Final Report

Robust Generalization in Automatic Speech Recognition: wav2vec 2.0 and Whisper

Sivan Ding (sd3507), Alexandria Guo (ag4475), Anh-Vu Nguyen (an3078), Antonin Vidon (av3023), Julia Wang (jw4169), and Maxwell Zhou (mtz2110)

M.S. in Data Science, Columbia University

December 18, 2022

1 Introduction

1.1 Background

The capstone project aims to test and improve the performance of open-source ASR (Automatic Speech Recognition) models: Facebook AI's wav2vec 2.0, wav2vec 2.0 with a 4-gram language model, and OpenAI's Whisper. We obtain model performance baselines, measured by word error rate (WER), by transcribing audio data from the LibriSpeech corpus, and then test the models' robustness on noisy and downsampled versions of the data. We also perform tests to obtain transcription accuracy on English audio recorded by speakers with different accents and also on recordings in different languages (using wav2vec 2.0 XLSR). We build upon these models by fine-tuning them on audio data from the Fleurs dataset. The resulting models from our custom fine-tuning pipelines significantly improve WER performance over their pretrained counterparts: fine-tuned wav2vec 2.0 XLS-R shows an average of 44.43% (at best, 74.2%) relative reduction in WER and fine-tuned Whisper achieves an average of 36.28% (at best, 66.6%) reduction in WER compared to their base models.

All our codes are open-sourced and accessible through this link: github.com/anhvung/Capstone-Audio-Transcription

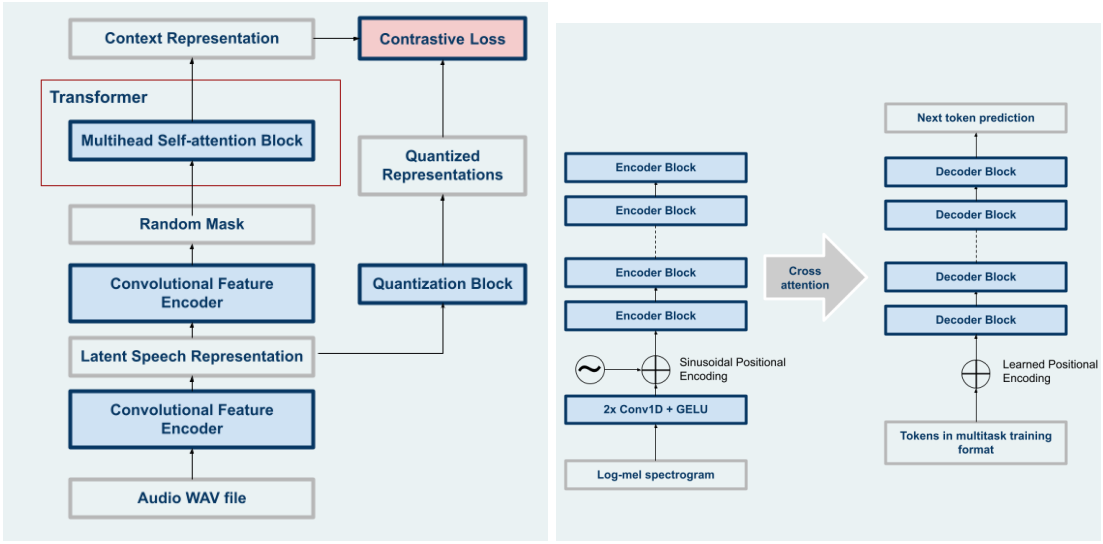
1.1.1 Model comparison and specifications

The wav2vec model was suggested by our mentors. It is an open-source model shared by Meta research (formerly Facebook) [1]. We will test it against the new Whisper model from OpenAI [2].

wav2vec 2.0 + language model: Often in ASR models, there is a language model included. The purpose of the language model, given a sequence of words, is to compute the probability of the next word. For example, this model should be able to identify that "recognize speech," is much more likely than "wreck a nice beach," and can help decide between two similar-sounding interpretations of the audio, thus improving the accuracy of the ASR model. Given this model is used during inference, the model typically needs to be small and lightweight, so a popular choice is a 4-gram model, which from the wav2vec 2.0 paper (see Appendix A.1) is shown to boost performance from the base wav2vec 2.0 model.

During the pretraining step of wav2vec 2.0, a portion of the audio representation is masked, and the model tries to predict the hidden part. This allows the model to be pretrained on a large amount of unlabeled data, and to learn rich representations of the audio features.

Whisper: Whisper is an ASR model trained on multilingual speech data from various sources on the web in a multitask fashion. It is able to detect and transcribe audio data, and can also translate non-English audio data by providing English transcription. Whisper is based on a transformer architecture. Because it is newer and trained on a larger amount of data, we expect to measure error rate improvements over the wav2vec model.



(a) (b)

Figure 1: Model architectures for wav2vec 2.0 (pretraining step) (a) and Whisper (b).

1.2 Literature Review

Before the use of attention modules that are part of Whisper and wav2vec 2.0, ASR models like Li-GRU [3] or CNN + TD-filterbanks [4] were often based on fully convolutional neural networks or encoder-decoder architecture resulting in lower transcription accuracy compared to their newer transformer-based counterparts.

wav2vec: wav2vec [5] is the predecessor of wav2vec 2.0 [1]. While they both use convolutional layers to compute a general representation of the raw audio data, their approach to the pretraining step differs. During the pretraining step, wav2vec tries to distinguish true future audio samples from negatives while wav2vec 2.0 uses a mask layer to hide a portion of the audio representation.

wav2vec XLS-R: wav2vec XLS-R [6] is based on the wav2vec 2.0 architecture [1] but enables multilingual capabilities by pretraining a single model on multiple languages. Results show that cross-lingual pretraining improves the model performances over their monolingual counterparts. This is because the latent speech representations are shared across languages, with increased similarities for related languages.

HuBERT: Another current competing self-supervised approach to learning speech representations is HuBERT [7], an expansion upon BERT-like algorithms used in natural language processing. If BERT uses bidirectionally trained transformers, HuBERT utilizes and builds upon this idea in the context of speech data, where it also learns short, "hidden units" (Hu) segments of audio that could potentially correlate to words or tokens of text. HuBERT shares its convolutional network plus transformer architecture with wav2vec, but after learning hidden units from clustering of short audio segments extracted from the data, it uses a different training process based on BERT's masked language modeling with cross-entropy loss.

1.3 Problem Statement

We plan to subject state-of-the-art ASR models to performance transcription tests on challenging audio samples that include common recording hazards: background noise, poor quality, multiple accents of speakers, etc. This way we will be able to assess the robustness of these models in settings close to real-life applications. We are also interested in the ability of these models to build upon their current knowledge when given extra fine-tuning data. Hence, we will try to enhance model performance by using various open-source audio datasets and developing custom fine-tuning pipelines.

2 Methods

2.1 Data and EDA

2.1.1 Dataset Descriptions

We used the following datasets in data exploration, baseline, and use-case testing, as well as training and fine-tuning of our models. All of the datasets are open-source datasets available online to the public.

LibriSpeech: The audio dataset that we are using primarily for baseline exploration and testing is the LibriSpeech dataset. The LibriSpeech dataset is a corpus of around 1000 hours of reading English speech that is derived from audiobooks from the LibriVox project [8]. The audio files are 16 kHz, and the dataset is split into clean and other development, training, and testing sets of different sizes. The clean and other sets roughly contain the same number of audio files and are divided by speaker, according to the WER of the WSJ model's transcripts of that speaker's speech. The lower WER speakers are in the clean dataset and the higher WER speakers are in the other dataset. The corresponding text transcript for each audio file is also provided. For exploratory data analysis, we mainly leveraged the Librosa library in Python for transforming and visualizing our audio files in our analysis.

FLEURS: We use the Fleurs dataset for training and fine-tuning our models. Fleurs stands for Few-shot Learning Evaluation of Universal Representations of Speech and is an n-way parallel speech dataset that contains audio data from 102 languages, for a total of around 1400 hrs of speech [9]. The audio samples are the recorded speech of native speakers of various languages, reading scripts of 3001 sentences from English Wikipedia (from the FLoRes-101 dataset) that have been human translated into their respective languages; approximately three different speakers recording a reading of each sentence, prior to a quality control check. Samples are also recorded at 16kHz and at lengths of at most 30 seconds, with a 30-70 or better split on speaker gender maintained across languages.

