language embedding has the same dimension as that of the feature vector. We add the learned language embedding to every feature vector, as shown in Figure 2. Feature vectors from a language are added with the corresponding language embedding vector. This model is referred to as ***Lang_embed Trans*** in this paper. *Lang_embed Trans* was found to improve performance accuracy even over *Onehot Trans*, as shown in Tables 5 and 6. *Lang_embed Trans* makes use of the language embedding learned during training while decoding a new utterance. Due to this additional language information, it was observed that there was no confusion among languages during decode. ***Lang_embed Trans + LID*** was an extension to *Lang_embed Trans* with LID tokens in the target sequence as in *LID Trans*. It was not given LID tokens while decoding. Its results were similar to that of *Lang_embed Trans*.

5.3. Retraining

With the proposed universal (*Multi Trans* and *LID Trans*) and the language-specific (*Onehot* and *Lang_embed Trans*) models as the starting point, we retrained the model with only the target language. This retraining of models with only the desired language gave an additional boost to the model performance. The retrained model results are given in Tables 7 and 8. There was no confusion between languages in any of the retrained models either.

	Gujarati		Tamil		Telugu	
	Dev	Eval	Dev	Eval	Dev	Eval
Mono Trans	31.9	40.6	34.1	33.4	36.0	36.4
Multi Trans + retrain	29.6	38.6	33.1	32.3	34.5	34.9
LID Trans+ retrain	29.3	38.4	33.0	32.4	34.0	34.7
Onehot Trans + retrain	28.8	37.9	32.4	31.6	33.9	34.7
Lang_embed Trans + retrain	28.2	36.9	32.1	31.3	33.0	33.1

Table 7: Retrained model performance(% WER)

	Gujarati		Tamil		Telugu	
	Dev	Eval	Dev	Eval	Dev	Eval
Mono Trans	9.6	14.2	7.7	7.3	8.6	8.8
Multi Trans + retrain	8.8	13.3	7.3	7.0	8.1	8.3
LID Trans + retrain	8.6	13.3	7.3	7.0	7.9	8.2
Onehot Trans + retrain	8.5	13.0	7.2	6.8	7.9	8.2
Lang_embed Trans + retrain	8.3	12.6	7.1	6.8	7.7	7.8

Table 8: Retrained model performance(% CER)

6. CONCLUSION

In this paper, we have explored different methods to deal with low resource speech recognition in a Transformer framework. Two different ways of incorporating language information into the system have been proposed. Figure 3 shows a comparison between the monolingual baseline and the proposed approaches. We see that the proposed approach of learning the language embedding along with retraining gives the best accuracy.

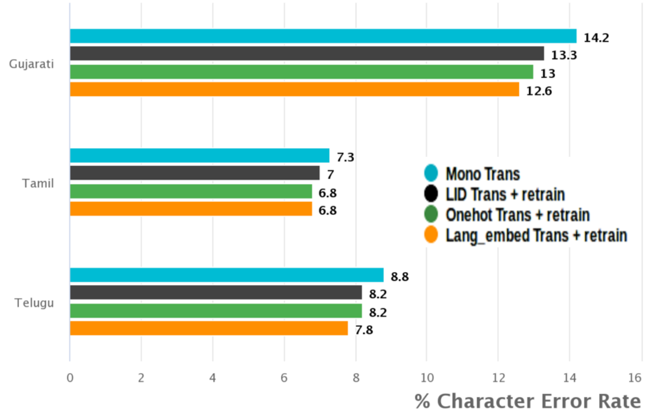

Fig. 3: Monolingual Baseline v/s proposed learnt language embedding performance on Eval set

7. REFERENCES

[1] Karel Veselỳ, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova, "The language-

Authorized licensed use limited to: MINCYT. Downloaded on July 16,2020 at 21:05:52 UTC from IEEE Xplore. Restrictions apply.

independent bottleneck features," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 336–341.

[2] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual training of deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7319–7323.

[3] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7304–7308.

[4] Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao, "Multilingual speech recognition with a single end-to-end model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4904–4908.

[5] William Chan, Navdeep Jaitly, Quoc Le, and Oriol 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)*. IEEE, 2016, pp. 4960–4964.

[6] Shinji Watanabe, Takaaki Hori, and John 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.

[7] Jaejin Cho, Murali Karthick Baskar, Ruizhi Li, Matthew Wiesner, Sri Harish Mallidi, Nelson Yalta, Martin Karafiat, Shinji Watanabe, and Takaaki Hori, "Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 521–527.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[9] Linhao Dong, Shuang Xu, and Bo Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[10] Shiyu Zhou, Linhao Dong, Shuang Xu, and Bo Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," *arXiv preprint arXiv:1804.10752*, 2018.

[11] Abdelrahman Mohamed, Dmytro Okhonko, and Luke Zettlemoyer, "Transformers with convolutional context for asr," *arXiv preprint arXiv:1904.11660*, 2019.

[12] Jie Li, Xiaorui Wang, Yan Li, et al., "The speech transformer for large-scale mandarin chinese speech recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7095–7099.

[13] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al., "A comparative study on transformer vs rnn in speech applications," *arXiv preprint arXiv:1909.06317*, 2019.

[14] Shiyu Zhou, Shuang Xu, and Bo Xu, "Multilingual end-to-end speech recognition with a single transformer on low-resource languages," *arXiv preprint arXiv:1806.05059*, 2018.

[15] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[16] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[17] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "Espnet: End-to-end speech processing toolkit," in *Interspeech*, 2018, pp. 2207–2211.

[18] Suyoun Kim and Michael L Seltzer, "Towards language-universal end-to-end speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4914–4918.

[19] Markus Müller, Sebastian Stüker, and Alex Waibel, "Language adaptive dnns for improved low resource speech recognition.," in *Interspeech*, 2016, pp. 3878–3882.

Authorized licensed use limited to: MINCYT. Downloaded on July 16,2020 at 21:05:52 UTC from IEEE Xplore. Restrictions apply.