Unlocking Voices, Unleashing Possibilities: Your Words, Our Recognition!

Sasikanth Kotti(P23CS0007)[1], *Venkata Brahmanandarao Nelluri(M22AI1014)*[1], *Sai Krishna Reddy Sanda(M22AIE225)*[1]

[1]IIT Jodhpur, India

p23cs0007@iitj.ac.in, m22ai1014@iitj.ac.in, m22aie225@iitj.ac.in

1. Introduction and Motivation

ASR systems are used to convert audio signals to text. The transcribed text can be used in downstream applications to analyse sentiment, understand intent among others. This may be the pipeline behind voice assistant systems such as Siri , Alexa and others. Alternatively, Automatic Speech Recognition (ASR) systems can also help with captioning in videos. Hence, high quality and error free ASR systems are important for overall performance of voice assistants and other systems. Speech to Text is also a challenging task due to the associated linguistic and text rules that are specific to each language. Example for this is the pronunciation of different words which are not one-one mapping for pronunciation in English. Hence, ASR is still an unsolved problem despite good performance. ASR in the context of noisy environment is of importance for real world deployments.

2. Objective

The objective of this project is to understand the performance existing ASR systems on noisy data and explore architectural improvements for the same.

3. Brief Literature Review

- **An Embarrassingly Simple Approach for LLM with Strong ASR Capacity** [1], In this work the authors proposed an LLM based ASR system called as SLAM-ASR. It is a simple composition of available speech encoder, LLM and linear project which is trainable.
- **Branchformer: Parallel MLP-Attention Architectures to Capture Local and Global Context for Speech Recognition and Understanding**[2], In this the authors proposed an architecture inspired by Conformer, consisting of parallel branches for modelling various ranged dependencies for speech recognition. This is achieved via attention modules.
- **Revisiting End-to-End Speech-to-Text Translation From Scratch**[3], Authors analysed the performance of End-to-end (E2E) speech-to-text translation (ST) without pre-training. Along with they also proposed best practices for better performance.
- **Self-supervised learning with random-projection quantizer for speech recognition**[4], This work proposed flexible and effective method of self-supervised learning approach for speech recognition. This is based on prediction of masked speech signals.
- **It's Raw! Audio Generation with State-Space Models**[5], This work proposed SaShiMi, a new multiscale architecture which is an improvement of s4 model for modelling raw audio waveforms.
- **Robust Speech Recognition via Large-Scale Weak Supervision**[6], The authors presents a study where the capabilities of speech processing systems are analysed when trained on internet scale audio transcripts.
- **Pre-training for Speech Translation: CTC Meets Optimal Transport**[7], Authors identified that pre-training with connectionist temporal classification (CTC) loss enables better final speech-to-text translation (ST)accuracy. Additionally they also proposed CTC combined with optimal transport to further increase performance.

4. Datasets

LibriSpeech ASR corpus [8] which is a corpus of 1000 hours of 16kHz read English speech is considered for this project. However, this dataset doesn't represent real world settings where background noises are common.

Hence, a noisy version of LibriSpeech called as **Noisy LibriSpeech** is constructed by combing **wham noise** with vanilla **LibriSpeech**.

The **Noisy LibriSpeech** consists of training set of 100 hours ”noisy” speech, development set of ”noisy” speech and test set of ”noisy” speech.All the pretrained models are evaluated using the test set of ”noisy” speech for performance.

5. Evaluation metrics

The performance of the systems are evaluated using WordErrorRate(WER).

The WordErrorRate(WER) is a common metric used for evaluation of speech systems. This is defined as below :

$$\mathbf{WER} = \frac{\mathbf{S} + \mathbf{D} + \mathbf{I}}{\mathbf{N}}$$

where

- S, D, I indicate substitutions, deletions and insertions
- C number of correct words and N number of reference words

6. Proposed Algorithm

In this section we briefly describe the pretrained architectures that are evaluated for performance with **Noisy LibriSpeech**

6.1. wav2vec2.0

As proposed by [9] , this framework was the first to show that pretrained by large corpus and then followed by finetuning even with small dataset in speech domain outperformed the best semi-supervised methods. This approach was simpler with just a two step process for training. This consists of 4 modules, namely feature encoder, context network, quantization module, and the contrastive loss. The contrastive loss is used as a pretraining objective. The feature encoder is a simple 7-layer convolutional neural network with 512 channels at each layer. The quantization module enables learning discretization of the continuous speech signal by sampling from Gumbel-Softmax distribution. The context network is a transformer encoder where relative positional embeddings are learnt with grouped convolution layer. Pretraining is performed with contrastive loss. finetuning is then performed on the target task. The authors utilized CTC loss for finetuning the model for speech recognition.