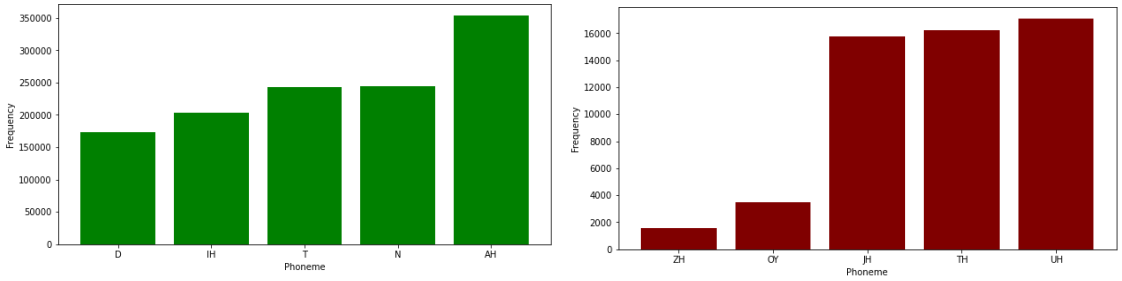
Speech Accent Archive: Another open-source dataset that we use for testing is the Speech Accent Archive [10]. This dataset audio files and metadata for speakers from a variety of language backgrounds, both native and non-native English speakers. Each speaker reads the same passage: "Please call Stella. Ask her to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station." As a result, the dataset contains thousands of audio files of speakers with different accents reading this same exact transcription.

Common Voice: The Mozilla Common Voice dataset [11] contains recorded speech that was utilized in pretrained wav2vec 2.0 XLS-R for many languages; further, fine-turning was also tested for Whisper models on this data. As of December 14th, 2022, with the latest addition of the 12.0 corpus, the Common Voice dataset contains over 17000 validated hours of audio, in over 104 languages. Audio samples are crowd-sourced, and validated using crowd-sourcing as well—samples with notable background noise or multiple speakers are, according to policy, rejected. With scripts sourced from a combination of Wikipedia excerpts and community submitted and approved examples, 48kHz audio samples of these scripts can be downloaded.

2.1.2 Language Analysis EDA

The LibriSpeech dataset contains some metadata with regard to the speakers and source for the audiobook clips, including reader gender. This was found to be approximately equally balanced for male and female speakers across all sub-datasets of LibriSpeech; for our further investigations on phonemes, we focus on a larger split of the data in 'train-clean-100.'

From linguistic theory, we obtain the concept of levels of language; in the context of ASR, we are particularly interested in the addition of phonetics to other levels more relevant to natural language processing on text-only samples. Thus, we examined the transcripts of the clips in 'train-clean' and used the CMU Pronouncing Dictionary (CMUdict), found in the NLTK Python library and based on the ARPAbet, to convert words into their component phonemes. Approximately 98% of the words in the corpus were included in CMUdict and successfully converted to their most standard pronunciation. Disregarding distinctions in phonemes based on emphasis, we then find the most and least common phonemes present in LibriSpeech 'train-clean' in Figure 2:



(a) (b)

Figure 2: Top five most (a) and least (b) represented phonemes in the LibriSpeech 'train-clean' 100 hours subset.

We can observe some similarities in this curated dataset's frequency counts for phonemes to that of conversational American English [12] — 'AH' (or 'ə' as in comm**a**, in commonly spoken words such as "the" or "a" or "of") is the most common phoneme, and in general the top four matches that of Mines et al.'s findings exactly. Some variations in phoneme definition make it more difficult to compare the less common phonemes, but overall LibriSpeech shares some phonetic likenesses to colloquial English, despite being composed of select audiobook clips. Although our study focuses on word-based metrics, phoneme-based metrics could also be utilized to identify if certain sonic units of language are better or worse understood by our algorithms.

2.2 Analytic Methods

Each leveraging large-scale training data, Whisper and wav2vec 2.0 both take in raw audio and map it to high-quality pretrained representations that lead to state-of-the-art speech recognition results. However, these models have different structures that contribute to their different levels of performance on downstream tasks. For example, wav2vec 2.0 consists of CNN and transformer layers, while Whisper uses a transformer encoder-decoder architecture. Moreover, wav2vec 2.0 needs a performant decoder and a fine-tuning stage in order to provide useful outputs for speech recognition, whereas Whisper aims to simplify this process and provide an end-to-end human-level transfer performance without fine-tuning. These two models are also trained in different fashions, as Whisper is trained with multilingual and multi-task weak supervision while wav2vec 2.0 is trained with self-supervision.

Given their different training techniques and architectures, we want to better understand how different components (i.e., acoustic and language modules) entangle with each other, as well as how each of them helps with robustness and generalization ability under various contexts, through a holistic evaluation across models based on wav2vec 2.0 and Whisper frameworks. Inspired by the human-level performance of Whisper on downstream multilingual speech recognition tasks, we aim to test the robustness and generalizability of their speech representations through a series of challenging scenarios in ASR: a) degraded audio quality with low sample rate and additive noise; b) English language with multilingual accents; c) four languages linguistically distant from Indo-European languages: Hebrew, Telugu, Chinese, and Korean.

Similarly to the method outlined in Whisper (see Appendix A.3) in the first part of their evaluation, we perform ASR with LibriSpeech. Since the LibriSpeech dataset is included in the training data in supervised wav2vec 2.0 but not in Whisper, this evaluation is in favor of wav2vec 2.0 as a held-out but in-distribution test. For Whisper, this is a more challenging out-of-distribution zero-transfer test. Despite these differences, we aim to compare the results of quality-downgraded tests with the originally reported results of each model to demonstrate their robustness and sensitivity to audio quality.

The second part of the evaluation includes English language ASR with three types of accents from the Speech Accent Archive dataset: Spanish, Arabic, and Mandarin. In contrast to the first part, this task focuses on how the models generalize to different variants of English, thus using only the clean speech data with the original quality.

The final part of the evaluation aims to perform out-of-distribution ASR for all the models. We first evaluate the zero-shot transfer performance of models on the test set of the multilingual Fleurs dataset, then fine-tune the models with the training set from Fleurs to evaluate their in-distribution performance with the same amount of additional supervision. Besides the comparison

between models on direct metrics, we also perform confidence score analysis to demonstrate the distribution and characteristics of model output.

2.3 Data Preprocessing

In this section, we introduce the preprocessing methods during the three stages of pretraining, evaluation, and fine-tuning.

Pretraining & Fine-tuning: Since we build our work upon the pretrained models of wav2vec2 2.0 and Whisper, the preprocessing workflows in both original pretraining and fine-tuning stages are similar.

In the wav2vec 2.0 base model, the feature extractor takes in 960 hours of LibriSpeech (LS-960) and 53.2k hours of LibriVox (LV-60k). The raw waveform of each example is cropped to 250k samples (approximately 15.6 seconds for 16 kHz sample rate) to create batches, which are normalized to mean zero and unit variance as input to the encoder [1].

In Whisper, raw audio input is resampled to 16 kHz, broken into 30-second segments (either truncated or zero-padded), then transformed to an 80-channel log magnitude Mel spectrogram with a 25ms window size and a 10ms stride size. Before feeding features into the encoder, the input is globally scaled to -1 and 1 with approximately zero mean across the pretraining dataset. Unlike the unlabelled input in self-supervised wav2vec 2.0, Whisper takes pairs of segmented audio features with the subset of the transcript that occurs within each time segment regardless of the inclusion of speech event [2].

Evaluation: As part of our initial exploration of model performance, we created a handful of helper functions to help downgrade the quality of audio data for both models to conduct challenging ASR tasks. These audio preprocessing functions include adding background noise, overlaying two audio samples, and downsampling the audio data. All functions return audio signals with a consistent sample rate of 16 kHz.

