Wav2Vec Transcriptions for Second Language Speakers

Madurya Suresh
masuresh@ucsd.edu

1 Introduction

Generally speaking, it is important for NLP models to be robust to all kinds of inputs that they might see while performing their task. As the use of NLP models for Automatic Speech Recognition (ASR) is quite popular already, and continues to grow in usages, it's vital that ASR is robust to a wide variety of speakers of a given language, including speakers of different dialects or second language speakers. I wanted to investigate how well Facebook's Wav2Vec2 960h (Baevski et al., 2020) performs ASR on audio data from second language speakers (English is not the first language) vs. native speakers. Wav2Vec2 960h is a transformers-based model, pretrained on thousands of hours of data, and fine-tuned on hundreds of hours of English audio data. Because it was fine-tuned on English data, I was curious to see how it would perform on diverse examples from English. Here is what I set out to do, and what I accomplished:

- Collect, create, and preprocess test dataset: DONE.
- Build and run inference using Wav2Vec2 960h on collected dataset and examine its performance: DONE.
- Collect, create, and preprocess train, eval datasets: DONE.
- Finetune Wav2Vec2 960h on a train, eval datasets for the ASR task: DONE.
- Compare results of the two models on test set: DONE.

2 Related work

I reviewed work pertaining improving ASR robustness.

In (Chang and Chen, 2022), the authors discuss different techniques for improving ASR robustness during pre-training and fine-tuning. They suggest contrastive learning, both self-supervised and supervised, in which you make "noisier" embeddings look closer to some chosen embedding for the label, so that their representations are more similar. They also use self-distillation, which is a technique used to reduce label noise while training.

In (Cui et al., 2021), the authors discuss how we can leverage pre-trained language models to actually create ASR noise to train models on to improve robustness – offering a new and supposedly better approach to data augmentation. Most interestingly, their noisy-data-trained ASR models keep better track of semantic meaning when performing ASR than the baselines, making them less likely to have ASR errors.

In (Cheng et al., 2023), the authors show how one can use two models, one trained on manually annotated transcripts, and one trained on ASR transcripts, can train together and learn from each other (mutural learning). This paper argues that currently, ASR transcripts and annotated transcripts are treated the same, and this ends up causing contrastive learning to push the representations away from each other, when they should be semantically similar. They use self-supervised and supervised contrastive learning, as well as mutual learning and self-distillation to improve the representations of the models.

In (Likhomanenko, 2021), the authors argue that simply training the model with additive noise in the data will greatly improve model robustness to real-world noise, and the model's ability to

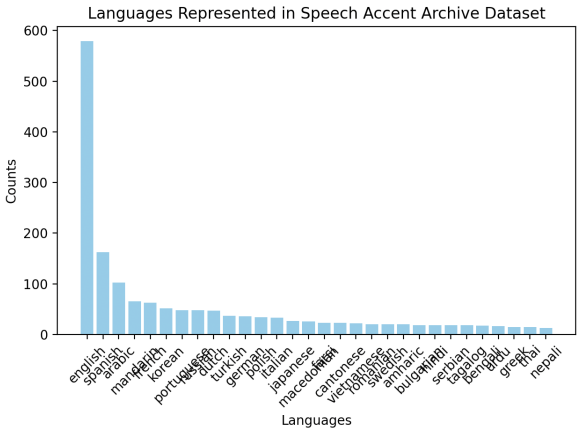
generalize. They also argue that testing on any single dataset is not sufficient for fully assessing a model's performance and robustness, and a diverse set of real-world, noisy data is needed to fully test a model.

In (Hirayama et al., 2012), the authors discuss building ASR models robust to different dialects and variance in speech. They use phoneme-sequence transducers to transform the input from raw linguistic corporas, and they train the language models using these transducers. They essentially use come up with predictions based on the phonemes, then assign them to pronunciations. Although this paper is a bit older, I find it interesting, as in the tutorial I used for my fine-tuning later, it was mentioned that using phonemes is super effective in ASR, so these techniques seem to continue to be used today.

