Final Report
Capstone Project, JPMC Team 1
Columbia University, Data Science Institute

Interpreting Transformer-Based Models for Speech-Related Tasks

Team Members:
Mridul Gupta
Siddhant Kumar
Siddhant Mahurkar
Angad Nandwani
Eubin Park

Mentors:
Akshat Gupta (JPMC)
Dr. Sining Chen

Table of Contents

1. Introduction

1.1 Motivation

The field of Natural Language Processing (NLP) has enjoyed great improvement over recent years owing to breakthroughs such as Word2Vec and BERT, significantly advancing the capabilities of machines to process text. Researchers have now turned their attention to processing *speech,* one of the most instinctual ways in which human beings interact with one another. Recently published transformer-based models such as Wav2Vec 2.0 and HuBERT have already begun leading these efforts.

However, neural networks such as Wav2Vec 2.0 and HuBERT are notorious for being *black boxes*; although researchers know their inputs and outputs, it is difficult to understand how these models have arrived at the results they produce. Many of the decisions made by the intermediary layers of a neural network can seem simply arbitrary. As a result, the lack of interpretability of these models have deterred some industries such as finance or medicine from fully adopting them into their applications. Insights into the inner workings of neural networks may also benefit researchers in their endeavor to improve such systems.

1.2 Problem Statement

Although their contributions to the field of speech-processing are promising, the Wav2Vec 2.0 and HuBERT models, like other neural networks, lack interpretability. This poses problems in terms of their potential utility and improvement. In order to gain insight into the inner workings of the Wav2Vec 2.0 and HuBERT networks, this project will conduct a *probing* investigation into the layers of these models in order to better understand how they are working under the hood.

1.3 Background

1.3.1 Transformer-Based Models for NLP

Wav2Vec 2.0 and HuBERT are both transformer-based neural networks developed for the purpose of processing speech. Transformers, a novel framework introduced in 2017, are the current state-of-the-art technique in the field of NLP. They are able to perform sequence-to-sequence tasks such as speech-processing with much better performance and training times.

a) Wav2Vec 2.0

Wav2Vec 2.0 [1] is a speech encoder model released by the Facebook AI team in 2020, improving on the earlier Wav2Vec model released in 2019 [2]. The network allows for semi-supervised training, requiring initial training on a large unlabeled dataset, then fine-tuning on a smaller labeled dataset. There are four important elements that comprise the structure of the Wav2Vec 2.0 model: the feature encoder, the quantization module, the context network, and the contrastive loss objective.

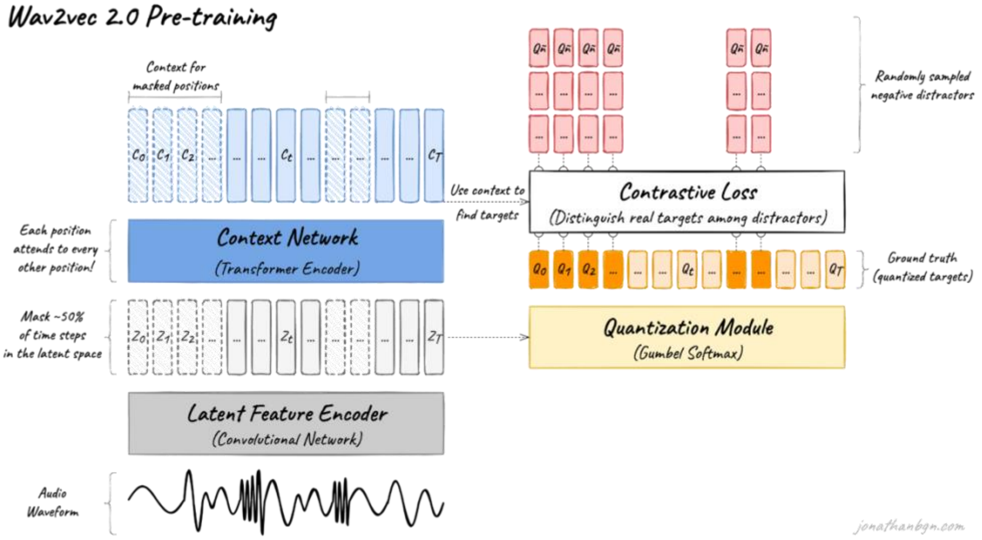

Wav2vec 2.0 Pre-training
Context for masked positions
c_0 c_1 c_2 ... ... c_t ... ... c_T
Each position attends to every other position!
Context Network
(Transformer Encoder)
Mask ~50% of time steps in the latent space
z_0 z_1 z_2 ... ... z_t ... ... z_T
Latent Feature Encoder
(Convolutional Network)
Audio Waveform
Use context to find targets
Contrastive Loss
(Distinguish real targets among distractors)
Randomly sampled negative distractors
$\tilde{q}$
q_0 q_1 q_2 ... ... q_t ... ... q_T
Ground truth (quantized targets)
Quantization Module
(Gumbel Softmax)
jonathanbgn.com

Figure 1. Wav2Vec 2.0 Model Architecture (Boigne 2021) [1]

b) HuBERT

HuBERT [3] is another transformer-based neural network that performs semi-supervised learning for speech-processing. It was released by the Facebook AI team in 2019. The main driver in the HuBERT training process is to learn ‘hidden units’ (hence, the ‘Hu’ in HuBERT) from speech data in order to give some structure to it. These hidden units can be regarded as analogous to words in written text. The overall process of HuBERT contains two phases: first, to extract the hidden units, and second, to train with the masked language modeling objective.