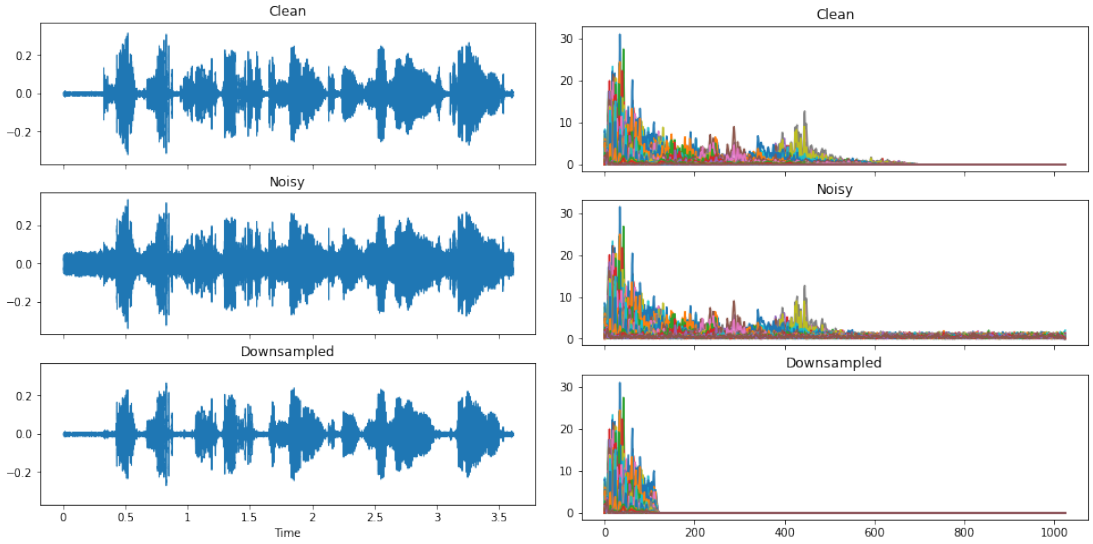
- To reproduce the evaluation of robustness to noise (see Appendix A.3), we build the `add_noise` function to add three different noise types: white, pink, and brown to raw audio input. Additionally, to see each model's sensitivity to noise, we evaluate the models on different levels of additive noise intensity to demonstrate the relationship between model performance and signal-to-noise ratio (SNR) from 1% to 6%.

- The `add_signals` function takes two audio samples, aligns their sample rate, and overlays them. Additionally, the user is able to control the decibels of the sample added. This function helps us to reproduce a similar experiment to the pub noise study shown in A.3.

- Besides reproducing the analysis of robustness to additive noise, we evaluate the robustness of the two models to degraded audio quality. The purpose of the `down_sample` function is to remove different levels of fidelity in the audio by sampling it to a lower sample rate (e.g., 8 kHz as the sample rate of phone calls), and then resampling back up to 16 kHz. We choose the sample rate between 500 Hz to 16 kHz.

Visualization and Analysis: After applying the two preprocessing techniques mentioned above to the clean audio files in LibriSpeech, either adding noise or downsampling the audio file), we randomly selected an audio file from the LibriSpeech test-clean dataset (61-70968-0001) and show four types of visualization of the three audio files side-by-side including time domain waveforms, frequency domain waveforms, spectrograms, and Mel spectrograms.

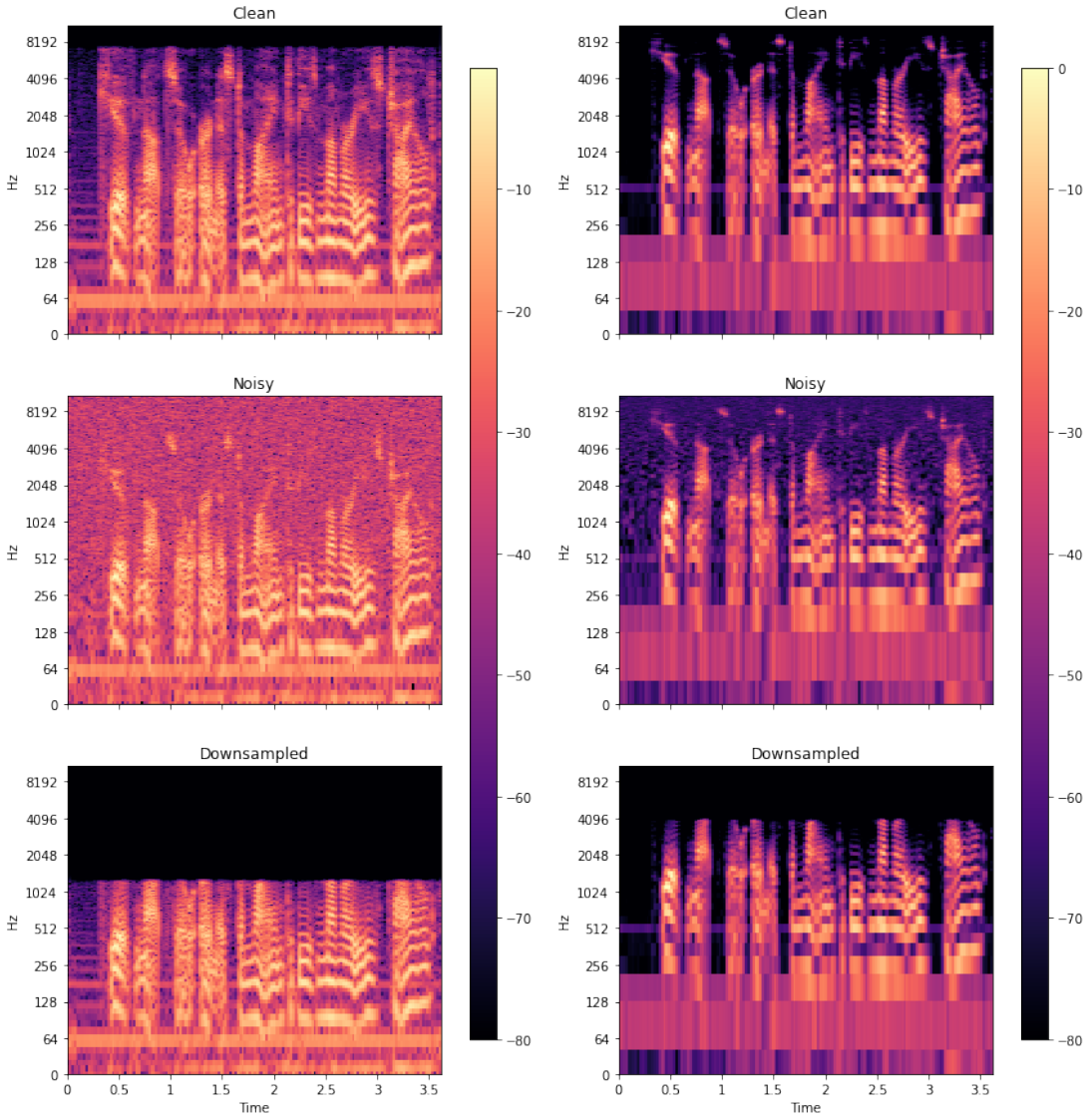
Original text transcription: `Give not so earnest a mind to these mummeries child.`



(a) (b)

Figure 3: Waveforms (a) and Fourier transforms (b) of the clean, noisy, and downsampled audio. In time-domain waveforms (a), the horizontal axis refers to the time in seconds, and the vertical axis refers to amplitude in dB. In frequency domain transformations (b), the horizontal axis refers to the frequency in Hz, and the vertical axis refers to amplitude in dB.

To begin, we plotted the waveforms of each audio file. The waveform for the clean data is the waveform generated from the original, unprocessed audio file. We can see that there is a spike in amplitude lining up with each word that is spoken, which confirms that our audio data looks like standard waveforms of the spoken English language. Looking at the noisy data, we can see that the amplitudes across the entire duration audio file have increased, which is especially noticeable in times when the speaker is between words or not talking (silence). We can see that our prepossessing successfully added the signals from the speech and white noise together to create noisy audio. Additionally, looking at the downsample waveform, we can see the amplitudes are smaller, showing that after downsampling the original audio (to 4 kHz) and upsampling (back to 16 kHz) we have lost some information. The Fourier transforms of our audio file, which is a transformation of signals from the time domain into the frequency domain, confirm the observation we saw above.



(a) (b)

Figure 4: Spectrograms (a) and Mel spectrograms (b) of the clean, noisy, and downsampled audio.

Next, we plot the spectrograms and Mel spectrograms of each of our audio files. We plot our spectrograms in order to be able to observe how the frequencies in our data change over time, with the x-axis being time, the y-axis being a frequency, and the color being the amplitude of that frequency at that time. Similar to the waveforms, we can visually connect the words from the speaker to time in the spectrogram, where periods of high amplitudes are various frequencies corresponding to when the speaker is talking. We observe in the noisy spectrogram that amplitudes of frequency that were once much lower in the original audio are now higher, showing the addition of white noise to the audio. We can also see that when we downsample the audio to 4 kHz and back up to 16 kHz, we lose information at high frequencies. We also plot the Mel spectrograms beside our normal spectrograms for comparison. The Mel scale is a logarithmic transformation of the signal's frequency and transforms the data to allow it to be more interpretable by humans. We see similar patterns to what we observed in the normal spectrograms.

