Improving Phoneme Recognition In Cross-lingual Setting Via Universal Phone Recognition

Chitrank Gupta
Department of Computer Science
The University of Texas at Austin
chitrank@utexas.edu

Jordi Ramos
Department of Computer Science
The University of Texas at Austin
rjordi@utexas.edu

Abstract

In this paper, we perform rigorous experiments on top of the frameworks proposed for improving phoneme recognition in cross-lingual settings by [1] and [2]. Phoneme recognition, the task of predicting the pronunciation sequence from the audio, is an important problem in the field of speech technologies. For this project, we investigate phoneme recognition in cross-multi-lingual settings, i.e., training and testing on different sets of many languages. Traditional works on phoneme recognition train a "private" or "shared" model for training languages. The primary disadvantage of "private" models is that these models are of little help when evaluated on out-of-distribution languages. On the other hand, "shared" models, although share phonemes across languages and seem to be more resourceful they perform even worse than "private" models. The reason being, different sounds (phones) in some languages can correspond to the same phoneme and in some other languages not. In this paper we train and analyze models that use this explicit knowledge in the form of phone-to-phoneme mappings before predicting phonemes across languages. Furthermore, we also study frameworks that help us when the test languages consists of phones not seen during the training phase.

1 Introduction

Phoneme recognition (tPR- t is short for *task of*) is a seq-2-seq task in the field of speech technologies. The goal is, given a speech-audio input and the language of that speech, to output the pronunciation in terms of IPA.

tPR is important for various other downstream tasks in both automatic speech recognition and text-to-speech. In the former, once we transcribe speech to its pronunciation, we can then use pronunciation-to-syllables or pronunciation-to-words transducers and further refine the output sequences with language model scores to recognize words and sentences. In closely related textless-transliteration task as well, this task is useful. In text-to-speech, such a tool could be helpful in accent-transfer tasks wherein after identifying the pronunciation in the source language, a text-to-speech can be directly applied on the pronunciation (which is limited to the phonemes of that language, and not phones) to generate speech in target's language. Another most important significance of this task is in speech recognition in non-orthographic languages [1].

The goals of this project are multifold-

1 To get experience in coding, training, and working with multilingual ASR models, especially in cross-lingual settings where we train on one set of languages and test on another set of languages.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Verify the hypotheses presented in several past works on improving phoneme recognition in cross-lingual settings using external allophone database[1] and external phonetics database [2].

For our project, we were also interested in the motivation and the future potential for phoneme recognition in multilingual training and inference. There are around eight thousand languages in the world, most of which do not have any training set and therefore many existing end-to-end approaches are not feasible. Instead of building end-to-end models directly, we can build a pipeline which decomposes the linguistic components such as the one shown in figure 1. Although our work and experiments focus primarily on phoneme recognition, it can be combined with other works to form a powerful end-to-end pipeline for multilingual speech recognition.

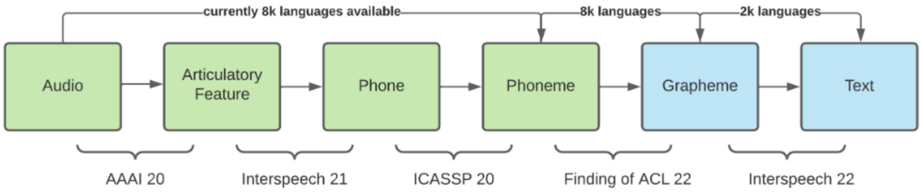

Figure 1: A multilingual speech recognition pipeline. Image adapted from https://www.xinjianl.com/Research/.

In terms of the actual model implementation part of our project, we have implemented the following models:

1 The Shared Phoneme Model [1, 3].

2 The Allosaurus Model [1].

3 The Hierarchical Model [2].

2 Background

2.1 Allophone

The main motivation for our work is to build speech tools benefiting low-resource languages and non-orthographic languages. Our approach is to recognize *phonemes* because they are perceptual units of sound for a given language. Phonemes are closely related with, but are not the same thing as *phones*, which is defined to be the actual sounds made in speech. When doing multilingual phoneme recognition, the concept of *Allophone* is important. Allophones are sets of phones that correspond to a particular phoneme in a given language. This means that distinctions in certain phones may be important in some languages and negligible in others. A good example is shown in figure 2. This example shows that, in English, the two different phones [p] and [p^h] correspond to the same phoneme /p/.

