ECE 6254 Report: Analysis of SSL Models for Speaker Identification

Hemantha Krishna Bharadwaj, Venkata Sai Ritwik Kotra, Vignesh Srinivasa Naidu Prakash
Department of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, United States
hbharadwaj6@gatech.edu, vkotra3@gatech.edu, vprakash40@gatech.edu

***Abstract*—Self Supervised Learning (SSL) paradigm has been a boon for low resource learning problems, with pretrained models being able to adapt very well to downstream tasks with very low amount of data. SSL models learn the structure on a large corpus of unlabelled data, generally using contrastive or non-contrastive masked reconstruction as objective. We use two such speech SSL models for our Speaker ID task- HuBERT-Base and Wav2Vec2-Base. We closely follow the SUPERB benchmark, while making few custom changes to the models. An in-depth layer wise analysis of different regions of the pre-trained models is essential for interpretability and possible downsizing of these large SSL models. Pre-trained models currently available online are predominantly the final layer of the SSL model, while not providing clarity on how other layers perform or what length of audio is necessary. Very little detail is provided on the role of "attention masks" and whether the different padding configurations affect the results or not. We attempt to address a few of these issues in our work, while greatly outperforming the available online models with few small changes from our end, which will hopefully help the speech community.**

***Index Terms*—Speaker Identification, Self Supervised Learning, Pretrained Models, Layer-wise Analysis, Attention Masks, Audio Padding**

I. Introduction

Despite being a technology originating within the last century, speaker recognition, which mainly achieves its goals by analyzing speech inputs and extracting speakers' characteristics, has seen rapid development in the past few decades. As Deep Learning has become more prevalent, speaker recognition has benefited from breakthroughs in efficiently modeling data lying on a structured manifold.

Modern speaker recognition mainly addresses four topics: Speaker Verification (SV), Speaker Diarization (SD), Speaker Identification (SI), and robust speaker recognition. SV attempts to determine if two speech recordings are from the same speaker; SD divides recordings consisting of multiple speakers into segments by speaker; SI attempts to determine from which of the registered speakers a given utterance comes from. A flow chart of train and testing is shown in Figure 1.

Although speaker identification in particular (and speaker recognition in general) have benefited greatly from the use of supervised deep learning approaches [1] [2] [3], supervised learning has a bottleneck in terms of the requirement of large amounts of annotated data, which is both difficult and expensive to create. Self-supervised learning overcomes this obstacle since it requires unlabelled data for the majority of the training, which is abundant and readily accessible in the modern world.

Previous studies have evidenced this by achieving state-of-the-art performance using SSL approaches [4], [5], [6]. Wav2vec 2.0 [7] uses contrastive learning along with random masking to learn self supervised representations from unlabelled data. HuBERT [8] builds on the popular BERT [9] model by the addition of a hidden clustering unit that generates noisy labels. Both models have shown exceptional performance on downstream speech-based tasks and are widely used for pretraining speech based representations.

The current standard methodology for utilizing SSL representations is to extract embeddings from a pretrained state-of-the-art model and fine-tune them for a downstream task. This black-box approach relies on the assumption that the embeddings output by the final layer of these models is the required representation for providing optimal results. However, there is no substantive proof for this being the case in current literature. A layer-wise analysis of the representations obtained can provide insight into the type of information stored in embeddings at each layer and help understand what information is crucial for the particular downstream task. Authors in [10] did a layer wise analysis of Wav2vec 2.0 for ASR tasks and hypothesized that the final few layers play little role in adding helpful linguistic content and thus in improving performance.

The objective of this paper is three-fold. Firstly, we build on the layer-wise analysis approach by taking intermediate embeddings from the 12 layer versions of Wav2vec 2.0 (*wav2vec2 base*) and HuBERT (*HuBERT base*) and analyzing performance on the downstream task of speech identification. We calculate performance on embeddings extracted from the 6th, 9th and 12th layers respectively, and also analyze an ensemble approach. Secondly, we evaluate the importance of attention masks extensively used in SSL models, and virtually add silence to speech audio to find out if adding silence improves performance. Lastly, we perform these analyses for different lengths of training and testing audio to determine the trade-off between length requirement and model performance.