2.4 Evaluation

The performance of our models is evaluated using word error rate (WER). WER is computed as:

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

where

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- C is the number of correct words,
- and N is the number of words in the reference (where $N = S + D + C$).

In some languages, the main unit of speech in the text is not words, formed by letters and demarcated by whitespace on either side, but instead characters. When testing in these languages, such as Chinese, we use a similar metric called character error rate (CER). CER is computed as:

$$\text{CER} = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

where

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- C is the number of correct characters,
- and N is the number of words in the reference (where $N = S + D + C$).

2.5 Results

We illustrate our results in the following four parts: main comparison, robustness to additive noise, robustness to downsampling, robustness to accents, and multilingual performance.

2.5.1 Main Comparison: wav2vec 2.0 vs. wav2vec 2.0 + 4-gram vs. Whisper Baselines

To familiarize ourselves with the models, we performed a set of experiments across three different models to recreate the results in their respective papers. The three models we used were wav2vec 2.0, wav2vec 2.0 with a 4-gram language model, and Whisper. The pretrained wav2vec 2.0 model we use is the 95M-parameter "base" model trained with 960 hours of labeled and unlabeled LibriSpeech data, and the pretrained Whisper model we use is the 74M-parameter "base.en" multi-lingual model. Out of the consideration of our limited computing sources, we choose these two pretrained models of similar sizes.

For our main comparison, we use the three pretrained models to test on the LibriSpeech test-clean and test-other datasets. They contain around 360 hours and 500 hours of speech data respectively, and the details of the datasets are shown in Appendix A.2. Additionally, we also test all models on the LibriSpeech test-clean dataset with different preprocessing techniques such as downsampling and adding noise as mentioned in Section 2.3, applied to the data.

Table 1: Main comparison between wav2vec 2.0, wav2vec 2.0 + 4-gram, and Whisper on LibriSpeech test data.

Model	test-clean	test-clean downsampling (8 kHz)	test-clean white noise	test-other
wav2vec2.0-base-960h	3.4	4.2	8.3	9.3
wav2vec2.0-base-960h + 4-gram	2.6	3.1	6.1	7.2
Whisper-base.en	4.3	4.8	6.1	10.4

The results of our main comparison are shown in Table 1. Comparing wav2vec 2.0 with wav2vec 2.0 + 4-gram, we can see that the wav2vec 2.0 + 4-gram model performed the best in all test categories, demonstrating the advantages of the additional 4-gram language model which may help with

choosing the logical homophone in the audio transcription. However, this performance improvement comes at the cost of added model complexity and run time. To our surprise, the supposedly better model, Whisper, does not show any advantages over wav2vec 2.0 except under the white noise condition – overall, it performs similarly to wav2vec. This result, however, appears reasonable when we look into the Whisper Large and wav2vec 2.0 Large 960h comparison found in Appendix A.4. Although these models are of different sizes, they perform similarly on the LibriSpeech test-clean. As a result, we understand that these models may not show divergent performances on the original LibriSpeech dataset; this thus inspires us to dive deeper into comparing their robustness under different kinds of low-quality conditions.

Since we want our ASR framework to be robust to speech hazards, we use our preprocessing functions to make already-labeled audio tracks more challenging to transcribe. This is a much more convenient way to proceed than creating new tracks from scratch. The preprocessing functions enable us to emulate background sound by adding white noise, and audio compression by decreasing the sample rate (downsampling). We envision performing light fine-tuning of these models in the future to keep most of their knowledge while making them more robust to hazards and more task flexible.

2.5.2 Robustness to Downsampling and Additive Noise

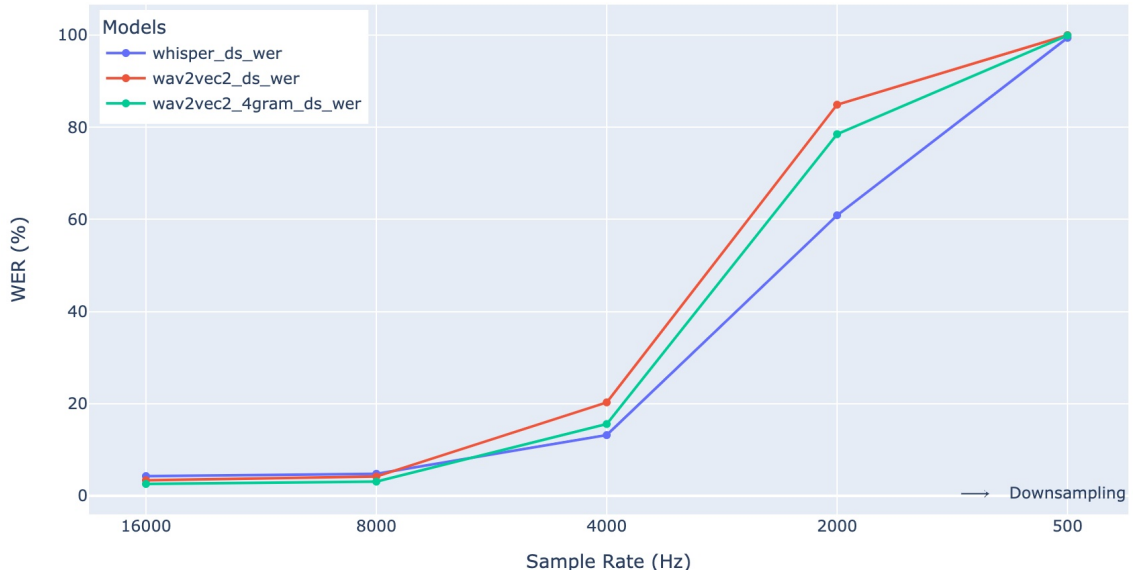


Figure 5: Robustness of Whisper_base.en, raw wav2vec2.0_base_960h, and wav2vec2.0_base_960h + 4gram to downsampling.

To see how model performance degrades with low-quality speech data, we evaluate the wav2vec 2.0 base 960h (with or without the 4-gram model) and Whisper base.en models on downsampled and noised-added versions of the LibriSpeech test-clean dataset.

For downsampling, we resampled the original 16000Hz speech data to 8000 Hz, 4000 Hz, 2000 Hz, and 500 Hz. The comparison of model performance is shown in Figure 5, where the plot reveals that downsampling to 8 kHz (sample rate of audio calls) has little impact on WER for both models. With lower sample rates, however, the over-interpolation of the sound signal seems to be too much to handle for both models. For example, for wav2vec 2.0, we can see a significant gap in performance from 4 to 2 kHz: a jump from 20 to 85% WER. This shows us that the high frequencies, which are cut off when downsampling, carry a lot of information useful to wav2vec 2.0 when transcribing. Besides, we can observe that in terms of robustness to a decreasing sample rate, with an additional language model, wav2vec 2.0 performs slightly better than the raw model. Compared to the two wav2vec-based models, Whisper demonstrates better robustness as the performance degrades more slowly when the sample rate drops.