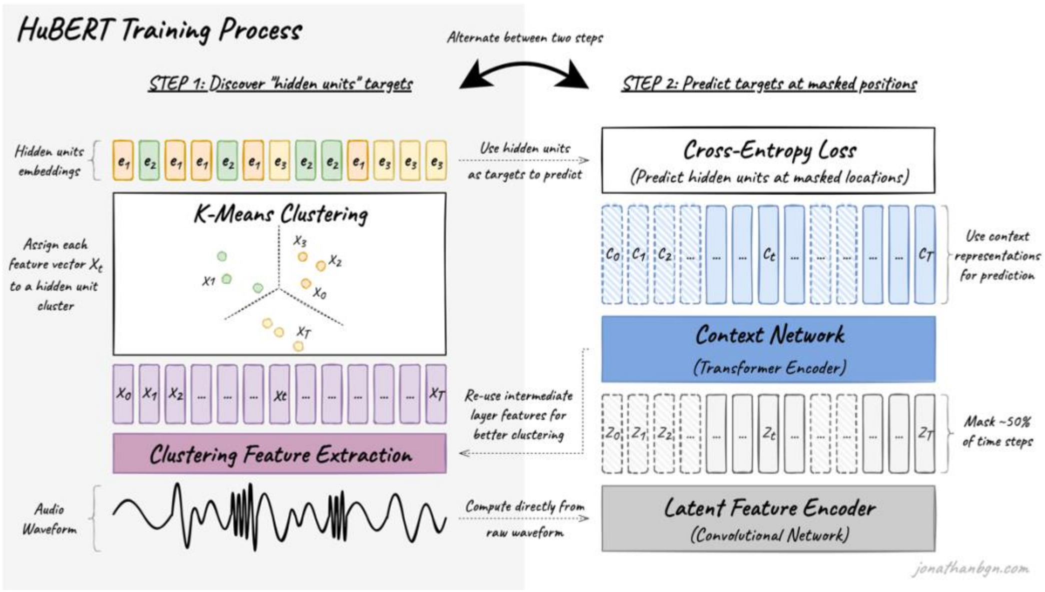

HuBERT Training Process
Alternate between two steps
STEP 1: Discover "hidden units" targets
STEP 2: Predict targets at masked positions
Hidden units embeddings
e_1 e_2 e_1 e_1 e_2 e_1 e_3 e_2 e_2 e_1 e_3 e_3 e_3
Use hidden units as targets to predict
Cross-Entropy Loss
(Predict hidden units at masked locations)
K-Means Clustering
Assign each feature vector X_t to a hidden unit cluster
X_1 X_2 X_3 X_0 X_T
X_0 X_1 X_2 ... ... Xt ... ... X_T
Clustering Feature Extraction
Re-use intermediate layer features for better clustering
c_0 c_1 c_2 ... ... c_t ... ... c_T
Use context representations for prediction
Context Network
(Transformer Encoder)
z_0 z_1 z_2 ... ... z_t ... ... z_T
Mask ~50% of time steps
Audio Waveform
Compute directly from raw waveform
Latent Feature Encoder
(Convolutional Network)
jonathanbgn.com

Figure 2. Framework of HuBERT Model (Boigne 2021) [3]

1.3.2 Speech-Based Tasks

Speech is one of the most instinctual ways in which human beings interact with one another, and being able to automate its procession would bring a lot of benefits. Within the field of speech-processing, there are three main speech-based tasks currently being pursued:

a) Speech Emotion Recognition

Speech Emotion Recognition (SER) is the fundamental task of detecting the emotion conveyed by some length of speech. Emotion in speech is extremely important in conveying key contextual information needed in social communications. The accurate detection of the speaker’s emotional state is required by speech processing systems to produce personalized responses or to make intelligent decisions.

The Wav2Vec 2.0 model has already been proven to be successful at Speech Emotion Recognition. Using a transfer learning method, researchers were able to achieve a 84.3% average recall on the RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset and a 67.2% average recall on the IEMOCAP (Interactive Emotional Dyadic Motion Capture Database) dataset [4]. Similarly, the HuBERT model was found to produce a 79.58% weighted accuracy using the IEMOCAP dataset [5].

b) Keyword Spotting

Keyword Spotting (KS) aims to detect when a single word is spoken within a set of ten or fewer target words, with as few false positives as possible from background noise or unrelated speech. One prominent application of Keyword Spotting in industry has been to detect keywords in phone conversations or other audio recordings in order to gauge customer satisfaction or product interest.

Keyword Spotting has also been tried previously with the Wav2Vec 2.0 model and the Google Speech Command Dataset V2. The model was able to achieve 97.8% accuracy, higher than the previously state-of-the-art DenseNet-BiLSTM model [6].

c) Language Identification

Language Identification (LID) is the task of determining which natural language is being spoken in some length of speech. One common use case for this task is within speech-translation applications, which must be able to detect what language is currently being spoken in order to translate it to the desired language.

Researchers were able to use the Wav2Vec 2.0 model to perform Language Identification with an Equal Error Rate (EER) of 3.47% on the AP17-LR dataset [7].

1.3.3 Probing Tasks in NLP

Probing tasks in NLP [8] are tasks that use the encoded representations, or *embeddings*, of some neural network to train another model to perform some classification task of interest. These embeddings are the intermediary values that are stored in the hidden layers of a neural network; one neural network can have multiple sets of embeddings depending on the number of layers it contains. The tasks chosen are usually tasks that perform some specific linguistic function. If the isolated embeddings perform well at the chosen probing task, it can be concluded that that linguistic function has been learned at the layer from which the embeddings were taken. In this way, by taking the embeddings of some neural network and performing probing tasks with them, one can learn the regions within a network that has encoded certain linguistic features.

In this project, we aim to perform probing tasks for the Wav2Vec 2.0 and HuBERT models. The probing tasks we have chosen are the three main speech-based tasks outlined in the previous section. In doing so, we hope to be able to identify the regions within these networks that best encode these three tasks and whether there is some structure or hierarchy in the way that these tasks are learned.

1.4 Literature Review

A previous paper, 'BERT Rediscovers the Classical NLP Pipeline' by Tenney *et al.* [9], demonstrates a similar probing investigation for the BERT model, a prominent pre-trained text encoder. The authors of this paper used eight labeling tasks from the traditional NLP pipeline: part-of-speech, constituents, dependencies, entities, semantic role labeling, coreference, semantic proto-roles, and relation classification, in order to identify which regions within the BERT model are able to encode what specific linguistic information and whether there is some hierarchy to this structure.

This investigation was designed with the observation that the initial layers of a language model usually store information of local syntactic structures, whereas the subsequent layers are left to encode more complex semantics. The results of this paper were able to confirm this observation by showing that the BERT model stores encodings of the above eight tasks in order of their traditional hierarchy from the traditional NLP pipeline.

1.5 Overall Approach

This project is divided into two phases. The first phase is to fine-tune existing state-of-the-art language models: Wav2Vec 2.0 and HuBERT. The second phase is to perform probing tasks on each layer of the two pre-trained models in order to interpret where certain speech-based information is being learned. (Refer to Fig. 3 below.)

To fine-tune the Wav2Vec 2.0 and HuBERT models for the three tasks: Speech Emotion Recognition, Keyword Spotting, Language Identification, we will leverage their pre-trained versions and fine-tuned them by unfreezing the final layers.