3 Your dataset

I used two datasets in my project: the Speech Accent Archive and the L2 Arctic Corpus.

The Speech Accent Archive contains audio recordings of many speakers from different first-language backgrounds, from English, to Vietnamese, to Amharic. For some languages, there are many different speakers (each has one audio clip), but for most, there are 1-3 speakers each. To give an idea of how many languages are represented in this dataset, here is a bar graph showing the distribution of audio recordings across a select set of languages (there are 200 languages total, and 2138 audio files total, so I couldn't include all languages in one visualization):



In each audio file, the speaker reads out the same text:

"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."

Because they all read out the same text, this gives a good idea of how the Wav2Vec 960h model will perform when only the speaker differs, and not the ground truth (non-phonetic) transcription.

The L2 Arctic Corpus similarly contains audio files from second-language English speakers, along with transcriptions from each. Unlike the Speech Accent Archive, L2 Arctic Corpus contains natural language (speakers were given a prompt to answer independently) rather than reading off of a given text.

There are less languages represented in the L2 Arctic Corpus than the Speech Accent Archive, as the L2 Arctic Corpus only has 6 different native languages (compared to 200!). The L2 Arctic Corpus has 40295 transcribed audio files.

Because the Speech Accent Archive essentially has one ground truth label (the passage mentioned above), it doesn't make sense to fine tune a model on this, because it would learn what the correct answer was trivially, rather than learn different representations of the audio data. So, this is why

I trained on the L2 Arctic Corpus, and then tested on the Speech Accent Archive.

Additionally, because of compute power limitations, I could only use a sixteenth of the L2 Arctic Corpus. So, I had 1290 entries in the train set, and 160 entries in the eval set.

There are 2138 entries in the test set (Speech Accent Archive). I removed some files from the most common language audio clips (such as English, Spanish, Arabic), and reduced the test set to 1490, to account for my limited computing resources.

3.1 Data preprocessing

I did dataset creation for the L2 Arctic Corpus. The audio data and transcripts were in separate files across different folders, and I created a json list dataset, which can be unpacked for training.

In terms of audio data preprocessing for both datasets, I resampled all audio files in the dataset to 16000. For the L2 Arctic Corpus, I also padded the audio files to a max length of 3 seconds. For training, the audio data is converted into an array, and then passed through the Wav2Vec2Processor, which processes the audio data. Additionally, I removed special characters and turned the text data lower case. The text data is also padded to a max length.

For training and evaluation of my fine-tuned model, I created a vocabulary of characters, which the model uses to make its predictions. For reference, I used a tutorial on how to Fine-Tune Wav2Vec2 for English ASR to help me learn how to do the data preprocessing and fine tuning process for Wav2Vec for a specific ASR task.

4 Baselines

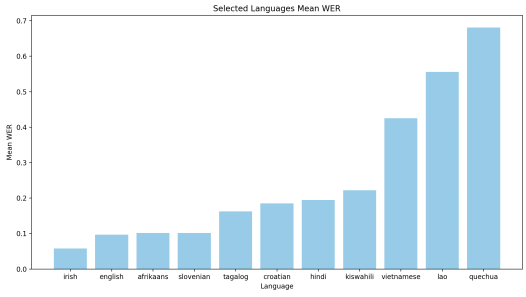
I used Wav2Vec2ForCTC pretrained model Wav2Vec-base-960h, and the Wav2Vec2Processor to process inputs and decode predictions. This is the baseline for my ASR task. I used Word Error Rate (WER) to compute the baseline's performance.

Here is an example transcription done by the baseline model, for a first language speaker of Spanish:

PLEASE CALL A STAILOR ADS HER TO BRIN THISE THINGS WITH HER FROM THE STORE SIX SPOONS OR FRESH A SNOW PIESE FI THICK ASLAFF OF BLUE CHEES AND MAYBE AND SNAK FOR HER BROTHER BUB WE ALSO NEED A SMALL BASTY A SNAKE AND A BIG TOY FROK FOR THE KEYSE SHE CAN ASCOOP THESE THINGS INTO THREE RED BAGS AND WE WELL GO MEET HER GWEN STAY AND THE TRAIN ASTATION

I evaluated this setup on the Speech Accent Archive dataset and analyzed the WER – overall, and aggregated by language (mean WER). The average WER across all languages was **0.2869**, which is a great baseline! The standard deviation from the mean was 0.1550. While the average WER over all languages was 0.2869, some languages had as high as .88 for their WERs. So, there is still room for improvement for some languages.