As for adding white noise, Figure 6 reveals that adding white noise on a scale of a few percentage points ($1\% \sim 6\%$) on top of the audio track is enough to completely mislead both models, and the relationship between noise percentage and WER looks close to linear for both models within

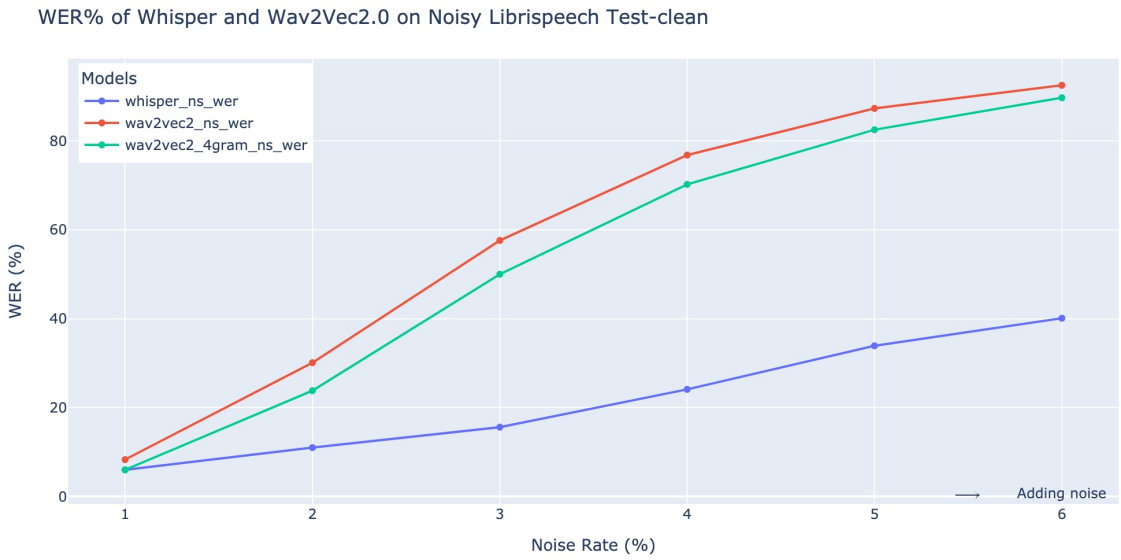


Figure 6: Robustness of Whisper_base.en and raw wav2vec2.0_base_960h to noise.

our chosen range. Unlike the similar performance drop between Whisper and wav2vec 2.0 in the downsampling evaluation, the two models appear to react differently to an increasing noise rate. In this case, Whisper demonstrates advantages over wav2vec as its WER grows relatively slowly and peaks at 40% when the noise rate is 6%, while wav2vec reaches over 90%.

Through these two comparisons, we chose ranges for noise percentage and sample rate so as to push the models all the way towards their points of complete inability to transcribe speech (100% WER). And consequently, this preliminary result showcases both models have the ability to handle hazards when building context, while Whisper appears to be more robust to noise. From this comparison we are inspired to conduct more research on different types of data and challenging tasks for the two models: for example, multi-speaker, multi-accent, and mixed language tasks. In doing so, we can conclude more about model characteristics and compare the usefulness among different deep learning-based speech processing techniques.

2.5.3 Robustness to Accents

We use the previously mentioned Speech Accent Archive dataset in 2.1.1 to test the ability of wav2vec 2.0 and Whisper to transcribe English audio from different accents. In Table 2 below, we can see the WERs of the base models of both wav2vec 2.0, wav2vec 2.0 + 4-gram language model, and whisper, and how they perform on accent audio data from the speech accent archive by the native language of the speaker.

Table 2: Comparison of WER of wav2vec2.0, wav2vec 2.0 + 4-gram, and Whisper on accent data.

Language (frequency)	wav2vec2.0	wav2vec2.0 + 4-gram	Whisper
English (579)	9.75%	9.40%	2.90%
Spanish (162)	27.21%	26.96%	18.07%
Arabic (102)	27.08%	27.00%	22.76%
Mandarin (65)	32.60%	32.60%	23.84%

2.5.4 wav2vec 2.0 XLS-R and Whisper Fine-tuned Performance on Fleurs

In Tables 3 and 4, we can see the performance of both wav2vec 2.0 XLS-R and Whisper on the Fleurs dataset before and after fine-tuning. Here, we mainly focus on 4 languages for fine-tuning and testing: Chinese, Korean, Hebrew, and Telugu. In Radford et al., a strong, negative linear correlation was found between hours of pretraining data and WER for languages, using Whisper

(see Appendix A.5); however, 4 languages were highlighted as deviating from this pattern, with unexpectly high WERs given the amount of training data available. As a result, these 4 languages may have the most room to improve, with further fine-tuning.

Table 3: wav2vec 2.0 XLS-R fine-tuned performance on Fleurs dataset (Chinese, Korean, Hebrew, and Telugu).

Language	# Pretrained Hours	# Fine-tuning Hours	Test WER w/o Fine-tuning	Test WER w/ Fine-tuning	Relative Improvement
Chinese	90	10	100.0%	25.8%	74.2%
Korean	61	8	100.0%	57.4%	42.6%
Hebrew	77	10	100.0%	60.8%	39.2%
Telugu	62	8	100.0%	78.3%	21.7%

Table 4: Whisper fine-tuned performance on Fleurs dataset (Chinese, Korean, Hebrew, and Telugu).

Language	# Pretrained Hours	# Fine-tuning Hours	Test WER w/o Fine-tuning	Test WER w/ Fine-tuning	Relative Improvement %
Chinese	23446	10	38.0%	19.7%	48.2%
Korean	7993	8	32.4%	28.4%	12.3%
Hebrew	688	10	70.6%	57.9%	18.0%
Telugu	4	8	120.7%	40.3%	66.6%

We can compare the impact of fine-tuning wav2vec 2.0 XLS-R and Whisper on a small amount of training data in different languages and observe the large increase in performance from both models. For this use case involving multiple languages, we use wav2vec 2.0 XLS-R instead of wav2vec 2.0, because wav2vec 2.0 is only able to process and transcribe English audio, as it was only trained on English audio samples. wav2vec 2.0 XLS-R is built on top of wav2vec 2.0 and learns cross-lingual speech representations by pretraining a single model from the raw waveform of speech in multiple languages. As a result, wav2vec 2.0 XLS-R is the multilingual version of wav2vec 2.0, having been trained on multiple languages from the Common Voice dataset.

In Table 3, we can see that wav2vec 2.0 performs very poorly on all 4 languages without fine-tuning, but after fine-tuning, we see the WER/CER of each language decrease drastically. One likely reason why pretrained wav2vec2 WER is 100% is that the pretrained model learns contextualized speech representations, and thus is a more general multi-task self-supervised model. Spotchecking the pretrained outputs yields results that are vastly different than the true label. In fine-tuning we add a linear layer, trained using labeled data, to focus our model on our exact use case. We see that the CER of Chinese improves the most, from 100% CER to 25.8% CER, which is a 74.2% relative improvement. Out of these 4 languages, Chinese was trained for the most hours in wav2vec 2.0 XLS-R and also was fine-tuned on the greatest number of rows. As a result, Chinse shows the largest improvement in CER. We can see that Telugu improves the least out of the 4 languages, with a 21.73% relative improvement from 100% WER to 78.3%. We found that grammatically, wav2vec 2.0 and all variations of the model found it hard to perform well on Telugu. We can also see that wav2vec 2.0 XLS-R was trained on a smaller amount of Telugu hours and fine-tuned on fewer training examples from Fleurs as well. Many of these languages still have a very high WER/CER even after fine-tuning, which is an area where more improvements can be made.

In Table 4, we can observe how Whisper performs on the Fleurs test data after fine-tuning. We can see that fine-tuning also decreases WER/CER for Whisper as well. For example, the performance of Whisper before fine-tuning on Chinese is a CER of 38.0%, and it improves to 19.7% after fine-tuning. Whisper is already trained on 23446 hours of Chinese audio, but fine-tuning still allows the performance to improve by 48.2% relatively. Additionally, we see that the improvement for Telugu (66.6% relatively) is the highest out of these 4 languages, from 120.7% to 40.3%. This drastic increase is likely due to Whisper only being trained on 4 hours of Telugu, but the lower layers in the model have learned more general features of the language well after being trained on other similar languages. This allows Whisper to decrease the WER of Telugu by a large percentage after fine-tuning.

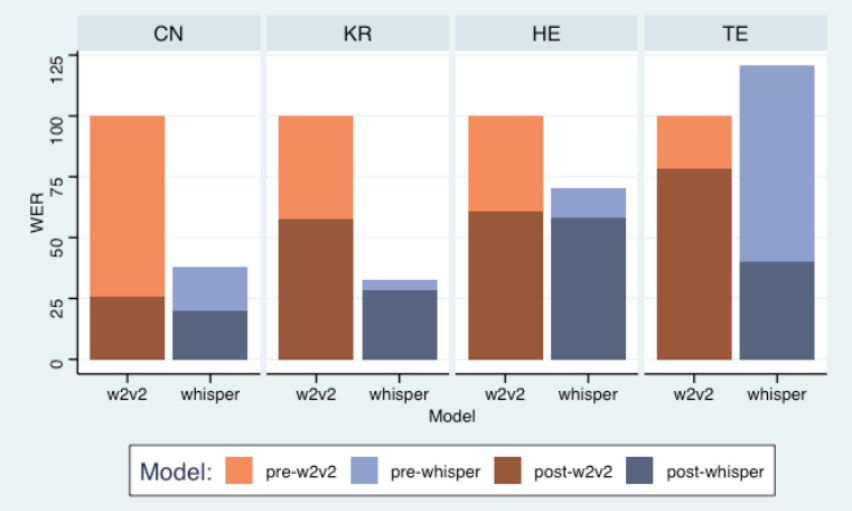


Figure 7: Comparison of wav2vec 2.0 XLS-R vs. Whisper before and after fine-tuning.