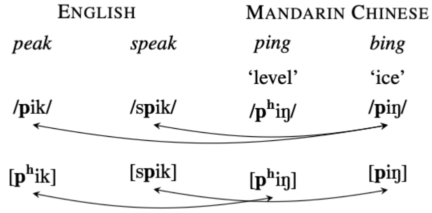

Figure 2: Words, phonemes (slashes), and phones (square brackets)[1].

Due to the allophone set, simple methods such as taking the union of the phoneme sets does not work well when it comes to multilingual phoneme training. From the same example in figure 2, we see that [p] and [$\mathrm{p^h}$] would get assigned to the same phoneme /p/ in English, which would produce poor results if the model wanted to recognize Chinese where the same two phones correspond to two distinct phonemes /p/ and /$\mathrm{p^h}$/.

2.2 Phone-Phoneme Mappings (Allophones)

Suppose there are $|L|$ training languages, and each language L_i has a private phoneme inventory Q_i. Then a *shared phoneme inventory* Q_{sha} is defined to be:

$$Q_{\text{sha}} = \bigcup_{1 \le i \le |L|} Q_i. \tag{1}$$

The allophone set P_q^i maps each phone $p \in P_q^i$ to a phoneme $q \in Q_i$ for language L_i. The *universal phone inventory* P_{uni} is defined as:

$$P_{\text{uni}} = \bigcup_{1 \le i \le |L|} \bigcup_{q \in Q_i} P_q^i. \tag{2}$$

A *signature matrix* $S^i = \{0, 1\}^{|Q_i| \times |P_{\text{uni}}|}$ describes the association between phones and phonemes in each language and is used in some of our models.

2.3 Multilingual Recognition Models

There are four models of interest in our project: the private phoneme model, the shared phoneme model, the *Allosaurus* (**allo**phone **s**ystem of **au**tomatic **r**ecognition for **u**niversal **s**peech) model, and the hierarchical model. These models are shown in figure 3 and figure 4. We have implemented all four models but only performed experiments on the last three due to time constraints. We also wanted to emphasize that we coded each of the models from scratch without ever looking at any code repositories of existing approaches.

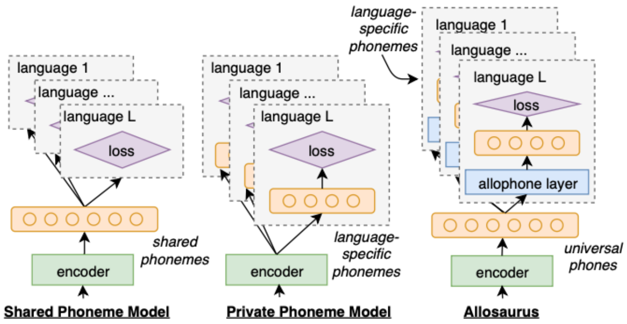

Figure 3: Traditional approaches predict phonemes directly, either for all languages (left) or separately for each language (middle). On the contrary, our approach (right) predicts over a shared phone inventory, then maps into language-specific phonemes with an allophone layer [1].

The private phoneme model performs phoneme classification for each language separately. This model is best used as a baseline model since it is trivial: The approach is less than ideal since it "completely ignores cross-lingual phonetic associations and is not applicable to recognition of new language" [1, 4, 5].

The shared phoneme model performs classification on the shared inventory of all the phonemes from the training languages. This approach "fails to consider the disconnect between phonemes across languages" [1, 3].

The Allosaurus model is a novel method proposed by Li et. al. [1]. The model first computes a universal phone distribution and then uses something called the *allophone layer* that maps the phone distribution into a phoneme distribution for each language. The allophone layer is first initialized with the signature matrix introduced in section 2.2 and has the option to be optimized during the training process. The Allosaurus model is then able to perform multilingual training and inference with an increased performance but has the limitation of not able to handle unseen phones well.