6.2. Conformer

Authors [**?**] in this work exploited the properties of both transformers and convolution neural networks for speech recognition. Transformers utilized attention mechanism and also parallelize the computation of sequential tasks. In Conformer the authors combined transformers with convolution layers so that the network can capture both local and global context.

Conformer uses Multi-Headed self-attention with Relative Positional Embedding in the Conformer block. These along with convolution and feed forward layers are stacked together to form Conformer Encoder. This Encoder can then be combined with any Language Model(LM) Decoder for speech recognition task.

6.3. Branchformer

In this work a variant of Branchformer [10] namely E-Branchformer for speech recognition is considered for evaluation. The E-Branchformer [11]. In Branchformer [10] , context from different modules such transformer and convoutions are combined point-wise and linearly. However, in E-Branchformer context merging was further enhanced by modifying the merge module to take into consideration the temporal information into account. This includes the usage of Depth-wise Convolution and Squeeze-and-Excitation modules to merge contexts. This architecture showed good performance with LibriSpeech dataset for speech recognition.

6.4. TSConformer

The Conformer architecture utilizes Multi-Headed self-attention with Relative Positional Embedding as the basic block for extracting global context. While self-attention being the work horse of transformed showed good performance, they still suffer from quadratic complexity with sequence length.

In the recent work Zoology: Measuring and Improving Recall in Efficient Language Models [12] authors explored two sub-quadratic operators: short convolutions and linear attention. However, these were explored to improve associative recall (AR) for language modelling and in-context modelling.

[13] proposed that Linear Attention can be obtained from scaled dot product attention by removing exponential terms along with rearrangement. However, this showed sub-optimal results for associative recall (AR), hence authors [12] proposed to approximate the exponential terms with Taylor Series. This is referred to as Taylor Series Linear Attention. This has sub-quadratic complexity although not perfectly linear.

In TSConformer, we propose to use Taylor Series Linear Attention instead of the standard Multi-Headed self-attention with Relative Positional Embedding in the Conformer block. This is shown in Figure 1 1

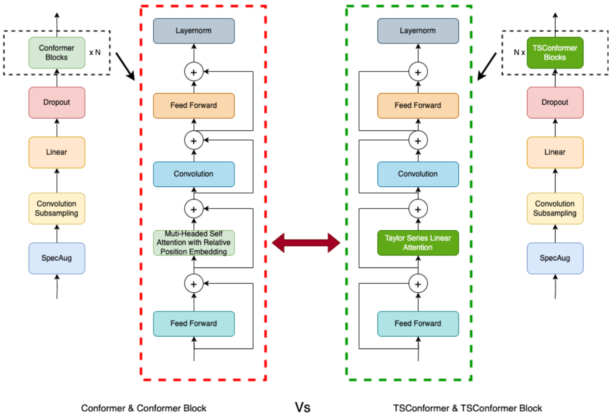

Figure 1: *TSConformer*

We hypothesise that this may result in better inference speeds along with comparable performance to vanilla Conformer.

7. Experiments

The models pretrained on vanilla LibriSpeech dataset for architectures wav2vec2.0, Conformer and E-Branchformer were obtained from Huggingface and were evaluated on test set of ”noisy” speech

Of these Conformer model pretrained on vanilla LibriSpeech dataset is finetuned on training set of 100 hours of **Noisy LibriSpeech** dataset. The transformer decoder is unfrozen and finetuning is carried out with Adam optimizer (betas of 0.9, 0.98 and eps of 0.000000001) for 50 epochs with an initial learning rate of 0.0008. Evaluation of the finetuned model is performed using test set of ”noisy” speech.

The proposed TSConformer is finetuned on training set of 100 hours of **Noisy LibriSpeech** dataset by unfreezing Taylor Series Linear Attention layers. All other layers were kept frozen. finetuning was carried with Adam optimizer (betas of 0.9, 0.98 and eps of 0.000000001) for 80 epochs with an intial learning rate of 0.008. Evaluation was carried out using test set of ”noisy” speech.

A batch size of 16 was used when finetuning both Conformer and TSConformer.

All the finetuning steps were carried out on AWS EC2 instance with A10 GPU. finetuning for 80 epochs took more than 12 hours of GPU time for execution.

8. Results

From table1, it is evident that pretrained wav2vec2.0 showed better performance i.e. lowest WER and CER when evaluated on test partition of **Noisy LibriSpeech**. However, the performance of pretrained Conformer is comparable to wav2vec2.0. Hence, Conformer architecture is updated to create TSConformer.