From Figure 7, we can compare the results of fine-tuning for wav2vec 2.0 XLS-R and Whisper. We can see that Whisper performs far better than wav2vec 2.0 XLS-R before fine-tuning for all languages except Telugu. This is because Whisper is trained with far more hours of audio data for most languages, except Telugu. Additionally, the architecture of both models is extremely different, and we can see that Whisper generalizes and performs better in most cases. After fine-tuning, Whisper ends up performing better than wav2vec 2.0 XLS-R in all languages, including Telugu. These results show that the power of Whisper to generalize exceeds that of wav2vec 2.0 XLS-R, as even with a small amount of fine-tuning data, Whisper can learn and improve the WER/CER in various languages. For both models, our custom fine-tuning pipeline drastically decreases the WER/CER from the pre-finetuned performances across all the languages that we focused on. Through our research, we successfully utilized the state-of-art open-source acoustic ASR models and enhanced the power and performance of those models by building custom fine-tuning pipelines and training on various open-source audio datasets.

2.5.5 Confidence Scores Analysis

In addition to the WER metric, we choose to use the logit score of a selected token as a proxy of the model's confidence in its prediction. In doing so, we are able to observe how the models struggle to transcribe with more granularity at the word/token level and do a side-by-side comparison of wav2vec 2.0 vs. Whisper on a single sentence. We can also compute a confidence metric for the whole prediction at the sentence level and check the correlation between prediction confidence and error.

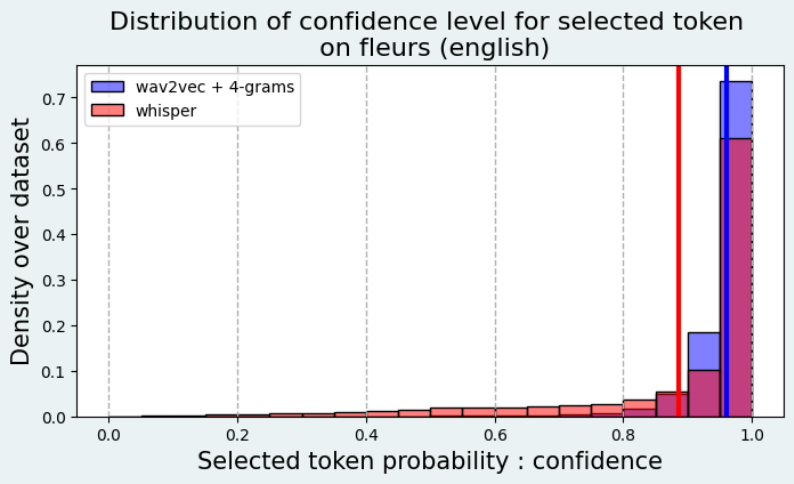


Figure 8: Density of confidence level per token : wav2vec 2.0 + 4-grams vs. Whisper. The vertical lines stand for the mean of their respective distributions.

Figure 8 shows the distribution of the selected token probability over Fleurs dataset for both wav2vec 2.0 + 4-grams and Whisper. As expected with highly trained ASR models, the confidence levels are high, with the majority of predicted tokens above 90% confidence. Overall, Whisper seems to be a little less confident in its prediction with a lower probability mean per token and a heavier left tail.

g.truth : the sport is primarily played in a duel format one fencer dueling another
w2v2+4g : the sport is primarily played in a dual form at one fencer duly another - confidence range : [84.4%, 100.0%] - WER= 28.6 %
whisper : the sport is primarily played in a dual format one fencer dualing another - confidence range : [51.9%, 99.6%] - WER= 15.4 %

g.truth : this might require filling out a form with the local police or a visit to the immigration offices
w2v2+4g : this might require filling out a form with the local police or visit to emigration offices - confidence range : [93.4%, 100.0%] - WER= 5.6 %
whisper : this might require filling out a form with the local police or revisit to the immigration offices - confidence range : [54.1%, 99.8%] - WER= 11.8 %

g.truth : police said they suspect an alleged daesh isil militant of responsibility for the attack
w2v2+4g : police said they suspect an alleged dash i s l militant of responsibility for the attack - confidence range : [92.0%, 100.0%] - WER= 29.4 %
whisper : police said they suspect an alleged dish isil militant of responsibility for the attack - confidence range : [15.0%, 99.8%] - WER= 7.1 %

Figure 9: Confidence scores at token level for predictions with wav2vec 2.0 + 4-grams and whisper

Figure 9 reveals that prediction errors tend to match tokens with lower confidence levels. In other words, when the model is not confident in its transcription, it is much more likely for it to be inaccurate. We also observe that the two models tend to struggle on the same portions of the sentences, with some words being particularly challenging (e.g. nouns like "daesh" or words with close pronunciation neighbors in the language like "duel"/"dual"). Models sometimes add extra words in their predictions, although it should be easily avoidable with a decent ability to read the audio. Usually, these words are linking words like prepositions and their presence in the predictions might be caused by the language model component of the models' architectures.

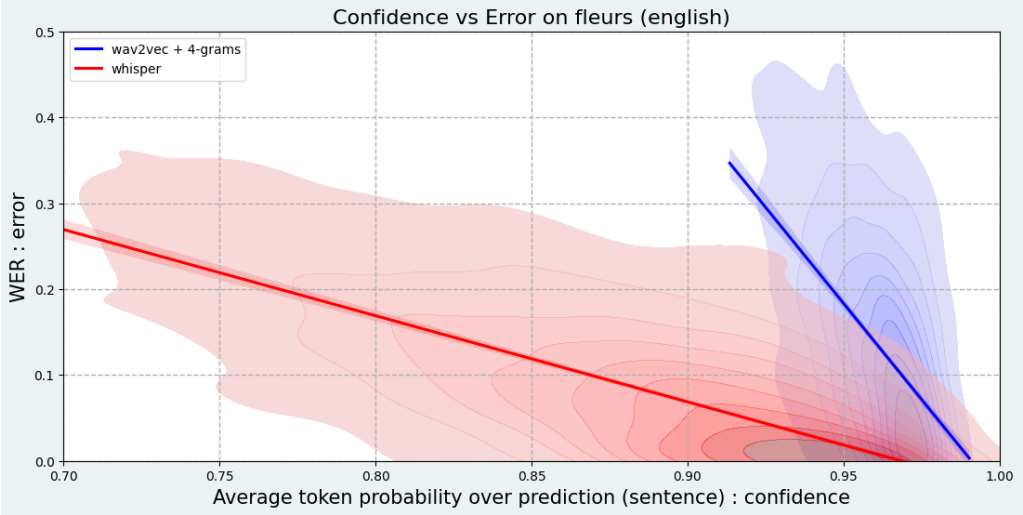


Figure 10: Kernel density estimation of average selected token probability across prediction vs WER for wav2vec 2.0 + 4-grams vs. Whisper. The lines show a linear regression.

Figure 10 reveals the negative correlation between confidence and error, as per intuition. Although — as we saw on Figure 8 — wav2vec tends to be more confident in its prediction on average, its WER is much more penalized by a slight decrease in confidence level, as indicated by the difference between the slopes of the regression lines.

3 Discussion

3.1 Summary

Whisper and wav2vec 2.0 were the two ASR models we focused on. The aim of our work was to analyze the audio transcription performance of the models in various contexts to test how well they can perform in a large set of situations. The idea is to have an idea of their ability to generalize well for certain domains or languages.

While a baseline comparison on the LibriSpeech dataset might indicate that wav2vec 2.0 performs similarly or even better than Whisper, altering the testing data by adding noise or by downsampling showed that wav2vec 2.0 suffers more WER degradation. This indicated that inference from audio samples that are less similar to the data the model has been trained on might favor Whisper over wav2vec 2.0.

One such occurrence might happen when dealing with non-native English speakers. Testing has once again confirmed that Whisper generally performs better in those situations where the transcription difficulty is higher.