The hierarchical model is another novel method [2] whose architecture is shown in figure 4. This model is a direct response to the limitation of the Allosaurus model, wherein the model cannot predict a score for a phone not seen during the training time. To mitigate the same, it first decomposes phones into articulatory attributes (obtained from an external database) and computes a dense representation of that phone from the embeddings of those articulatory attributes. The rest of the model is rather similar to the Allosaurus model.

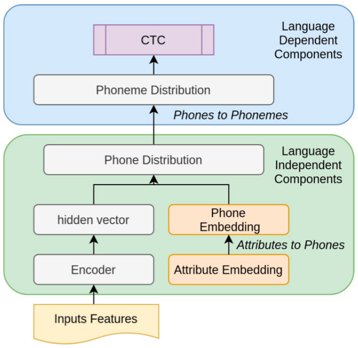

Figure 4: The architecture of the hierarchical model. We first compose the phone embeddings from their attribute embeddings. Then we compute the phone distributions using the embeddings and the hidden vector from the encoder, Next, the language-independent phones are transformed into language-dependent phonemes with the help of allophone mappings, which would finally be optimized by the loss (CTC) function [2].

3 Related Works

Li et. al. (2020) [6] proposes a zero-shot learning multilingual acoustic model. It decomposes phonemes into articulatory attributes and computes the phoneme distributions based on that. They evaluated 13 training languages and 7 unseen test languages and found a 7.7% increase in phoneme error rate (PER) on average compared to the baseline model.

There are two main approaches to tPR: the shared phoneme approach calculates a distribution over the universal phonemes [3, 7, 8, 9, 10]. This approach is popular and has been around for more than two decades. The private phoneme approach treats the training of each language separately and calculates a phoneme distribution for each [11, 12, 13]. Besides being used as a baseline model, others have tried to apply transfer learning on the private phoneme models for training across different languages.

The Allosaurus model [1] has experimented on multilingual ASR over 11 languages, and the model improves testing performance by 2% phoneme error rate absolute in low-resource conditions. Experiments on two indigenous languages show that the model achieves phone accuracy improvements of more than 17%.

The hierarchical model [2] is able to recognize phones that do not appear in the training set. It is evaluated on 47 unseen languages and it outperforms baseline models by 13.1% PER.

Grapheme-to-Phoneme (G2P) models [14, 15, 16, 17] are useful to map phonemes to graphemes. To perform a full end-to-end multilingual speech recognition task, G2P models are indispensable. In the particular case of multilingual G2P models, a phylogenetic tree [14] can be used to identify unseen languages by examining the top-k nearest languages in the training set. This approach was tested on over 600 unseen languages and outperformed the baseline significantly [18].

4 Implemented Model Description

To differentiate our implementation from the models described in other papers, we will refer to them as Shared (the shared phoneme model), Allsrs (the Allosaurus model), and Heir (the hierarchical model).

We used the same encoder for all the models we implemented. Starting from the raw audio file in ".wav" format, we re-sampled the audio files to be 16,000 samples per second if the sample rate is higher than that so that all the audio files have the same sample rate. We then used the MFCC compute feature from Python speech features with 40 dimension MFCCs and 80 number of filters in the filter bank. To work better with the BiLSTM model, we also padded our input features with 0s across the batch so each input feature within the same batch has the same dimension. We then fed the padded input into a bi-directional LSTM. The choice of the hyper-parameters of the BiLSTM is further discussed in section 6.

Although we did not perform any experiments on the private phoneme model, we still attempted to build the model. For the attempt, we used a 1-layer BLSTM with 12 hidden states. The UCLA dataset has about 94 languages. When we attempted to create a different BiLSTM for all these languages, our computer crashed. We choose not to investigate this problem further due to the time constraint on this project

For the Shared model, we first counted the number of all the phonemes available in the training language. We then used this number to create a linear layer that maps the output of the BiLSTM to a universal phoneme distribution. We then use the CTC (Connectionist Temporal Classification) [19] loss for back-propagation.

