Phonetic Transcription for Low Resource Languages Using Cross Lingual Transfer

Kyle Ng

1. Introduction

While Automatic Speech Recognition (ASR) from audio to text is quite robust for high resource languages, lower resource languages (<10 hours of labeled speech data) make universal multilingual ASR difficult to scale.

Thus, this paper introduces a new model Wav2Vec2IPA fine-tuned on International Phonetic Alphabet (IPA) transcriptions of the TIMIT speech corpus aiming to understand if audio to IPA ASR pipelines can aid cross-lingual transfer in search for universal multilingual ASR.

To evaluate model performance, we compare our new model against a multilingual Wav2Vec2 variant, Wav2Vec2XLSR. Since the TIMIT dataset uses a different phonetic transcription than IPA, we devised a preprocessing step which aligns TIMIT phonetic transcriptions to the IPA output of this Wav2Vec2 variant's vocabulary.

Rather than using the traditional Word Error Rate (WER) metric, we evaluate our model on a normalized Phoneme Error Rate (PER) and minimum edit distance.

Our results show that although the model minimizes training loss to 0.10, validation loss is 0.65 with a normalized (phoneme) error rate of 100% indicating over fitting on the training dataset and thus underperforming the baseline with an error rate of 0.285.

Code: https://github.com/Ky-Ng/IPA-ASR

2. Background

2.1. Phonetics and Phonology

Linguistic theory (phonetics and phonology) proposes that the production and interpretation of sound units (phonemes) are universal across all languages. In other words, sounds in specific languages can be mapped to a language agnostic representation that are shared between many languages.

Thus, the goal of this project is to train the base Wav2Vec2 model to learn a universal transcription which could be robustly applied to a novel language the model has not seen.

Thus, we specifically avoided the use of a Language model on top of Wav2Vec2 such as implementations in Wispr or Wav2Vec2-BERT in order to avoid depending on the word co-occurence in a lanaguage which may not be consistent cros-linguistically.

3. Related Work

3.1. Wav2Vec2

The Wav2Vec2 Model(Baevski et al., 2020) released by Meta AI introduced learning latent speech representations directly from raw waveforms to robust latent speech representations. These representations can be fine-tuned for a variety of downstream tasks such as audio classification or ASR.

Wav2Vec2 feature extraction from raw waveforms is replaced the traditional Mel-Frequency Cepstral Coefficients (MFCCs) derived from Discrete Fourier Transforms (DFT).

Wav2Vec2 first encodes raw waveforms using a multi-layer CNN followed by a 12 deep transformer block trained on a contrastive loss task (differentiating between phonemes).

To handle phoneme alignment, the Connectionist Temporal Classification alignment scheme is used.

3.2. Wav2Vec2XLSR

Wav2Vec2XLSR (Cross Lingual Speech Recognition)(Conneau et al., 2020) builds upon Wav2Vec2 by simultaneously training latent speech representations with multilingual IPA transcriptions.

The architecture is almost idential to Wav2Vec2 and uses a shared CNN and transformer encoder across audio inputs from 53 languages across 10 language families (CommonVoice, BABEL, Multilingual LibriSpeech).

4. Dataset and Evaluation

4.1. Dataset

We fine-tuned the Wav2Vec2-Base model on the TIMIT[1] database which contains 5 hours of read speech with

[1]Texas Instrument Massachusets Institute of Technology TIMIT on Hugging Face

phoneme level transcriptions sampled at 16kHz. We used a 70-10-20 Train-Validation-Test split.

Since the database was created in 1993 at a time where IPA symbols could not be digitalized, this paper introduces a phonemic transcription of the 1993 TIMIT phonetic symbols to current 2024 IPA transcription.

Table 1. Sample of the proposed translation system between the 1993 TIMIT and 2024 IPA transcription

TIMIT	IPA
tcl	t
t	t^h
ng	$ŋ$
aw	$aʊ$

After applying the TIMIT to IPA lookup table, we then remove any IPA symbols not present in Wav2Vec2XLSR for baselines comparisons[2] since it is easier to compare a base model on a reduced set of vocabulary in the already fine-tuned XLSR model.

4.2. Evaluation

This paper uses 2 main methods for evaluation: (1) Minimum Edit Distance[3] and (2) Normalized Phoneme Error Rate

We use Phoneme Error Rate rather than the traditional ASR Word Error Rate since the models we are looking to evaluate are on a character level rather than a word level[4].