The second part of our work was to perform fine-tuning on non-English languages. We used wav2vec 2.0 XLS-R which is a multilingual model. While any comparison is difficult due to the different architectures and pretraining data, fine-tuning often resulted in Whisper performing better than wav2vec 2.0 XLS-R. The main takeaway is that it is possible to perform fine-tuning on both models with relatively low resources and yield satisfactory results.

Finally, we computed confidence scores from the logit output of the models. A well-trained model is expected to have the confidence score negatively correlate with the WER, as observed in both instances, with wav2vec 2.0 being more confident than Whisper for similar error rates.

3.2 Conclusion

In conclusion, we performed in-depth research into using the open-source ASR models wav2vec 2.0 and Whisper and built custom fine-tuning pipelines to train those models on various open-source datasets, drastically improving the WER of those models on different use cases, such as multilingual speech and speech with accents. In our robustness tests adding white noise or downsampling clean audio from LibriSpeech, wav2vec 2.0 + 4-gram performs slightly better than Whisper in all base tests, but as the audio is downsampled to lower frequencies and more white noise is added, Whisper

starts performing better, showing that Whisper is more robust to greater disruptions and noise in the data. Similarly, we see that Whisper performed better than wav2vec 2.0 XLS-R both before and after fine-tuning on a multilingual dataset. Whisper is able to transcribe audio from different languages and accents into English with a much lower WER than wav2vec 2.0 XLS-R can, and the power of both of these models is greatly enhanced after fine-tuning on various open-source multilingual datasets in our pipeline.

3.3 Future Work

In interest of a holistic evaluation, we have studied how each model survives or fails on the above challenging tasks. However, we believe there are still more scenarios to explore for insights not only in terms of strict improvement, but also in different aspects including dataset curation, inductive biases, and training regimes.

In terms of a more in-depth evaluation of the generalizability of these state-of-the-art speech representations, we hope to continue examining their fine-tuned versions on other out-of-distribution multilingual datasets for each language. Moreover, we hope to see how different distributions of domain-specific speech data (e.g., healthcare-related speech) in training and testing affect their performance.

To push these speech representations to the edge, besides testing with degraded audio quality and accents, we may even test how they transfer across modalities from speech to singing audio as a variant of speech. We can also examine the transferability of these representations by testing them under more challenging tasks such as speech audio with multiple speakers with far-field speech and mixed languages. We have already identified potential open-source datasets, such as the VOiCES dataset and the DAPS (Device and Produced Speech) dataset, to work with for these use cases.

As a convenient solution for enhancing accuracy on certain tasks (e.g., ASR for 4 non-Indo-European languages), we have added more in-distribution supervision by fine-tuning. However, we have noticed alternative possible directions for improvement during the training stage, such as adding training objectives, regularization in architectures, and augmentation in audio processing.

Lastly, as a promising application of ASR, real-time speech recognition is extensively used in video transcription and voice services on devices. We hope to build a user-friendly interface demo to enable real-time interactions with our enhanced ASR models.

3.4 Ethics and Other Considerations

As with all deep learning models, there are ethical and security considerations that should be kept in mind when training and deploying the models for production. One ethical concern when building a multilingual model such as Whisper or wav2vec2.0 XLS-R is that it is nearly impossible for the developers to be fluent in all of these languages. Thus, when performing quality checks on translations or even verifying training datasets, it is important to have additional experts in each language to confirm that translations and transcriptions are reasonably accurate and appropriate in order for them to be included. This should be the case especially with languages included in models with only 4 hours of training data such as Telugu in Whisper; additionally, Telugu is among the languages potentially adversely affected by standard normalization practices, such as the removal of accents, which can greatly affect the final calculated accuracy as noted in Radford et al. [2]. Without this validation, it could be possible to build erroneous models for certain languages, which could be detrimental and unfair to certain demographics.

Secondly, as more of one's personal data is exposed and shared online, differential privacy has become an important topic to discuss and consider. One's privacy should be ensured within the end-to-end modeling process, and one's identity should not be identifiable from malicious hacking of model training gradients. While a cleaned and labeled dataset like LibriSpeech might seem like an uncontroversial training dataset, it can be shown that it is possible to identify the identity of the speaker in training data with up to 34% top 1 accuracy and 51% top 5 accuracy [13]. With the assumption that the correct training transcription label is known, it can be shown that by gaining access to training gradients, we can reconstruct speech features using Hessian Free Gradient Matching in a speaker identification model. As long as there are publicly available speech samples for each of the speakers to help identify them, we can compare the embeddings of the reconstructed speech features to the embeddings of the public speech examples, where the embeddings with the same speakers should be the most similar. Dang et al. mention possible solutions such as dropout

and Differentially Private SGD; however there is a significant performance hit by using these techniques, and is a considerable tradeoff to factor in.

Finally, adversarial attacks have become more prevalently studied and are a concern for ASR models as well. It has been shown that slight perturbations of data can cause significantly different predictions for deep learning models [14]. While the majority of research has been on image classification, recently there has been advancement in the field of ASR models. It has been shown that using psychoacoustic hiding techniques, two audio clips might sound exactly the same to the human ear, but the ASR model might output something completely different for each [15], i.e. an audio clip of "Alexa, order toilet paper" could be transcribed into "Alexa, order a tv to this address". With a growing prevalence on Internet of Things, both Alexa and Google devices have the ability to control shopping, security alarms, etc., and adversarial attacks could have devastating effects. While there have yet to be any real life examples of abuse of these ethical or security vulnerabilities, these are all considerations that should be kept in mind when training and deploying any deep learning model.

4 Contributions

- **Sivan Ding:** evaluated pretrained Whisper on LibriSpeech with degraded audio quality; evaluated pretrained Whisper on Fleurs; fine-tuned Whispers on four languages with data preprocessing; evaluated fine-tuned Whisper on Fleurs.
- **Alexandria Guo:** worked on language analysis section of EDA, conducted experiments with Whisper on unmodified, downsampled, and noisy audio tracks. Worked on preprocessing of Fleurs and further fine-tuning of Whisper for Chinese on Common Voice.
- **Anh-Vu Nguyen:** set up GCP instances for the group, and created documentation. Did some early testing on wav2vec with LibriSpeech. Worked on the audio preprocessing functions to generate noisy datasets. Worked on testing whisper on the accent data. Added the ability to output confidence scores from the wav2vec 2.0 output, and contributed to the fine-tuning of wav2vec XLS-R.
- **Antonin Vidon:** preprocessed datasets, conducted experiments with raw wav2vec 2.0 on downsampled and noisy audio tracks, conducted confidence scores analysis.
- **Julia Wang:** worked on the preprocessing functions, conducted noisy environment experiments with wav2vec 2.0 with 4-grams. Additionally, evaluated pretrained wav2vec 2.0 XLS-R language models on Fleurs, as well as manually fine-tuned wav2vec 2.0 XLS-R on Korean.
- **Maxwell Zhou:** worked on finding datasets and on EDA for audio data, conducted experiments with wav2vec 2.0 and wav2vec 2.0 with 4-grams. Worked on preprocessing accent data from speech accent archive and performing testing on accent data with wav2vec 2.0. Worked on building pipeline and fine-tuning wav2vec 2.0 XLS-R on Fleurs - Hebrew and tested base model and fine-tuned model performances.

References

[1] Baevski, A., Zhou, H., Mohamed, A., and Auli, M. 2020. "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations" in *Advances in Neural Information Processing Systems.*

[2] Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. 2022, Preprint. "Robust Speech Recognition via Large-Scale Weak Supervision" in *OpenAI.* Available at `https://cdn.openai.com/papers/whisper.pdf`.

[3] Ravanelli, M., and Bengio, Y. 2018. "Learning Speaker Representations with Mutual Information" in *Proc. Interspeech 2019.*

[4] Zeghidour, N., Xu, Q., Liptchinsky, V., Usunier, N., Synnaeve, G., and Collobert, R. 2018. "Fully Convolutional Speech Recognition" in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.*

[5] Schneider, S., Baevski, A., Collobert, R., and Auli, M. 2019. "wav2vec: Unsupervised Pre-training for Speech Recognition" in *Proc. Interspeech 2019*.

[6] Conneau, A., Baevski, A., Collobert, R., Mohamed, A., and Auli, M. 2020. "Unsupervised Cross-lingual Representation Learning for Speech Recognition" in *Proc. Interspeech 2021*.