To form interpretations of how these models are encoding speech-based information, we will perform probing tasks on each layer of the language models. The models chosen to perform these tasks are Logistic Regression (LR), Support Vector Classifier (SVC), Random Forest (RF), and Long Short-Term Memory (LSTM). Moreover, we will compare class-wise accuracies across each layer for each of the three tasks.

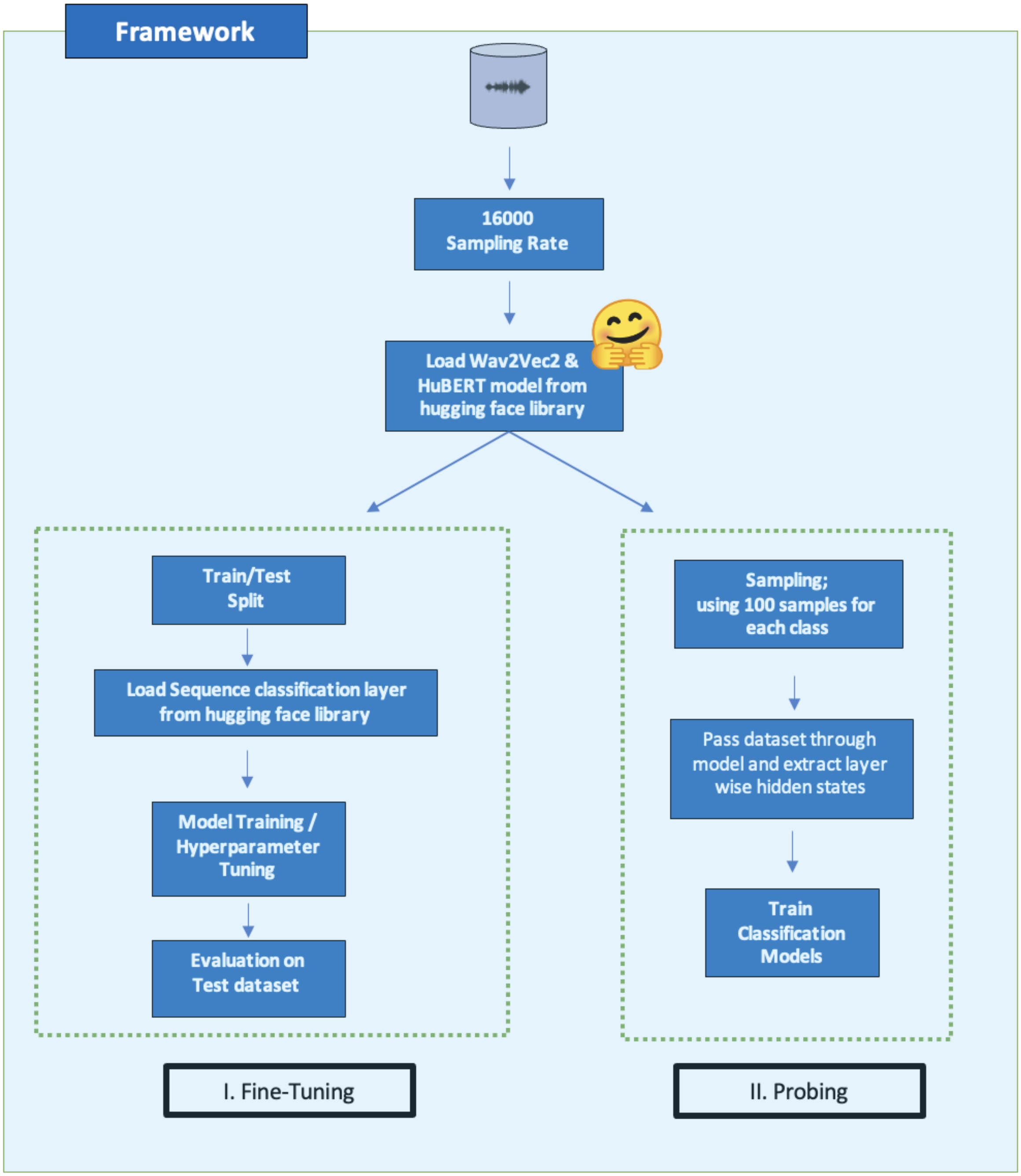


Figure 3. Overall Approach of the Project

2. Methods

2.1 Data

Task	Dataset	Description	Train/Val/Test Split
Speech Emotion Recognition	CREMA-D	• Consists of 7,442 clips • 6 emotion labels - happy, sad, anger, fear, disgust, and neutral	• All 7,442 clips with the 6 emotions have been taken • Train/Val/Test Split - 90/5/5
Keyword Spotting	Google Speech Commands	• Consists of 60,793 utterances. • Each utterance is 1 second long & contains a spoken word from a list of 12 keyword labels	• All 60,793 clips with the 12 keywords (Yes, No, Up, Down, Left, Right, On, Off, Stop, Go, _SIL_, _UNK_) have been taken • Train/Val/Test Split - 84/5/11
Language Identification	VoxLingua107	• Contains 6,628 hours of data • 107 languages represented (labels)	• 10 hours of data with 10 languages (Abkhazian, Belarusian, Bulgarian, Tibetan, Japanese, Korean, Russian, Serbian, Ukrainian, Mandarin Chinese) have been taken • Train/Val/Test Split - 90/5/5

Table 1. Brief summaries of all datasets utilized: CREMA-D, Google Speech Commands, VoxLingua107

2.2 Framework

2.2.1 Pre-Processing

Raw speech data are continuous signals that need to be discretized in order to be used within the context of machine learning. In order to prepare the audio files to be input into the Wav2Vec 2.0 and HuBERT models, the files were initially loaded using the ***librosa* library, transforming the .wav files into floating point arrays**. All files were resampled to a rate of 16 kHz in order to match the sampling rate of the files used to train the pre-trained encoders employed later on. The final step taken before the data was fed into the networks was to pass it through a **Wav2Vec feature extractor** to process the data into the format expected by the models.

2.2.2 Model Training

For model training, we used the pre-trained Wav2Vec 2.0 and HuBERT models from the Hugging Face library, each with an additional sequence classification head. This classification head consists of a SoftMax layer at the end of each of the Wav2Vec 2.0 and HuBERT models. This is the layer that performs the classification step of all three speech-based tasks.