Table2, shows that TSConformer performance is worse than Conformer, after both are finetuned on **Noisy LibriSpeech**. The performance of TSConformer degraded and became worse. Hence, vanilla Taylor Series Linear Attention was not helpful. Further analysis is needed to understand the observed performance degradation.

Table3, shows total MACs (Multiply-Accumulate Operations) for Conformer and TSConformer. TSConformer appears

Pretrained Models			
Metrics	wav2vec2.0	Conformer	Branchformer
WER	10.20	10.80	111.80
CER	8.83	9.61	68.32

Table 1: *WER and CER of Pretrained Models*

Finetuned Models		
Metrics	Conformer	TSConformer
WER	6.58	1.04e+02
ACC	9.35e-01	1.54e-01

Table 2: *WER and ACC of Finetuned Models*

Pretrained Vs Finetuned Models	
Model	MACs(G)
Conformer	438.37
TSConformer	699.93

Table 3: *Model Complexity(GMACs)*

to have more GMACs when compared to Conformer. This is against to theoretical understanding that usage of Linear Tailor Series Attention reduces complexity despite both having same number of parameters i.e. 386.4 Million. This needs further investigation.

Below are some of the possible causes for performance degradation of TSConformer:

- Vanilla Taylor Series Linear Attention doesn't have positional embedding encoded.
- Further tuning of hyper parameters may be needed for performance improvement.
- Pretraining followed by finetuning can also be carried out with full **Noisy LibriSpeech** for better results.

9. Conclusion

In this project, a **Noisy LibriSpeech** dataset is created from Librispeech. Then existing pretrained ASR models are evaluated utilizing WER and CER metrics. A modified version of Conformer with Taylor Series Linear Attention known as TSConformer is proposed and finetuned with **Noisy LibriSpeech** dataset. This is compared with the Conformer finetuned with same **Noisy LibriSpeech** dataset. It was observed that TSConformer showed performance degradation. Vanilla Conformer showed best results after finetuning.

9.1. Limitations and Future Work

- The proposed TSConformer didn't show improvement, infact it showed degraded performance
- Taylor Series Linear Attention doesn't seem to have positional embedding information encoded. We hypothesise that the may be a major factor for perforamnce degradation.
- Further hyper parameter tuning, full pretraining and finetuning along with architectural improvements can result in better model with comparable performance as that of finetuned conformer.
- TSConformer shows higher MACs (Multiply-Accumulate Operations) which is contrary to theoretical understanding. This needs further analysis and investigation.

9.2. Additional Details

- Code and execution details were listed in the README file
- Github Code : https://github.com/ksasi/asr
- Training, Evaluation logs, dataset, trained model and report were also provided in respective folders.
- Gradio demo of ASR is available in the demo folder of the repo

10. References

[1] Z. Ma, G. Yang, Y. Yang, Z. Gao, J. Wang, Z. Du, F. Yu, Q. Chen, S. Zheng, S. Zhang, and X. Chen, "An embarrassingly simple approach for llm with strong asr capacity," 2024.

[2] Y. Peng, S. Dalmia, I. Lane, and S. Watanabe, "Branchformer: Parallel MLP-attention architectures to capture local and global context for speech recognition and understanding," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 17 627–17 643. [Online]. Available: https://proceedings.mlr.press/v162/peng22a.html

[3] B. Zhang, B. Haddow, and R. Sennrich, "Revisiting end-to-end speech-to-text translation from scratch," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 26 193–26 205. [Online]. Available: https://proceedings.mlr.press/v162/zhang22i.html

[4] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 3915–3924. [Online]. Available: https://proceedings.mlr.press/v162/chiu22a.html

[5] K. Goel, A. Gu, C. Donahue, and C. Re, "It's raw! Audio generation with state-space models," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 7616–7633. [Online]. Available: https://proceedings.mlr.press/v162/goel22a.html

[6] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 28 492–28 518. [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html

[7] P.-H. Le, H. Gong, C. Wang, J. Pino, B. Lecouteux, and D. Schwab, "Pre-training for speech translation: Ctc meets optimal transport," 2023.

[8] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[9] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020.

[10] Y. Peng, S. Dalmia, I. Lane, and S. Watanabe, "Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding," in *International Conference on Machine Learning*. PMLR, 2022, pp. 17 627–17 643.

[11] K. Kim, F. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, "E-branchformer: Branchformer with enhanced merging for speech recognition," 2022.

[12] S. Arora, S. Eyuboglu, A. Timalsina, I. Johnson, M. Poli, J. Zou, A. Rudra, and C. Ré, "Zoology: Measuring and improving recall in efficient language models," 2023.

[13] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," 2020.