For the Allsrs model, we created a linear layer that maps the output of the BiLSTM to the number of universal phones. Depending on the training language of the current batch, we could perform a matrix multiplication with the corresponding allophone matrix W^i that maps from the universal phone to the phoneme of that language. The allophone matrix was initialized to be the signature matrix, and the preparation of the signature matrix is further discussed in section 5. After this, we used the CTC loss function to compute the loss and perform back-propagation. Note that we also have the option to train our allophone matrix, in which case our loss function can be described as:

$$\mathcal{L} = \sum_{1 \leq i \leq |L|} (\mathcal{L}^i_{ctc} + \alpha \|W^i - S^i\|_2^2), \tag{3}$$

with α as a hyper-parameter value whose choice is discussed in section 6.

The Heir model is similar to the Allsrs model. For this model, we further decompose phones into phonological articulatory attributes. We then assign each attribute an attribute embedding to encode its information. We can then calculate the phone embedding by summing up its corresponding attribute embeddings, which can be fine-tuned in the training process. We then take the inner product of the phone embedding and the output of the BLSTM to compute the phone distributions. After this, the Heir model computes the phoneme distribution using the allophone layer just like the Allsrs model does.

5 Training Details

For our experiments, we used two datasets: the Common Voice corpus [20] and the UCLA dataset [21].

The Common Voice corpus is an open-source multilingual collection of transcribed speech. It uses crowd-sourcing for both data collection and data validation. We sampled over 28,806 audio files with 40 hours worth of data in over 18 different languages. To make sure that we only use the best quality data, we only sampled data with two or more up-votes from the reviewers. Each of these utterances was long (around 5 seconds) and had long leading and trailing silence (for handling this we especially add a "silence" character, different from the CTC-Loss's "blank" character to the ends of the phoneme sequence). The average phoneme-sequence length was 38.65 ($\pm$19.2).

The UCLA dataset is from the UCLA phonetics lab archive prepared by the UCLA Department of Linguistics. It contains one (linguistic-)word-long recording for about 94 languages. Since utterances are just a word long, the average phoneme sequence length is about 4.16 ($\pm$1.62).

		CommonVoice (Training Lang)				CommonVoice (Non-Training Lang)				UCLA			
		TrainSig False		TrainSig True		TrainSig False		TrainSig True		TrainSig False		TrainSig True	
N	h	**SmAl**	LgAl	**SmAl**	LgAl	**SmAl**	LgAl	SmAl	LgAl	SmAl	**LgAl**	**SmAl**	LgAl
6	512	**36.8303**	37.7871	36.9154	37.4238	**35.3526**	37.0802	36.2347	36.2651	4.3906	**3.9808**	4.1763	4.0537
6	64	**32.5777**	33.6182	37.9872	37.0510	**32.1009**	33.4467	36.7590	33.3343	7.2015	5.3825	**4.3046**	9.3492
4	512	**34.6628**	36.4390	36.7811	37.7615	**34.1453**	35.9937	36.1094	35.0534	4.8060	4.3830	**4.3373**	6.9446
4	**256**	**32.4567**	33.9200	35.5827	35.6241	**32.0611**	33.3610	35.4262	45.2911	7.0735	6.8980	**4.4838**	11.5545

Table 1: Hyperparameter study on Allsrs model (PER results on Test set). The most important takeaways from this study are, the best performing (N,h) for Commonvoice is (4,256) while the worst performing is (6,512). Exactly opposite is true for UCLA dataset. This might be because of the distribution difference between these two datasets. CommonVoice Corpus has longer phoneme sequences and somehow we found that the length of the CTC-normalized output of our models depend on (N,h).

When processing the training data, we implemented our own customized sampler that works with PyTorch's Dataloader to sample utterances corresponding to only one language at a time. This enables us to direct the pipeline to the language-specific allophone matrix and phoneme distribution when needed and saves a lot of time.