To fine-tune the models for the required tasks, we adopted a transfer learning approach: we unfroze the last N layers and subsequently trained the model to achieve the greatest accuracy. N, the number of layers that were unfrozen depended on the classification task. For instance, the last 4 layers were unfrozen for the Speech Emotion Recognition task.

2.2.3 Probing

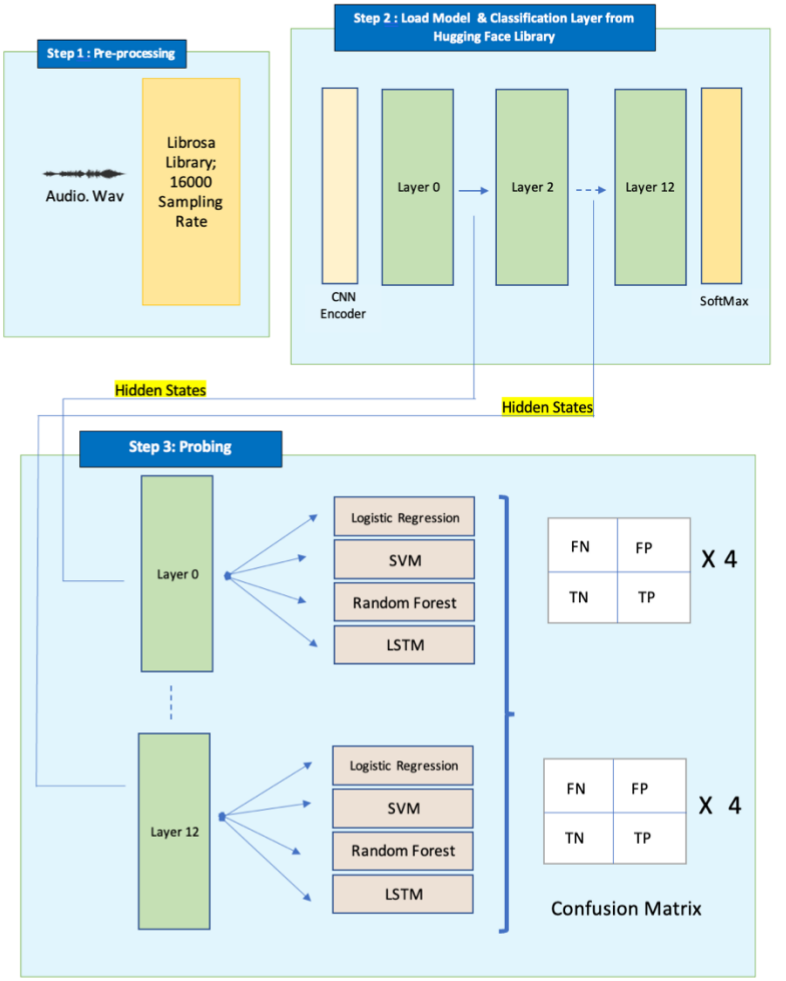


Figure 4. Framework for Model Training and Probing Tasks (Part 1)

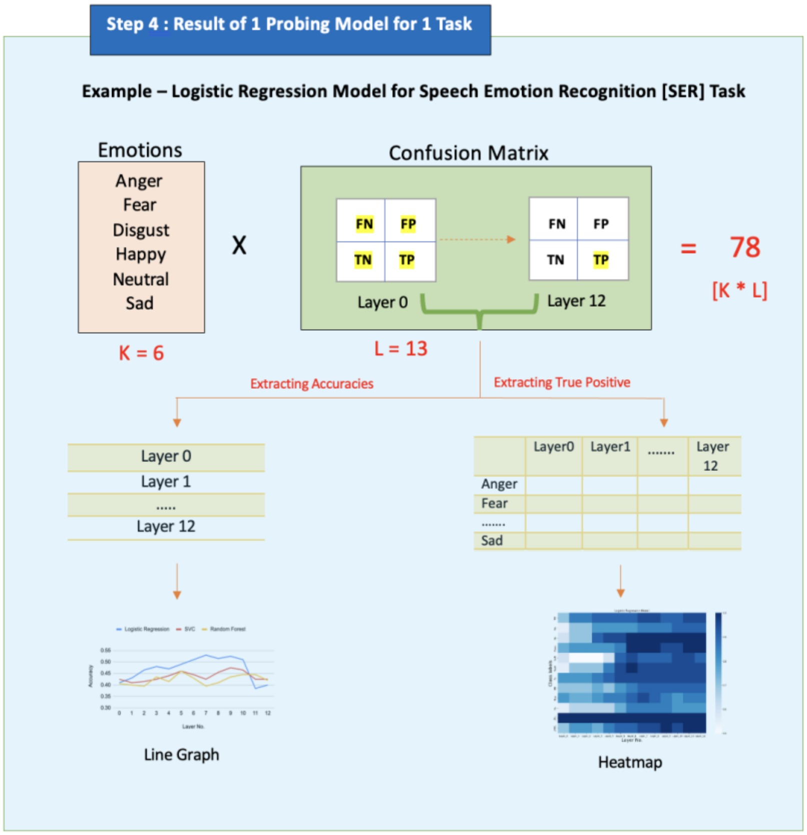
Once each of the Wav2Vec 2.0 and HuBERT models were trained for each of the three main tasks (2 * 3 = 6 models overall), the hidden state embeddings of layers 0-12 were sequentially extracted from the transformer encoders. In total, there were (2 * 3) * 13 = 78 sets of embeddings. These were used as inputs to the probing tasks performed later on.

Model	Task	Number of Hidden Layers	Total Number of Embedding Sets
Wav2Vec 2.0	SER	13	3 * 13 = 39
	KS	13	
	LID	13	
HuBERT	SER	13	3 * 13 = 39
	KS	13	
	LID	13	
Total Number of Embedding Sets: 39 + 39 = **78**			

Table 2. Total Number of Sets of Embeddings Extracted from the Models

The classification models chosen to perform the probing tasks were: Logistic Regression (LR), Support Vector Classifier (SVC), Random Forest (RF), and Long Short-Term Memory (LSTM). Each of the 78 sets of embeddings were then used as inputs to the pre-trained base models in order to perform their respective classification tasks. The accuracy of these models were measured in order to interpret at what layers certain tasks are being learned.