4.2.1. Edit Distance

We use edit distance as a naive[5] metric to understand how far away two transcriptions are.

$$\textbf{Edit Distance} = \textbf{substitutions} + \textbf{deletions} + \textbf{insertions}$$

4.2.2. Phoneme Error Rate

Secondly, we consider phoneme error rate or edit distance normalized by number of phonemes.

$$\textbf{PER} = \frac{\textbf{Edit Distance}}{\text{len(ground truth transcription)}}$$

However, PER over-penalizes model transcriptions which are much longer than the ground truth prediction:

Case 1) Truth = "ABC" and Prediction = "ABC12345"

$$PER = \frac{\textbf{Edit Distance}}{\text{len("ABC")}} = \frac{5}{3} = 1.667$$

Case 2) Truth = "ABC12345" and Prediction = "ABC"

$$PER = \frac{\textbf{Edit Distance}}{\text{len("ABC12345")}} = \frac{5}{5} = 1.0$$

In order to make our PER metric symmetric with respect to truth and prediction, we use normalized PER[6]:

$$\textbf{PER'} = \frac{\textbf{Edit Distance}}{\textbf{max}(\text{len(truth transcription), len(predict transcription)})}$$

4.2.3. Loss Function: Connectionist Temporal Classification

For the loss function we used to train the transformer layers in Wav2Vec2 we use the Average Connectionist Temporal Classification Loss which uses a Recurrent Neural Network (RNN) to handle sequence alignment between the prediction and truth prediction.

5. Methods

This paper evaluates two models: (1) Wav2Vec2XLSR and (2) a fine-tuned Wav2Vec2 model on 340 examples (5 hours) of TIMIT IPA transcriptions.

Since the first model is already trained, we applied the model to the Edit Distance and Phoneme Error Rate.

To ensure that the data is compatible for the Wav2Vec models, we used a resampling pipeline that automatically detects the input audio sampling rate and resamples to 16kHz. Since TIMIT was recorded at 16kHz this step was used mainly as a safeguard for other datasets that maybe used in the future.

Next, we fine-tuned wav2vec2 using 100 epochs of training at a learning rate of 10^{-4}. These hyperparameters were recommended by the Wav2Vec2 Hugging Face guides.

6. Experiments

6.1. Baseline Results

For Wav2Vec2XLSR, we report an average edit distance of 10.2 and an average PER of 0.285.

[2]There are cases where we have a release gesture "t" without a closing gesture "tcl" which is not possible. Thus, are parsing dataset preprocessing applies inserts closures if they are not already present. Note, the same is not true for releases–it is possible, if not common to have unreleased stops as distinct phonemes in non-English languages (i.e. Korean and Chinese)

[3]Using the Levenshtein Distance

[4]In fact, the notion of a word becomes quite difficult to quantify when trying to generalize the model cross-linguistically

[5]We consider edit distance as naive since we are not able to generalize what type of mistakes the model is making and is not normalized to a specific length. For example, a model which produces longer sequences are more likely to have increased edit distances.

[6]A nice property is that normalized PER is normalized between [0, 1]

To inspect cases of especially high Edit Distance or PER, we visualize errors using a Sequence Alignment. This helps us understand what kind of phonemes the model is most prone to.

```
ʃihɛd-dʒɛˠ--darksutʔɪŋgr---isɪ-warʃwaɪɚʔaljɪɚ
|||   |     || |||| |      ||  || || || ||||
ʃih--æd---jɚda-ksut-ɪ---ngɹis-iwa-ʃw-ɪɚ-aljɪ-
  Score=23
```

Figure 1. Sequence alignment for ”She had your dark suit in greasey wash water all year”. The sequence alignment score represents the maximum number of aligned phonemes given the minimum edits to each sequence

For this specific example, we can see how the vowels which differentiate ”ae” vs ”eh” and the extra consonant ”r” in the truth transcription is not picked up by the baseline models. By listening to the audio for this specific example, we can start to find a generalization that the baseline model does not perform well on Dialect Region ”New England”.

6.2. Wav2Vec2IPA

For our trained model, we used Wandb to track the PER and Connectionist Temporal Classification error over training epochs.