6 Results and Conclusions

6.1 Training Details

Since the basic Allsrs model has the most similar architecture to all other baselines and Heir , we chose to conduct hyperparameter tuning of the LSTM's hyper-parameters on the basic Allsrs model. Further, for all our experiments as well as the final results, we only trained on ~ 1 hr of training data and validated ~ 1 hr of validation data. However, for final testing of the model, we used ~ 20 hr of test data and ~ 1 hr of UCLA dataset (all datasets are described in the Dataset section). All models were trained using AdamW[22] Optimizer with a learning rate 1e-3 for 1000 epochs. For final evaluation of the test data, the model with the best average PER (averaged across utterances) on dev datasplit is used.

6.2 Allsrs

As explained before, we first present the HP-tuning study done on Allsrs model. The complete list of HPs is LSTM's number of layers (N), LSTM's hidden dimension (h), Regularization coefficient of eq 3 (α), signature matrix initialization, and finally whether or not to train the signature matrix during training at all. The last two hyper-parameters need further explanation here, as follows.

1 By signature matrix initialization, we actually mean what all knowledge sources were used to curate the set of allophone mappings. For our study, we used two external knowledge sources– one obtained from Transphone[18] and the other from Phoible[23]. If we use both of these external knowledge bases to initialize the signature matrix, then that setting is called Larger Allophone setting or LaAl setting. On the other hand, if we just initialize with the transphone library, then we call that setting Smaller Allophone setting or SmAl.

2 By training the signature matrix, we are referring to whether or not we train the signature matrix. Accordingly we name them SigTr (Signature matrix, *true* trainable) and SigFl (Signature matrix, *false* trainable).

The hyperparameter study done on Allsrs model is shown in Table 1. The most important takeaways from this study are as follows-

1 For CommonVoice Corpus dataset, the best performing (N,h) is (4,256). (6,64) follows very closely, as expected since these two settings have a similar number of parameters. The worst performing (N,h) setting is (6,512), which contains the most number of parameters. This could be attributed to the fact that the amount of training data is small and hence models with a large number of parameters end up being under fitted.

2 For the UCLA dataset, exactly the opposite observation is true. The best-performing setting happens to be (6,512) and the worst-performing (4,256). We empirically found out that the

		CommonVoice (Training Lang)		CommonVoice (Non-Training Lang)		UCLA	
N	h	$\mathrm{Shared} - \mathrm{Theirs}$	$\mathrm{Shared} - \mathrm{Ours}$	$\mathrm{Shared} - \mathrm{Theirs}$	$\mathrm{Shared} - \mathrm{Ours}$	$\mathrm{Shared} - \mathrm{Theirs}$	$\mathrm{Shared} - \mathrm{Ours}$
6	512	**36.2804**	36.8996	**35.4810**	36.1737	4.1666	**4.1505**
6	64	**34.4966**	35.1038	**33.8137**	34.6890	**6.1516**	6.2387

Table 2: HP study for shared model. Surprisingly the $\mathrm{Shared} - \mathrm{Ours}$ baseline model proposed by us ends up performing poorer, although intuition suggests otherwise. This could be due to the fact that by not allowing the artificial masking in $\mathrm{Shared} - \mathrm{Theirs}$, the model is challenged better during training and hence generalizes better. On the other hand, the comparison of (6,64) vs (6,512) is consistent with Allsrs (Tab 1, i.e., here too there is no clear winner.

		CommonVoice (Training Lang)		CommonVoice (Non-Training Lang)		UCLA	
N	h	Heir_SigTr_SmAl	Heir_SigFl_LaAl	Heir_SigTr_SmAl	Heir_SigFl_LaAl	Heir_SigTr_SmAl	Heir_SigFl_LaAl
6	512	33.6481	**32.7827**	33.9565	**33.1976**	**6.5707**	6.8822
6	64	39.1218	**32.6418**	37.4025	**33.1947**	**4.9621**	5.2320

Table 3: HP study on Heir model. The intuition behind using these selective HPs is described in sec 6. As we can see, using a large allophone database and keeping the signature matrix fixed achieves better results consistently for CommonVoice Corpus. Although the opposite is true for the UCLA dataset. One can verify similar results for the original Allsrs model as well (Table 1). Moreover, we find out, with 6 layers in LSTM, having 64 hidden dimension sizes works better.