I selected certain languages (many audio clips, some audio clips, 1 or 2 audio clips) to show their average WER:



As we can see, the model performs really well on English, and some of its worst WER scores were for Vietnamese, Lao, Quechua, among other languages. This aligns with my predictions that English native speakers would have the best WER (Irish having a better WER was unexpected, but there was only one audio clip, so it's hard to say it's significant. Whereas, there were many English clips), and second language English speech has worse WERs, to varying degrees.

5 Your approach

I trained the baseline model for 30 epochs, evaluating the model at every 500 steps. I set the learning rate to 1e-4, weight decay to 0.005, and warm up steps to 1000. This was based on the tutorial I mentioned before. Also, I used the same Wav2Vec2Processor as the baseline, but added the Wav2Vec2FeatureExtractor and Wav2Vec2Tokenizer. This was because, as mentioned in the Dataset Section, we had a custom vocabulary that was used for fine-tuning. To be clear, both models used the same processor.

Before using the above hyperparameters, I trained the model for 10 epochs, with a learning rate of 1e-10. The model did not improve at all. Under the new hyperparameters mentioned above, however, it was able to converge.

- (CONCEPTUAL APPROACH) I fine-tuned Wav2Vec-base-960h on a supervised task, where it was given the audio data input, and the label was the transcribed text. Because it was coming from a dataset of second language speakers, the hope is that the model will transcribe the ground truth English translations, despite smaller variances in pronunciation.

- (WORKING IMPLEMENTATION) I was able to create a working implementation that can be loaded and it can perform inference on audio data. It does not perform better than the baseline. But, when we look at the input later, we can see that it grasps the basic idea of most words. Still, the baseline is best at transcription for second language English speakers. I think it doesn't perform as well as the baseline for a couple reasons: (1) I wasn't able to use the best checkpoints (see RESULTS) in my test data evaluation, (2) there are more languages in the test data than in the train data, (3) perhaps contrastive loss could be a more useful strategy for alignment, and (4) perhaps training a model for transcription for each accent could be interesting. See finetune.ipynb for my code for training the model.

- (COMPUTE) I ran into many issues trying to get the model to train on my local computer! As I only have a CPU, I found an interesting problem, where the gradnorm would be NaN, and the loss would be 0.0. After some research, I found that either this could happen because of exploding gradient, or incompatible floating point precision on CPU vs. the GPU that the original model was trained on. I tried clipping the gradient, but still had the same problem. I transferred my data and code to Colab, used the GPU, and it worked! However, I still encountered issues with lack of storage and compute units.

- (RUNTIME) It took about one hour to train my model.

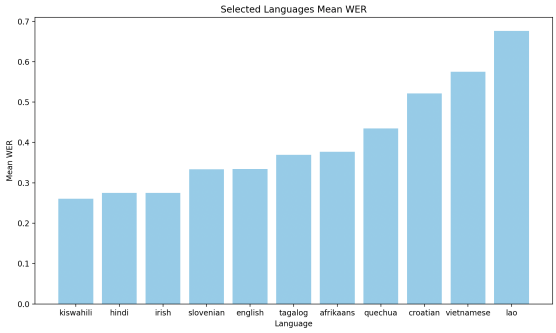
- (RESULTS) My model achieved a best WER on the validation dataset of **.271**. I had to stop the model training early because I ran out of storage. Additionally, I lost the best checkpoints at the end because of lack of storage on Google Drive/Colab. So, I had to use checkpoint 2500, with a WER of **.344**, to evaluate the fine-tuning.

  I tested the model on the Speech Accent Archive dataset, which has more speaker backgrounds than the training set, so hopefully the model will be able to generalize well to many different accents.