First, we start training off at 100 epochs at a learning rate of 10^{-4}

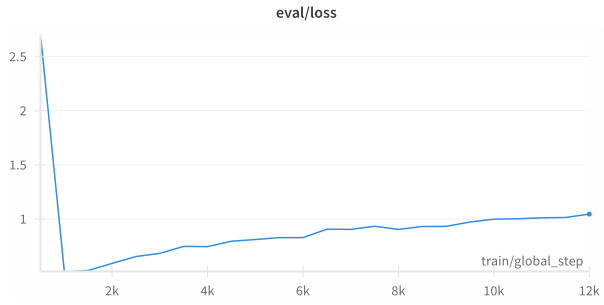

Figure 2. Validation loss vs. Training Steps (12K steps, 400 steps/epoch)

The plot above shows extreme overfitting at training step 1500 which translates to epoch 4.

This is also reflected in the Phoneme Error Rate

To combat this overfitting, we tried to increase the learning rate from 10^{-4} to 10^{-2} to possibly ”jump” out of potential local minima. However, with such a drastic increase in learning rate, the model is underfitting and unable to converge for

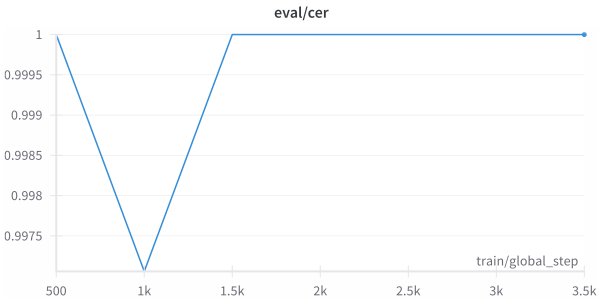

Figure 3. Phoneme Error Rate vs. Training Step

training loss. Above we can further see how the Phoneme Error Rate stays relatively high (close to 1) throughout all training.

7. Discussion

Due to the overfitting in the Wav2Vec2 model, it is clear that his model severely underperformed the baseline model.

Areas for improvement include continuing to train the hyperparameters of the Wav2Vec2 model, inspecting the custom vocabulary created for the IPA tokenizer, and looking at what specific errors the model is consistently making.

In addition to tuning the Learning Rate, we can also look at the test train split.

In addition, since the Wav2Vec2XLSR seems to perform very well on the TIMIT dataset, perhaps continuing to fine-tune this cross-lingual model rather than starting from Wav2Vec2 from scratch could help provide gains in PER almost ”out of the box”. However, in order to increase the number of phonemes in the vocabulary of the model, fine-tuning at perhaps the feature extraction layer might be needed.

8. Conclusion

In conclusion, in this paper, we created a pipeline for translating the traditional TIMIT transcription from 1993 into the modern 2024 IPA transcriptions. From there we evaluated two versions of Wav2Vec2, the baseline model Wav2Vec2XLSR and a fine-tuned Wav2Vec2 model. The fine-tuned Wav2Vec2 model had major overfitting.

Next steps for improving the performance against the baseline include continuing to tune hyperparameters for training while considering starting from the Wav2Vec2XLSR model instead of the Wav2Vec2-Base model.

However, as the need to increase vocabulary size occurs, we

may need to consider updates to the feature encoder as well.

9. Appendix

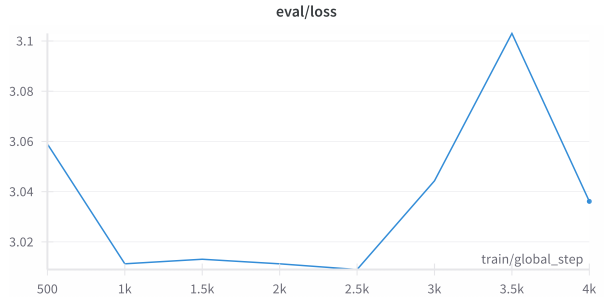

Figure 4. Evaluation Loss vs. Training Step for 10^{-2} Learning rate

References

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477, 2020. URL https://arxiv.org/abs/2006.11477.

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., and Auli, M. Unsupervised cross-lingual representation learning for speech recognition. *CoRR*, abs/2006.13979, 2020. URL https://arxiv.org/abs/2006.13979.