Because of time and resource (GPU & RAM) constraints, the scope of the probing portion of the project was limited to 100 samples for each label for each of the 3 speech tasks. This constraint was put in place to build the models faster; as a result, we were able to build many different models in order to compare their performances.



Step 4 : Result of 1 Probing Model for 1 Task

Example – Logistic Regression Model for Speech Emotion Recognition [SER] Task

Figure 5. Framework for Obtaining Results from a Single Probing Task (Part 2)

2.2.4 Walkthrough of Step 4: Results of Probing Model

In reference Fig. 5 above, the following explanation gives a walkthrough of how various plots and heatmaps are calculated for the probing tasks of Speech Emotion Recognition. The same process applies for the remaining Keyword Spotting and Language Identification. We start with passing a single audio sample into the model and extracting the hidden state representations (embeddings) from all 12 layers of the transformer as well as the initial input embedding - a vector representation of the input audio signal converted to digital format.

After successful extraction of hidden representations for all the samples in the dataset, we can build the actual probing models. We use each set of embeddings to individually train four probing models. This is repeated for each set of embeddings from each of the 13 layers of the 6 models. Overall, there are a total of 312 (4[unique models: LR, SVC, RF, LSTM] * 2[Wav2Vec & HuBERT] * 13[hidden states] * 3[classification tasks: SER, KS, LID]) probing models.

Considering a single Logistic Regression (LR) probing model, for Speech Emotion Recognition we have six unique emotions (k = 6): anger, fear, disgust, happy, neutral, and sad. We train the LR model on all the 13 input representations (L = 13) individually and calculate the multi-label confusion matrices for each emotion. Doing this process for each of the unique across all the layers we end up with 78 (k * L) confusion matrices.

We carry out two different analyses from the confusion matrices thus obtained. The first results are obtained by calculating the accuracy (TP + TN) / (TP + TN + FP + FN) for each model across the 13 different layers and plotting the respective line graphs to visualize the trends in accuracy. This analysis helps in getting a broader understanding of how the hidden representations from each of the layers of the transformer-based model is able to carry out the classification task at hand.

The second part of results are an in-depth analysis into how each layer of Wav2Vec 2.0 and HuBERT works to distinguish between the 6 unique emotions. To see how each label is classified across the layers we extract the true positives for a given emotion. True positives tell us, given n samples for a label, how many of them the layer was able to accurately classify. We generate heatmaps for these true positives through the layers to check the trend of how accurately a particular class is classified. Using a heatmap allows the reader to grasp the trend more quantitatively and clearly since the emotions that are inaccurately classified across most of the layers will be clearly visible.

2.3 Results

Our analysis broadly includes two tasks based on the two transformer-based speech models:

Task 1: Fine-tuning Wav2Vec 2.0 and HuBERT for the following two tasks:

a) Speech Emotion Recognition
b) Keyword Spotting
c) Language Identification

Task 2: Performing the following three probing tasks on Wav2Vec 2.0 and HuBERT hidden layer embeddings using Logistic Regression (LR), Support Vector Classifier (SVC), Random Forest (RF) and Long Short-Term Memory (LSTM) models:

d) Speech Emotion Recognition
e) Keyword Spotting
f) Language Identification

The results of our first task demonstrate our attempt at fine-tuning the existing state-of-the-art models for the tasks at hand. Fine-tuning a pre-trained model like Wav2Vec 2.0 or HuBERT for Speech Emotion Recognition, Keyword Spotting, and Language Identification can be a good way to get started with these tasks, as these models have already been trained on large amounts of data and have learned useful features for understanding natural language and speech.

The results of our second task showcase interpretations of the Wav2Vec 2.0 and HuBERT models in performing the three tasks through a process called *probing*. We use Logistic Regression (LR), Support Vector Classifier (SVC), Random Forest (RF) and Long Short-Term Memory (LSTM) models to train the probing classifiers. We use the hidden states of these transformer-based models to analyze how the two models perform across Speech Emotion Recognition, Keyword Spotting, and Language Identification.

Our results include the layer-wise and class-wise evaluation of these probing models. Layer-wise accuracies indicate how each of the 12 layers and the input embedding layer perform at the task at hand. On the other hand, the class-wise results indicate how each of the 12 layers perform across each of the label for a given task. For example, given the task of keyword spotting with class labels such as "yes", "no", "down", "upper", etc., we want to know how each of the 12 layers perform in classifying each of these classes accurately.

2.3.1 Fine-Tuning Results

Task	Model	Epochs	Accuracy
Speech Emotion Recognition	Wav2Vec2.0	10	78.00%
	HuBERT	5	77.14%
Keyword Spotting	Wav2Vec2.0	5	98.23%
	HuBERT	5	98.19%
Language Identification	Wav2Vec 2.0	5	86.44%
	HuBERT	5	85.35%

Table 3. Fine-tuning results of Speech Emotion Recognition and Keyword Spotting Tasks

We fine-tuned the Wav2Vec 2.0 and HuBERT models for keyword spotting and emotion recognition to obtain the accuracy results. The results show that the two models perform fairly well for these tasks.

2.3.2 Probing Results

2.3.2.1 Speech Emotion Recognition

a) Wav2Vec 2.0: Class-Wise Accuracy using Logistic Regression

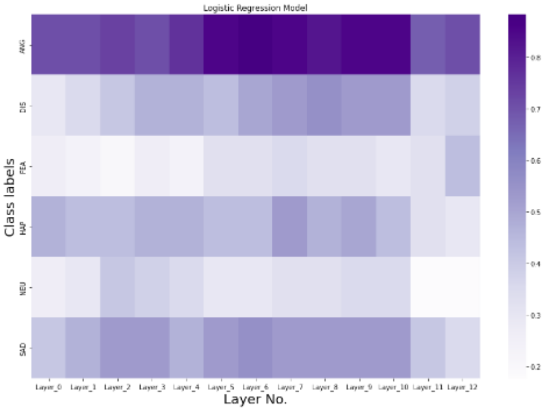


Figure 6. Layer-wise true positive score for each emotion using logistic regression