II. Methodology

In this project, we take the wav2vec2 and HuBERT base models which are state-of-the-art self-supervised speech rep-

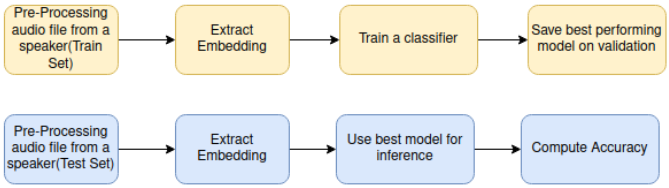

Fig. 1. Speaker Identification Process

resentation learning approaches, each of which have 12 layers of transformer encoder and 7 layers of CNN feature extractors(Fig.2). They have a downsample rate of 320 and are trained on 960 hours of Librispeech. These pretrained models act as the backbones for the task we set out to perform. We finetune the model on audio files obtained from the VoxCeleb1 dataset [11]. VoxCeleb is an audio-visual dataset consisting of short clips of human speech, extracted from interview videos uploaded to YouTube. VoxCeleb1 is a subset of VoxCeleb and contains over 100,000 utterances for 1,251 celebrities. For our project, we utilize the first 100 speakers(by id) of the VoxCeleb1 dataset. This corresponds to rouhly 1.5 hours of data. The splits are maintained as given by the VoxCeleb1 website. Total samples in an audio length range, for train dev and test split is shown in Table I.

TABLE I
NO.OF SAMPLES IN EACH AUDIO LENGTH BIN FOR TRAIN, TEST AND SPLIT

Audio Length	Train	Dev	Test
<2 secs	0	0	0
2-4 secs	131	7	10
4-6 secs	4863	281	242
6-8 secs	2387	135	131
8-10 secs	1388	72	55
>10 secs	2345	137	95

During the process of fine tuning, we consider layers 3,6,9 and 12 of the transformer encoder to extract embeddings. From the results in [10], we get a feel for what the individual layers contribute and we use that as a motivation to tap these layers to analyse them. Every third layer of a 12 layer transformer contains a particular information about the audio inputs. In order to extract the embeddings from the layers, we freeze all the layers except the layer we are on. The extracted embeddings then undergo mean pooling across the time axis, and these embeddings are just projected down using a linear layer. We use AdamW [12] optimizer which decouples the weight decay from the optimization step. with learning rate of $1e^{-4}$ for finetuning and Cross Entropy Loss for optimization. We train each model for 5 epochs. Validation accuracy is used as metric to determine the best model that will be used for evaluation. Our evaluation metric is also simple accuracy.

$$Accuracy = \frac{\sum_{n=1}^{NAudio} 1(Classifier(Audio) = SpeakerId)}{TotalNo.ofSpeakers} \quad (1)$$

For analysis, we compare the results obtained from the embeddings for 3s, 5s and 10s of audio. The dataset contains audio data with lengths varying from 3s to 35s. We try to adjust all the files to fit the time frame set above and therefore, truncate and if necessary, pad all audio files to these lengths. The zero padding is not removed even while taking the mean. This is done intentionally to see how the gaps in data,i.e., silence in audio contributes to the SI.

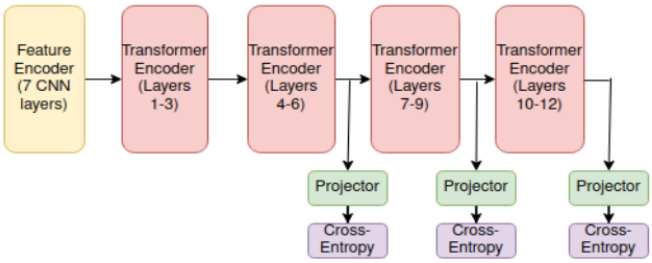

Fig. 2. Pre-trained model flow

Finally, we utilise all our findings and perform a comparative analysis with the released HuBERT-Base and wav2vec2-Base SI models on huggingface which use the SUPERB benchmark. The huggingface models pass attention mask to their models. It is necessary for the batch analysis of audios of varying lengths. We do not employ attention masking in our models because of the use of zero padding.