LSTM size affected the length of CTC-normalized prediction length and a larger number of parameters usually predicted smaller length and UCLA dataset utterances indeed are on average of shorter duration.

3 Comparing the sizes of the allophone dataset, we found out that having a smaller allophone database consistently outperformed the larger allophone counterparts. Although in the case of the UCLA dataset, we found out that the signature matrix is kept non-trainable and having a larger allophone database proves to be better.

4 Finally, we found out that for the CommonVoice corpus, keeping the signature matrix non-trainable is better, but for UCLA dataset, training the signature matrix is better.

From this extensive ablation study, we conclude that there is no best hyperparameter setting for Allsrs. However, as far as (N,h) hyperparameters are considered, (6,64) is a good enough winner. However, we think we can reach a stronger conclusion by training on a larger training dataset and for more iterations. One remaining HP we didn't study was α. Unfortunately, we weren't able to finish our studies with α and just used whatever α, [1] had used, which is just 10.0.

We then conducted HP study on other models. Instead of doing HP study all over again, we chose to use selective HPs from the previous ablation study. Specifically, we chose two pairs for (N, h) which are (6,512) and (6,64). The reason we specifically chose these two pairs is because the former one achieved much poorer results across datasets while the latter one achieved much better results across datasets. Moreover, by keeping N the same we can also do controlled study on h. The results and conclusions for Shared and Heir models are in Table 2 and 2 respectively.

Let's now do a final comparison across model architectures. For (N,h) to be (6,512) setting, surprisingly, Heir model works the best for the CommonVoice Corpus while Allsrs model works the best for the UCLA dataset. On the other hand, in (N,h) equal to (6,64) setting, Allsrs model works the best for both CommonVoice Corpus and UCLA dataset. However, Allsrs is still not an overall clear winner since its performance is not the best across datasets in the same setting of (N,h). But one clear conclusion is that use of an external allophone database to differentiate the meaning of phonemes across languages proves to be better.

Finally, let's also perform a quick analysis on the embeddings learned by the Heir model (specifically Heir_SigTr_SmAl (6,64) variant). The results are shown in Figure 5. As we can observe, the embeddings learned by the Heir model are of pretty good quality as demonstrated by the phonetic clusters– e.g., nasals, stops, fricatives, and back vowels are nicely clustered together. Moreover, consonants and vowels are clearly clustered apart from each other.

7 Future Works

For future work, we wish to deploy one of the G2P models described in section 3 to translate phonemes into graphemes. We hope to see that with the integration of G2P, whether we can still

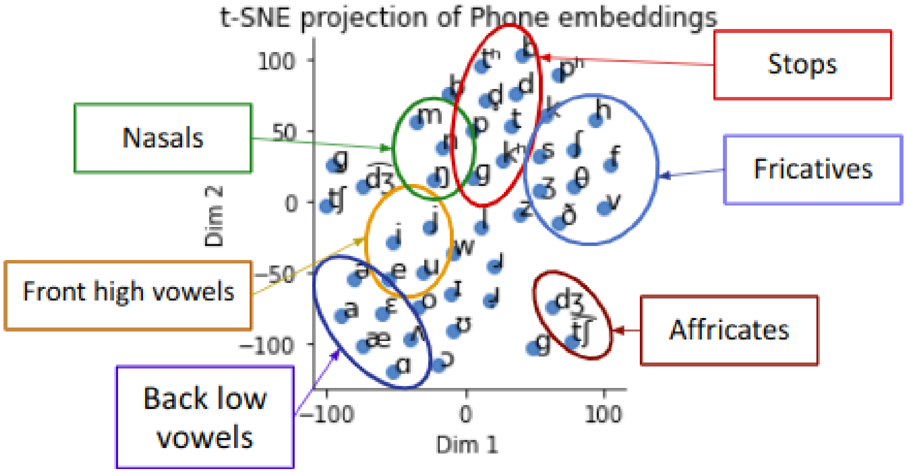

Figure 5: t-SNE projection of phone embeddings learned from Heir model

recognize unseen phones and languages. Another direction for future work is to perform speech recognition for languages without Audio. The motivation behind this is that a lot of low-resource languages do not have audio training data available. This work has already been explored by Li et al. (2022) [24], where they used raw text datasets or n-gram statistics in combination with language models to perform the recognition. One more interesting line of work is to perform multilingual recognition when there is more than one language on the same utterance. We were able to find one such paper [25] where Feng et al. uses BERT to learn sentence embeddings for semantic similarity across languages.