b) HuBERT: Class-Wise Accuracy using Logistic Regression

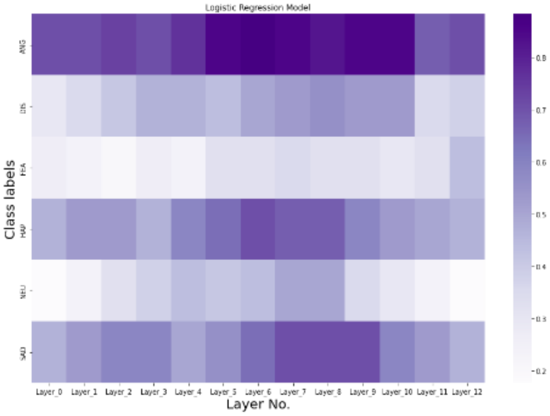
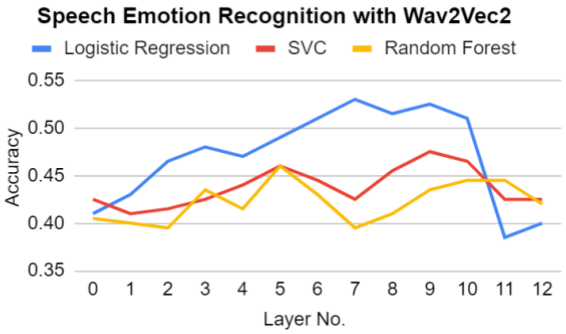


Figure 7. Layer-wise true positive score for each emotion using logistic regression

c) Per-Layer Probing Accuracies Comparison



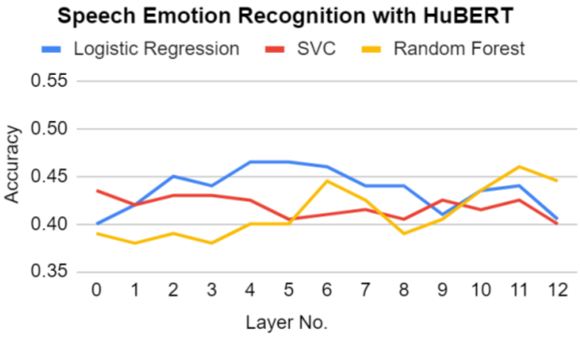


Figure 8. Wav2Vec 2.0 hidden representations layer-wise accuracy for different models

Figure 9.HuBERT hidden representations layer-wise accuracy for different models

From the heatmap for both Wav2Vec 2.0 and HuBERT we can clearly observe that the emotion angry “ANG” is the most accurately label classified across all the layers, while most other emotions are learned with fairly good accuracies. The emotions neutral “NEU” and fear “FEA” are the classes that are most inaccurately classified.

The layer-wise results follow the trends visible in the heatmaps. From the layers of the Wav2Vec 2.0 model we can observe that the middle layers perform better than the initial and last layers across all the three probing models. This finding is interesting since it can be assumed that layer 12 has all the representations from the initial embedding leading up to it and hence that its hidden representations should be able to perform the task most effectively. But the results contradict this and indicate that it is the middle layers (6 - 9) that are performing the most accurately. We can also observe that the LR model performs best among all the models for Wav2vec 2.0. Further, we can see that the same trend is visible for HuBERT where the middle layers perform the best. However, there is a difference in that although middle layers perform better, the increase in performance is not as significant as in Wav2Vec 2.0.

2.3.2.2 Keyword Spotting

a) Wav2Vec 2.0: Class-Wise Accuracy using Logistic Regression and LSTM

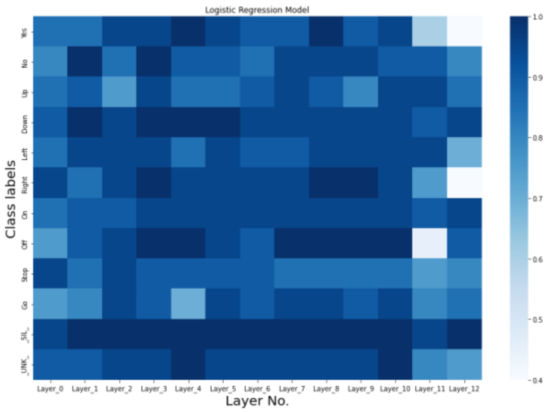


Figure 10. Layer-wise true positive score for each keyword using Logistic regression

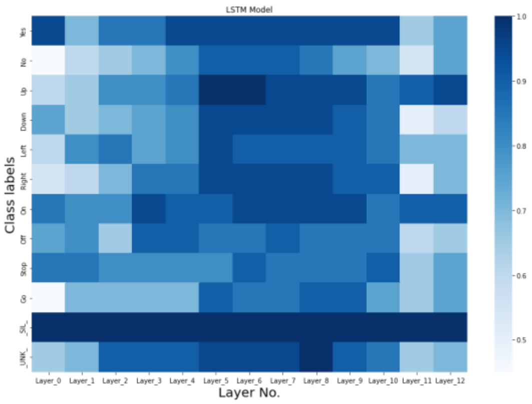


Figure 11. Layer-wise true positive score for each keyword using LSTM

b) HuBERT: Class-Wise Accuracy using Logistic Regression and LSTM

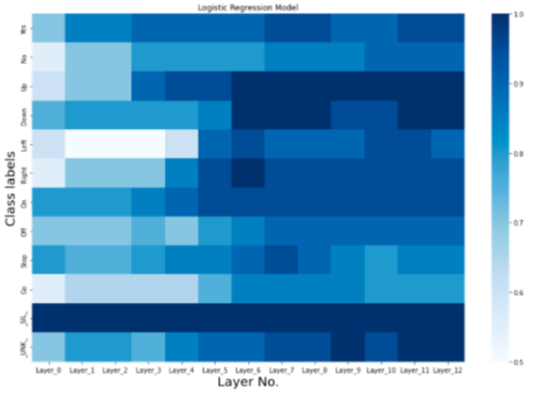


Figure 12. Layer-wise true positive score for each keyword using Logistic regression

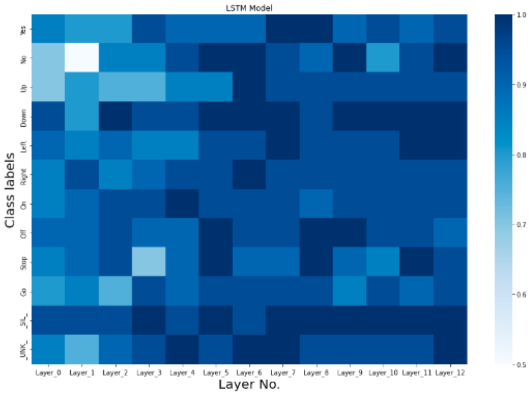


Figure 13. Layer-wise true positive score for each keyword using LSTM

c) Per-Layer Probing Accuracies Comparison

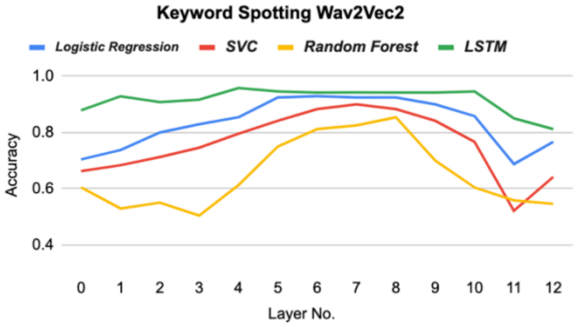


Figure 14. Wav2Vec 2.0 hidden representations layer-wise accuracy for different models