III. Results and Discussions

It is to be noted that in all our experiments, we refrain from using the attention mask. From the tables below, it can be seen that the HuBERT base performs better than wav2vec2 base almost as a general consensus for all layers. This is synonymous with the SUPERB benchmark results [13]. The accuracy is highest for the final layer, but it is interesting to note that the inner layers also provide decent predictions. Besides, it contains important information which can be used for SI. We build upon this idea to try and improve the accuracies by performing a simple mean of layerwise data. It results in the ensemble accuracies which are a significant improvement when compared to the individual layers. We observe from Table III that Hubert base trained with 10 seconds of audio, with embeddings extracted from 12^{th} layer has a whopping 95.25% accuracy. On performing score fusion we obtain a

Secondly, the length of the audio files play a role in the identification of the utterances. It can be seen that fixing the length to a higher value results in better accuracies. Audios with 10s(Table II) perform much better than its 5s(Table III) and 3s(Table IV) counterparts. This seems logical because having more amount of data to work with naturally helps in improving the prediction.

The above two results are generally true from the multiple runs. But we obtain interesting results which are seemingly contradictory from Fig.3. The graph is plotted for the average of the top 3 probabilities for each layer. It is also an analysis of the effect the layers have on different audio lengths. We aimed at trying to find out which range of audio lengths from

our originally taken dataset is identified the best from our models. The results showed that layer-6 performed marginally better than layer-12 for certain audio ranges. This is further evidence that the hidden layer has an abundance of untapped potential and can be utilized smartly for SI. Fig.3 also tells us that the smaller audios are more accurately predicted than the longer audios. This makes a lot of sense too because in the test dataset, having a longer audio means more uncharted territories which opens up room for error. Besides, this also shows the potential use of silence on the audio(zero padding).

The 2-4 secs of audio perform best and while that is because it has lesser unknowns, it also can be attributed to zero padding. As we make a 2s audio to 10s by appending zeroes, it is essentially adding silence to the audio. The silence part in the waveforms stores information of the speakers [14] which thereby assists the SSL models in it's identification.

Table V contain the results of the pretrained models which is the huggingface model which we use as a kickoff point. It is important to note that the huggingface model makes use of the attention mask. Attention mask does not mind a difference in audio lengths and so, it deletes any silence we end up adding thereby reducing the effect of zero padding and that is reflected in the accuracies. An important point to note here is that the accuracies obtained using the huggingface models are comparable to the results obtained using our custom model for the input audio of length 3s(Table IV). This offers a view at the vast improvement using our custom training procedures.

Table VI offers a view into a mismatch of audio lengths in training vs testing. We observe that passing in just 3 seconds of audio, to a model trained with 10 seconds in testing is enough to get really good accuracy. We hypothesize that the model learns to discriminate with small amounts of data much easier when trained on larger audio. This also means when deploying these models, the users don't need to speak for large amounts of time to achieve good performance, which is a huge bonus.

From Fig.4, we get the 5 worst probabilities assigned to labels based on the model prediction. We find that when the model gets it wrong, it gets it really wrong. We hypothesize that this is something that happens with 12^{th} layer, which struggles for specific speakers a lot.

To further probe into this effect, we do a speaker level analysis, by taking a majority vote out of all predictions for a speaker to compute the accuracy. This gives us a speaker level accuracy. As expected, for our best performing models the speaker level accuracy is 99%. More interestingly though, for Wav2Vec2-base and Hubert-base, the speaker level accuracy on majority voting for 6^{th} layer, with only 3 seconds audio is comparable to the Hubert-base 12^{th} layer result! This means that 6^{th} layer when making mistakes is making it uniformly across all speakers, while the 12^{th} layer model is faltering on a fixed set of speakers.

Since our results are promising, we run another aggressive analysis to see how bad the model performs in critical situations. We get the total speakers where we get all the audio files correct, and compare our models' performance. This allows us to see for how many speakers our model can get every audio file right. Very interestingly, except for the best performing model of Hubert-Base trained on 10 seconds of audio, the 6^{th} layer is the best for getting all instances of a speaker correct. This explains the superior performance layer 6 in top-3 classification as well as the superior ensemble performance. The graph is shown in Figure 5.