References

[1] Xinjian Li et al. “Universal phone recognition with a multilingual allophone system”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 8249–8253.

[2] Xinjian Li et al. “Hierarchical Phone Recognition with Compositional Phonetics.” In: *Interspeech*. 2021, pp. 2461–2465.

[3] Jessica A.F. Thompson et al. “How Transferable Are Features in Convolutional Neural Network Acoustic Models across Languages?” In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 2827–2831. DOI: `10.1109/ICASSP.2019.8683043`.

[4] Siddharth Dalmia et al. “Sequence-Based Multi-Lingual Low Resource Speech Recognition”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), pp. 4909–4913.

[5] Jui-Ting Huang et al. “Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 7304–7308. DOI: `10.1109/ICASSP.2013.6639081`.

[6] Xinjian Li et al. “Towards zero-shot learning for automatic phonemic transcription”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 8261–8268.

[7] Hui Lin et al. “A study on multilingual acoustic modeling for large vocabulary ASR”. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2009, pp. 4333–4336.

[8] P Cohen et al. “Towards a universal speech recognizer for multiple languages”. In: *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*. IEEE. 1997, pp. 591–598.

[9] Tanja Schultz and Alex Waibel. "Fast bootstrapping of LVCSR systems with multilingual phoneme sets". In: *Fifth European Conference on Speech Communication and Technology*. Citeseer. 1997.

[10] Tanja Schultz and Alex Waibel. "Language-independent and language-adaptive acoustic modeling for speech recognition". In: *Speech Communication* 35.1-2 (2001), pp. 31–51.

[11] Siddharth Dalmia et al. "Sequence-based multi-lingual low resource speech recognition". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 4909–4913.

[12] Jui-Ting Huang et al. "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 7304–7308.

[13] Xinjian Li et al. "Multilingual speech recognition with corpus relatedness sampling". In: *arXiv preprint arXiv:1908.01060* (2019).

[14] Xinjian Li et al. "Zero-shot Learning for Grapheme to Phoneme Conversion with Language Ensemble". In: *Findings of the Association for Computational Linguistics: ACL 2022*. 2022, pp. 2106–2115.

[15] Xinjian Li et al. "Phone Inventories and Recognition for Every Language". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. 2022, pp. 1061–1067.

[16] Sercan Ö Arık et al. "Deep voice: Real-time neural text-to-speech". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 195–204.

[17] Yajie Miao, Mohammad Gowayyed, and Florian Metze. "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding". In: *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE. 2015, pp. 167–174.

[18] Xinjian Li. *transphone*. URL: `https://github.com/xinjli/transphone` (visited on 11/22/2022).

[19] Alex Graves et al. "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 369–376. ISBN: 1595933832. DOI: `10.1145/1143844.1143891`. URL: `https://doi.org/10.1145/1143844.1143891`.

[20] Rosana Ardila et al. "Common voice: A massively-multilingual speech corpus". In: *arXiv preprint arXiv:1912.06670* (2019).

[21] CA: UCLA Department of Linguistics. The UCLA Phonetics Lab Archive. Los Angeles. *transphone*. URL: `http://archive.phonetics.ucla.edu/`.

[22] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: `https://openreview.net/forum?id=Bkg6RiCqY7`.

[23] Steven Moran and Daniel McCloy, eds. *PHOIBLE 2.0*. Jena: Max Planck Institute for the Science of Human History, 2019. URL: `https://phoible.org/`.

[24] Xinjian Li et al. "ASR2K: Speech Recognition for Around 2000 Languages without Audio". In: *arXiv preprint arXiv:2209.02842* (2022).

[25] Fangxiaoyu Feng et al. "Language-agnostic bert sentence embedding". In: *arXiv preprint arXiv:2007.01852* (2020).