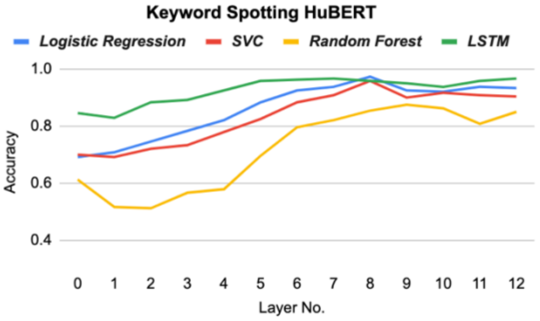


Figure 15. HuBERT hidden representations layer-wise accuracy for different models

From the above heatmap plots it can be inferred that the middle layers of the Wav2Vec 2.0 model are doing a better job in classifying keywords in comparison to the initial and final layers; this effect is most pronounced in the LSTM probe. In the HuBERT model, however, it seems that there is a continual increase in performance from the initial to final layers, with the final layers performing the best. Furthermore, keywords such as "_SIL_", "yes" and "up" are well classified by the HuBERT and Wav2Vec2 models, whereas, keywords like "left", "no" and "off" are not accurately classified by the models.

Lastly, from the line graphs for each of the probing models, we observe that the LSTM model is able to learn the hidden representation the best and that there is no drop in the final accuracies for probing on hidden representations of HuBERT as is present in the results of probing on the Wav2Vec 2.0 model.

2.3.2.3 Language Identification

a) Wav2Vec 2.0: Class-Wise Accuracy using Logistic Regression and LSTM

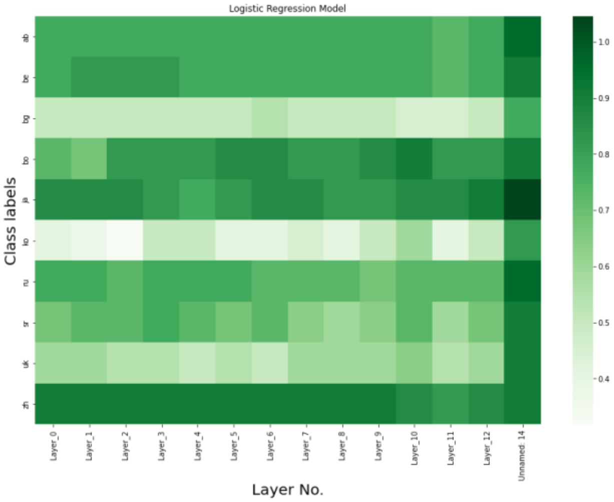


Figure 16. Layer-wise true positive score for each language using Logistic regression

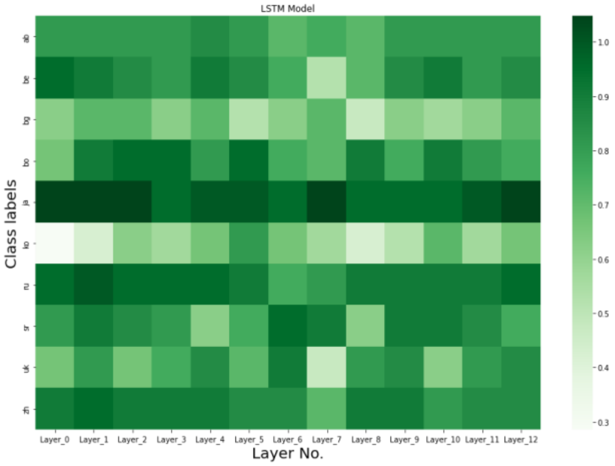


Figure 17. Layer-wise true positive score for each language using LSTM

b) HuBERT: Class-Wise Accuracy using Logistic Regression and LSTM

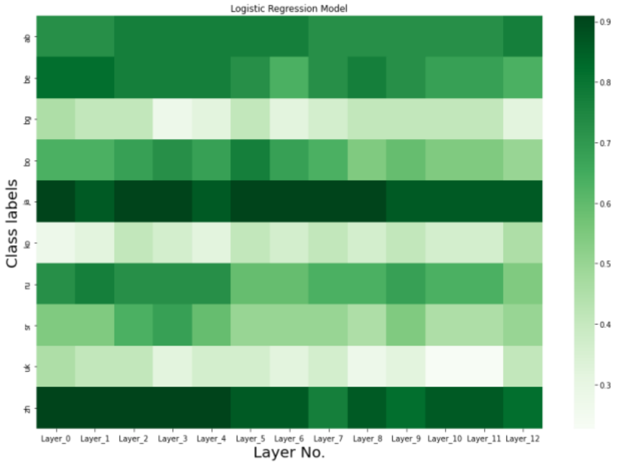


Figure 18. Layer-wise true positive score for each language using Logistic regression