TABLE II
INPUT AUDIO OF LENGTH 10 SECONDS NO ATTENTION MASK CUSTOM

Layer(10s)	Hubert(%)	Wav2Vec2(%)
6	89.2	90.51
9	90.5	89.7
12	***95.25**	90.66
Ensemble	****97.31**	93.83

TABLE III
INPUT AUDIO OF LENGTH 5 SECONDS NO ATTENTION MASK CUSTOM

Layer(5s)	Hubert(%)	Wav2Vec2(%)
6	84.17	82.91
9	77.85	82.44
12	**92.25**	85.92
Ensemble	92.41	**92.72**

TABLE IV
INPUT AUDIO OF LENGTH 3 SECONDS NO ATTENTION MASK CUSTOM

Layer(3s)	Hubert(%)	Wav2Vec2(%)
6	67.79	68.67
9	73.89	58.5
12	75.95	69.93
Ensemble	**80.49**	**77.37**

TABLE V
HUGGINGFACE SUPERB SID 12TH LAYER, WITH ATTENTION MASK

Pre-trained-Seconds	Hubert(%)	Wav2Vec2(%)
3	63.0	62.2
5	81.8	**74.7**
10	**82.7**	69.1

IV. CONCLUSIONS

From our experiments above and the results obtained, we can see that not using attention mask and zero padding significantly improves accuracy. This suggests the presence of silence helps SID like in [14]. While individually, the 12^{th} layer performs really well, we find that on performing a simple score fusion of the output probabilities obtained from models trained on different layers, we hit massive gains in accuracy on all the models. This suggests we must take help of multiple layers embeddings when performing downstream tasks. Longer audio is always better than shorter audio for identifying speaker. However the not so large difference between Hubert12 for 10 secs and 5 secs means we can get away with a sweet spot.

TABLE VI
USING DIFFERENT LENGTHS FOR TRAIN(COL) AND EVAL(ROW) HUBERT 12

Prediction Matrix	3sec	5sec	10sec
3sec	75.95	85.75	86.87
5sec	81.17	92.25	93.83
10sec	80.5	91.6	**95.25**

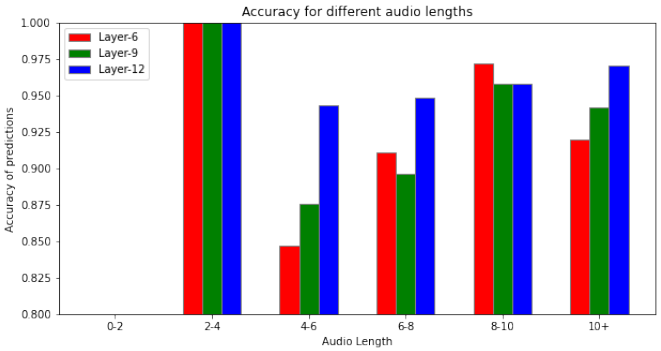

Fig. 3. Accuracy vs Audio Length Hubert-12-10secs(Top 3 accuracies)

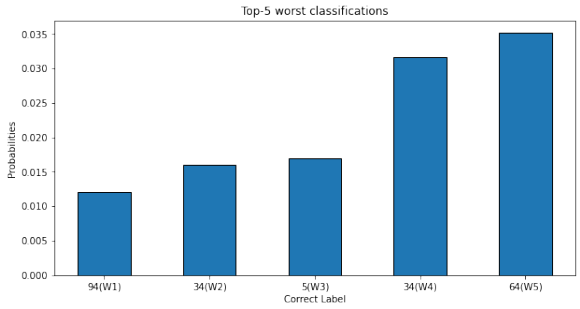

Fig. 4. Least confident misclassifications with Labels and Prediction Probability