[7] Hsu, W., Bolte, B., Tsai, Y.H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. 2021. "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units" in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

[8] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. 2015. "LibriSpeech: An ASR Corpus Based on Public Domain Audio Books" at *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5206–10. doi: /10.1109/ICASSP.2015.7178964.

[9] Conneau, A., Ma, M., Khanuja, S., Zhang, Y., Axelrod, V., Dalmia, S., Riesa, J., Rivera, C., and Bapna, A. 2022. "FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech" in *CoRR* abs/2205.12446 (2022).

[10] Weinberger, S. 2015. *Speech Accent Archive*. Retrieved from `http://accent.gmu.edu`.

[11] Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F.M. and Weber, G. 2019. "Common Voice: A Massively-Multilingual Speech Corpus" in *arXiv*.

[12] Mines, M.A., Hanson, B.F., and Shoup, J.E. 1978. "Frequency of Occurrence of Phonemes in Conversational English" in *Language and Speech*, 21(3), 221–241.

[13] Dang, T., Thakkar, O., Ramaswamy, S., Mathews, R., Chin, P., and Beaufays, F. 2021. "A Method to Reveal Speaker Identity in Distributed ASR Training, and How to Counter It" at *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4338-42. doi: /10.1109/ICASSP43922.2022.9746443

[14] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. 2013. "Intriguing Properties of Neural Networks" in *2nd International Conference on Learning Representations, ICLR 2014*.

[15] Schönherr, L., Kohls, K., Zeiler, S., Holz, T., Kolossa, and D. 2018. "Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding" in *26th Annual Network and Distributed System Security Symposium, NDSS 2019*.

A Appendix: Figures

Included are relevant figures from papers cited above, for easy reference.

Model	Unlabeled data	LM	dev		test	
			clean	other	clean	other
LARGE - from scratch	-	None	2.8	7.6	3.0	7.7
	-	4-gram	1.8	5.4	2.6	5.8
	-	Transf.	1.7	4.3	2.1	4.6
BASE	LS-960	None	3.2	8.9	3.4	8.5
		4-gram	2.0	5.9	2.6	6.1
		Transf.	1.8	4.7	2.1	4.8
LARGE	LS-960	None	2.6	6.5	2.8	6.3
		4-gram	1.7	4.6	2.3	5.0
		Transf.	1.7	3.9	2.0	4.1
LARGE	LV-60k	None	2.1	4.5	2.2	4.5
		4-gram	1.4	3.5	2.0	3.6
		Transf.	1.6	3.0	1.8	3.3

Figure A.1: From Baevski et al. [1]: WER on LibriSpeech when using all 960 hours of LibriSpeech as labeled data.

subset	hours	per-spk minutes	female spkrs	male spkrs	total spkrs
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-other	5.1	10	17	16	33
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166

Figure A.2: From Panayotov et al. [8]: Data subsets in LibriSpeech.

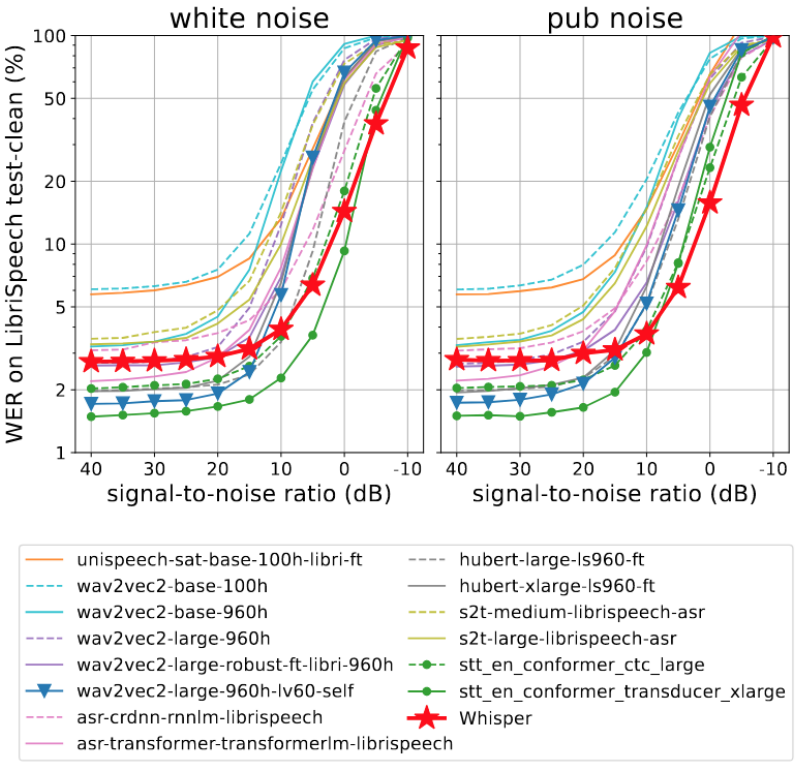


Figure A.3: From Radford et al. [2]: WER on LibriSpeech test-clean as a function of SNR under additive white noise (left) and pub noise (right).

Dataset	wav2vec 2.0 Large 960h	Whisper Large	RER (%)
LibriSpeech test-clean	**2.7**	**2.7**	0.0
Artie	24.5	**6.7**	72.7
Fleurs (English)	14.6	**4.6**	68.5
Common Voice	29.9	**9.5**	68.2
Tedlium	10.5	**4.0**	61.9
CHiME6	65.8	**25.6**	61.1
WSJ	7.7	**3.1**	59.7
VoxPopuli (English)	17.9	**7.3**	59.2
AMI-IHM	37.0	**16.4**	55.7
CallHome	34.8	**15.8**	54.6
Switchboard	28.3	**13.1**	53.7
CORAAL	38.3	**19.4**	49.3
AMI-SDM1	67.6	**36.9**	45.4
LibriSpeech test-other	6.2	**5.6**	9.7
Average	29.5	**12.9**	55.4

Figure A.4: From Radford et al. [2]: Detailed comparison of robustness on various datasets between wav2vec 2.0 Large 960h and Whisper Large.

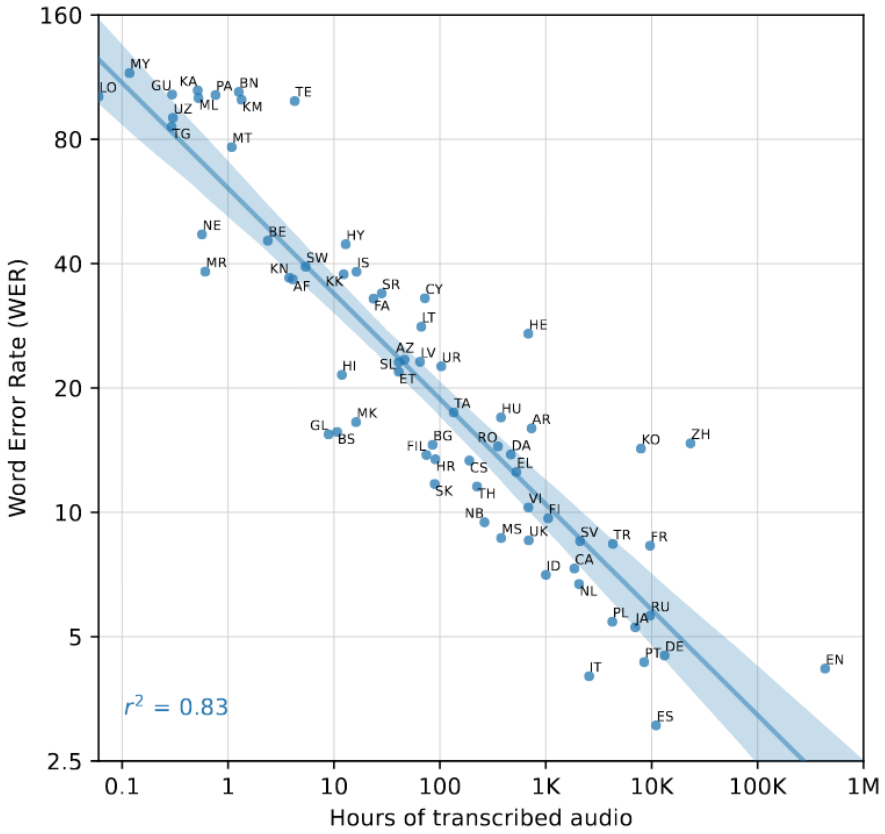


Figure A.5: From Radford et al. [2]: Correlation of pre-training supervision amount with downstream speech recognition performance.