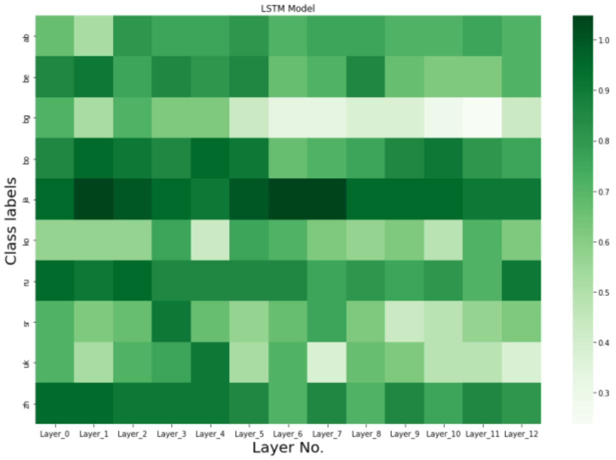


Figure 19. Layer-wise true positive score for each language using LSTM

c) Per-Layer Probing Accuracies Comparison

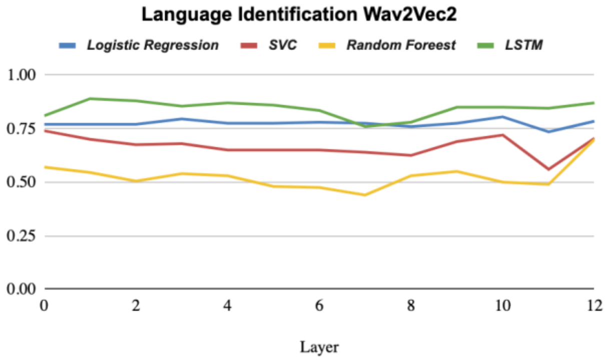


Figure 20. Wav2Vec 2.0 hidden representations layer-wise accuracy for different models

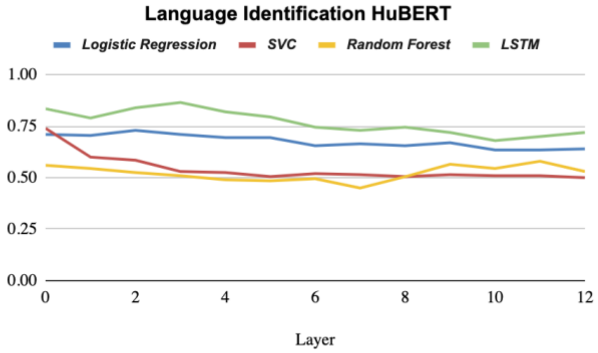


Figure 21. HuBERT hidden representations layer-wise accuracy for different models

From the heatmaps generated for language identification, there is no clear indication on which layers of Wav2Vec 2.0 and HuBERT are performing better than others based on the patterns in heatmaps. However, we do observe that certain languages are better classified than others. For example, languages like Japanese and Chinese are well classified by our language models however, our models are not doing a good job in classifying languages like Bulgarian, Korean and Ukrainian.

From the line graphs for each of the probing models, we observe that the LSTM model is able to learn the hidden representation better in comparison to other probing models. Lastly, we observe similar probing results on both Wav2Vec 2.0 and HuBERT for Language identification.

3. Discussion

3.1 Conclusion

On fine-tuning the Wav2Vec 2.0 and HuBERT models for the three main speech-based tasks: Speech Emotion Recognition, Keyword Spotting, and Language Identification, we observed the results to be very similar to the state-of-the-art results, which demonstrated their ability to perform well across different tasks.

For the probing portion, we observed that the middle layers of Wav2Vec 2.0 and HuBERT generally perform better than the initial and end layers, especially for the Speech Emotion Recognition and Keyword Spotting tasks. Language Identification, on the other hand, seems to be learned equally throughout all the layers. This is logical; the first two tasks – Speech Emotion Recognition and Keyword Spotting are more localized tasks that require more specific elements, whereas Language Identification is a more overarching theme. Furthermore, among all probing models, Logistic Regression and LSTM provided superior results. Lastly, we observed that the distribution of accuracies is not constant across all classes. That is, for each task, not all classes were learned accurately by the model, meaning the model learns some classes better than others.

3.2 Future Work

1. Leverage Wav2Vec 2.0 and HuBERT models for other tasks such as Speaker Verification.
2. Broaden probing scope by collecting more samples per class.
3. For each task, probe the models with multiple datasets to draw better conclusions.
4. Investigate drop in accuracy for penultimate layers of Wav2Vec2.0 model for Speech Emotion Recognition and Keyword Spotting.
5. A deeper analysis into why some classes for each tasks were not classified well by the models and identify which part of the machine learning pipeline (data collection, model tuning) can be improved to achieve better accuracies for such classes.
6. Adopting this strategy for other domains to improve model interpretability.

3.3 Ethical Considerations

There are no ethical concerns needed to be discussed with regard to this project. All datasets used were ethically sourced and publicly available.

4. References:

[1] Boigne, Jonathan. “An Illustrated Tour of wav2vec 2.0.” *Jonathan Bgn*, 30 Sept. 2021, https://jonathanbgn.com/2021/09/30/illustrated-wav2vec-2.html.

[2] Boigne, Jonathan. “The Illustrated wav2vec.” *Jonathan Bgn*, 29 June 2021, https://jonathanbgn.com/2021/06/29/illustrated-wav2vec.html.

[3] Boigne, Jonathan. “Hubert: How to Apply Bert to Speech, Visually Explained.” *Jonathan Bgn*, 30 Oct. 2021, https://jonathanbgn.com/2021/10/30/hubert-visually-explained.html.

[4] Chen, Li-Wei, and Alexander Rudnicky. "Exploring Wav2vec 2.0 fine-tuning for improved speech emotion recognition." *arXiv preprint arXiv:2110.06309* (2021)

[5] Wang, Yingzhi, Abdelmoumene Boumadane, and Abdelwahab Heba. "A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding." *arXiv preprint arXiv:2111.02735* (2021).

[6] Seo, Deokjin, Heung-Seon Oh, and Yuchul Jung. "Wav2kws: Transfer learning from speech representations for keyword spotting." *IEEE Access* 9 (2021): 80682-80691.

[7] Fan, Zhiyun, Meng Li, Shiyu Zhou, and Bo Xu. "Exploring wav2vec 2.0 on speaker verification and language identification." *arXiv preprint arXiv:2012.06185* (2020).

[8] Morger, Felix. “What Are Probing Tasks in NLP?” *What Are Probing Tasks in NLP? – Språkbanksbloggen*, 16 Nov. 2019, https://spraakbanken.gu.se/blogg/index.php/2019/11/16/what-are-probing-tasks-in-nlp/#:~:text=Probing%20tasks%2C%20which%20have%20also,(probing)%20task%20of%20interest.

[9] Tenney, Ian, et al. “Bert Rediscovers the Classical NLP Pipeline.” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, https://doi.org/10.18653/v1/p19-1452.

Member Contributions
Mridul Gupta – Methods
Siddhant Kumar – Results, Discussion
Siddhant Mahurkar – Methods, Results
Angad Nandwani – Methods, Discussion
Eubin Park – Introduction, Methods, Compilation of Report