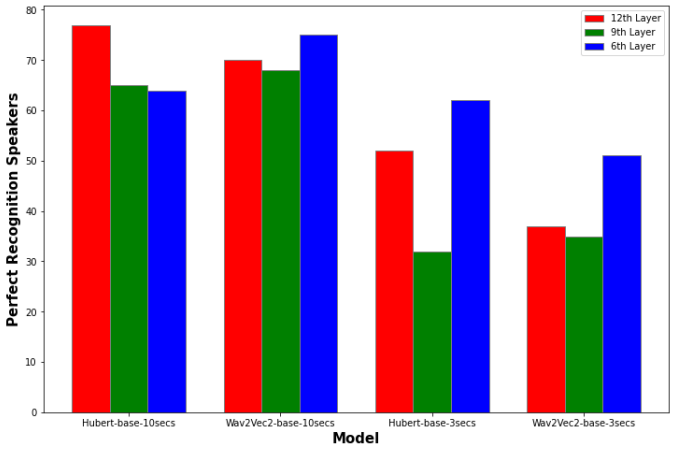

Fig. 5. No of Speakers perfectly recognised

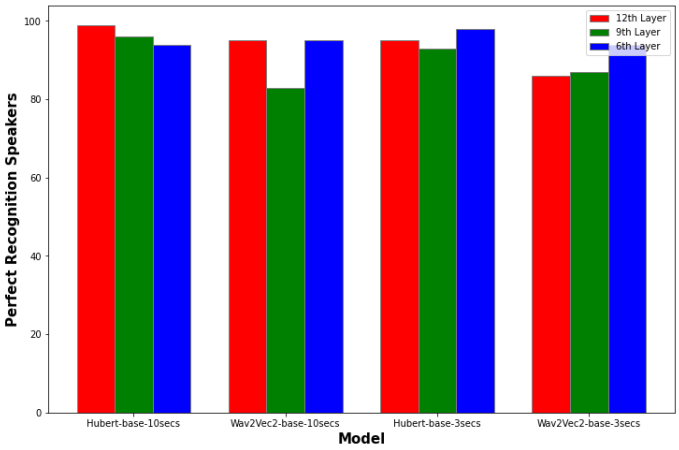

Fig. 6. Accuracy(in%) for Majority Voting

V. FUTURE WORK

There are multiple directions this work can be improved upon. Firstly, we want to solidify our claims by training on the 1251 speaker dataset and comparing the results. Secondly, we could use the fact that silence captures speaker characteristics to build speaker agnostic ASR systems. One obvious way is to pad your audio with silence, generate pre-trained representations and discard the embeddings regions corresponding to silence to feed to the ASR model. On the speaker id front itself, we want to explore what happens if we do an embedding fusion instead of score fusion for all these models. Not including attention mask works in our case since the base models were pre-trained without using the attention masks as well. It remains to be seen whether this would hold true when compared with models like HuBERT-Large and Wav2Vec2-Large, which were trained with attention masks. If it does not, we would automatically infer the importance of matching pre-training and fine-tuning configurations.

VI. TEAM MEMBERS' CONTRIBUTIONS

Ritwik is responsible for gathering the VoxCeleb1 dataset, coding, training and inference on the layerwise custom training on layers 3,6,9,12 on hubert-base and wav2vec2-base, attention mask analysis on hubert-base and wav2vec2-base models,train-test audio length mismatch analysis, majority voting analysis and writing corresponding analysis sections in report.

Hemantha is responsible for the dataset code for padding and truncating the audio lengths, running and identification using the custom trained wav2vec2 models for different audio lengths, ensemble analysis for hubert and wav2vec2 models, analysis of the worst predictions on hubert and wav2vec2 models and writing the report.

Vignesh is responsible for the inference analysis using the huggingface pretrained models for different times, identification using the custom trained hubert models for different times, top-3 accuracy extraction for layers 3,6,9,12, audio length wise SI comparative analysis and writing the report.

VII. Acknowledgement

This project was supervised by Professor Larry Heck. The team appreciates his guidance and the support he gave us. The team would also like to extend our gratitude to Georgia Institute of Technology for giving us the opportunity and facilities to undertake this project.

References

[1] Jee weon Jung, Hee-Soo Heo, Ju ho Kim, Hye jin Shim, and Ha-Jin Yu. RawNet: Advanced End-to-End Deep Neural Network Using Raw Waveforms for Text-Independent Speaker Verification. In *Proc. Interspeech 2019*, pages 1268–1272, 2019.

[2] Jee-weon Jung, Hee-Soo Heo, Ju-ho Kim, Hye-jin Shim, and Ha-Jin Yu. Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification, 2019.

[3] Joon Son Chung, Jaesung Huh, and Seongkyu Mun. Delving into VoxCeleb: environment invariant speaker recognition. *arXiv e-prints*, page arXiv:1910.11238, October 2019.

[4] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[5] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4037–4058, 2021.

[7] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

[8] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units, 2021.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[10] Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. Layer-wise analysis of a self-supervised speech representation model, 2021.

[11] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: A large-scale speaker identification dataset. In *Interspeech 2017*. ISCA, aug 2017.

[12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[13] Shu wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu hsien Huang, Wei-Cheng Tseng, Ko tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdel rahman Mohamed, and Hung yi Lee. Superb: Speech processing universal performance benchmark. In *Interspeech*, 2021.

[14] Chi-Luen Feng, Po-chun Hsu, and Hung-yi Lee. Silence is sweeter than speech: Self-supervised model using silence to store speaker information, 2022.