  For the baseline model, as aforementioned, I tested it on all the 1490 entries from the test set. Due to limited computing resources, I had to randomly select around a quarter (250) of that number to compute the average WER for my fine-tuned model. This is unfortunate, as I'd have liked to compare them directly, but my computer was unable to run this long of a process at once. Because I randomly selected the values, hopefully, it will still give a good comparison of the average WER over all samples.

  My fine-tuned model had a WER of **0.4760** on the test set, with a standard deviation of 0.1296. This is compared to the baseline,

which had a WER of **.2869**. Here's how the fine-tuned model performed on the selected languages that we also showed the baseline's performance on before:



The best WER here for the fine-tuned model is close to 0.3, whereas the best WER for the baseline was close to 0.05.

With more training, and perhaps techniques like contrastive loss integrated into the training pipeline, the fine-tuned model could prove to be more robust to second language English speech than the baseline.

6 Error analysis

Here is an example input that my baseline failed on, from a speaker whose native language is Macedonian:

PLEASE CALLS TELLER ASK HER TO BRING THESE THINGS WITH HER FROM THE STORE SIX SPOONS OF FRESH SNOW PEESE FIVE THICK SLABS OF BLUE CHEESE AND MAYBE A SNAK FOR HER BROTHER BOB WE ALSO KNEED A SMALL BLASTIC 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 A TRAIN STATION

Here is the same example input, passed through my fine-tuned model:

PLEASE CALL TELLA AS HER TO BRING THESE THINGS WITH HER FROM THE STORE SIX SPOONS OF FRESH NO PEASE FIVE TIX SLABS OF BLUE CHES AND MAY BE A SNAK FOR HER BROTHER BOB WE ALSO MET A SMAL PLASTICK SNAKE AND A BIG TOI FROG FOR THE KIS SHE CAN SCOOPE THESE THINGS INTO THREE RED BAGS AND WEWVILL GO MET HER WENESDAY AT THE TRANSTATION

I listened to the actual audio clip, and her speech was definitely able to be transcribed into the correct English spellings, if a human were to transcribe it manually. Interestingly, my fine-tuned model got "THE" at the end, which was correct according to what I heard, whereas the baseline model put "A". For both models, it seems like there are some spelling errors that result in non-English words. It could be useful to integrate some way to ensure that they pick the nearest valid English token. This seems to be the biggest area that the baseline model messes up on. The fine-tuned model seems to be much worse at this than the baseline, as well.

7 Conclusion

Some takeaways that I have from this project when it comes to the task I chose, is that training ASR for different dialects/accents is an extremely broad brush! The diversity of speech within one language is so vast – a very complex model, or perhaps many models working together, might be able to tackle the task best.

What was difficult to accomplish was when I came across the floating point precision and gradient explosion issue when I was trying to train on CPU. It was not a problem with my code, but rather with the machine. I often forget that all the computations are really happening at a low level, even if high level code seems to be correct.

If I could continue working on this project in the future, I would definitely want to implement contrastive loss. Here, I would want to see how I can bring different accents closer to the native speaker embeddings (although this is still painting a very broad brush!). This would be useful in some use cases – say some sort of legal ASR model!

8 Acknowledgements

I did not use an AI tool to help me write this report.
I did use ChatGPT to help me with pandas in

Python. I made moderate changes to its outputs.

References

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations.

Chang, Y.-H. and Chen, Y.-N. (2022). Contrastive learning for improving asr robustness in spoken language understanding.

Cheng, X., Cao, B., Ye, Q., Zhu, Z. Z., Li, H., and Zou, Y. (2023). Ml-lmcl: Mutual learning and large-margin contrastive learning for improving asr robustness in spoken language understanding.

Cui, T., Xiao, J., Li, L., Jiang, X., and Liu, Q. (2021). An approach to improve robustness of nlp systems against asr errors.

Hirayama, N., Mori, S., and Okuno, H. G. (2012). Statistical method of building dialect language models for asr systems.

Likhomanenko, T. e. a. (2021). Rethinking evaluation in asr: Are our models robust enough?