# Low-Resource Speech Recognition for Thousands of Languages

**Xinjian Li**
**CMU-LTI-23-006**

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

**Thesis Committee:**
Shinji Watanabe **(chair)**
Alan W Black **(co-chair)**
David R Mortensen
Florian Metze
Patrick Littell

*Submitted in partial fulfillment of the requirements*
*for the degree of Doctor of Philosophy*
*In Language and Information Technologies*

# Abstract

Recently, the performance of speech recognition has witnessed rapid improvement due to modern architectures. Those models typically require thousands of hours of training data for the target language. However, there are around 8000 languages in the world, the majority of which do not have any audio or text dataset, which significantly restricts the scope of target languages.

This thesis attempts to expand the target languages of speech recognition to more than thousands of languages by reducing the dataset requirement. In particular, we present a speech recognition pipeline that does not require any audio for the target language. The only assumption is that we have access to raw text datasets or a set of n-gram statistics for the target language. In the minimalist assumption, we only employ the lexicon from the target language. Our speech pipeline consists of three components: *acoustic model*, *pronunciation model*, and *language model*. Unlike the standard pipeline, our acoustic and pronunciation models use multilingual models without any supervision of the target language.

The first part of this thesis discusses the hierarchical acoustic model which can be decomposed into two submodules: the *universal phone recognition model* recognizes language-independent phones using phonological articulatory features, and subsequently the *allophone model* mapping phones into language-dependent phonemes. In the second part, we turn our focus on the pronunciation model and language model. We develop a zero-shot learning grapheme-to-phoneme (G2P) model which approximates the target language using nearest languages from the phylogenetic tree. G2P model serves as a pronunciation model. The language model can be built using n-gram statistics or the raw text dataset. We build our language model by combining it with a large endangered languages n-gram database and a lexicon database. In the last part, we introduce two databases we use in the pipeline and relevant alignment applications. Using the proposed pipeline and datasets, we build speech recognition systems for 6185 languages, which significantly expands the scope of target languages in speech recognition.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Low-Resource and Zero-Resource Languages

With the development of deep neural networks, there is growing interest in applying deep neural network models to speech recognition (Amodei et al., 2016; Chiu et al., 2018; Chan et al., 2016). Those deep models, however, are restricted to languages with a large amount of training set such as English and Mandarin (Godfrey et al., 1992; Panayotov et al., 2015), therefore, they are not available for most languages in the world. Additionally, the majority of the languages in the world have never been written (Coulmas, 2013), it has been unclear how to develop speech recognition systems for those languages.

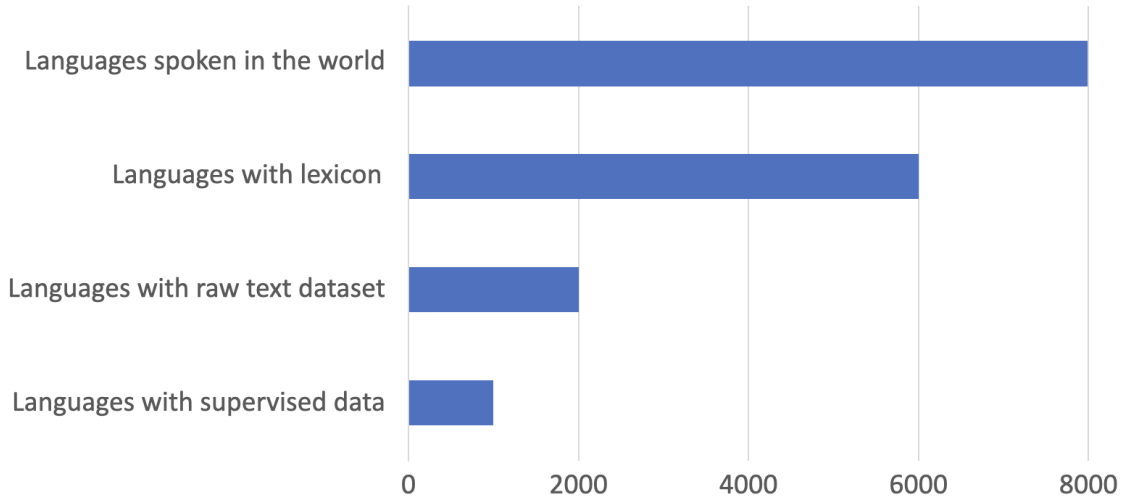



Figure 1.1: Statistics of the number of languages from the speech resources perspective

Ethonologue, which is one of the most extensive catalogs of world's languages, has estimated

the total number of languages is about $7000 \sim 8000$ (Lewis, 2009). However, only a small portion of those languages have audio or language resources, not to mention a clean parallel speech training corpus. Figure.1.1 shows some statistics related to the number of languages classified by their speech resource conditions. Out of the total 8000 languages, only half of the languages have a written form and can be written down in some form. However, only around 2000 languages are estimated to have source web text data available online (Scannell, 2007; Bapna et al., 2022). As of 2009, even Bible has just been translated into only 2508 languages (Lewis, 2009). The largest supervised dataset might also be the Bible audio collection, for example, the Bible.org website contains audios for around 1000 languages (Black, 2019; Pratap et al., 2023). Supervised speech recognition typically requires the paired audio and text dataset, therefore, the upper bound for existing supervised approaches is around 1000 languages. For example, Google's cloud Speech-to-Text service provides 120 languages and their variants, the Whisper model also covers nearly 100 languages (Radford et al., 2022). Meta's latest model has increased the number of languages to around 1000 languages by using the collected Bible dataset. (Pratap et al., 2023). However, 1000 is only a fraction of all languages. the main focus of this thesis is to propose speech recognition systems for those low-resource and zero-resource languages.

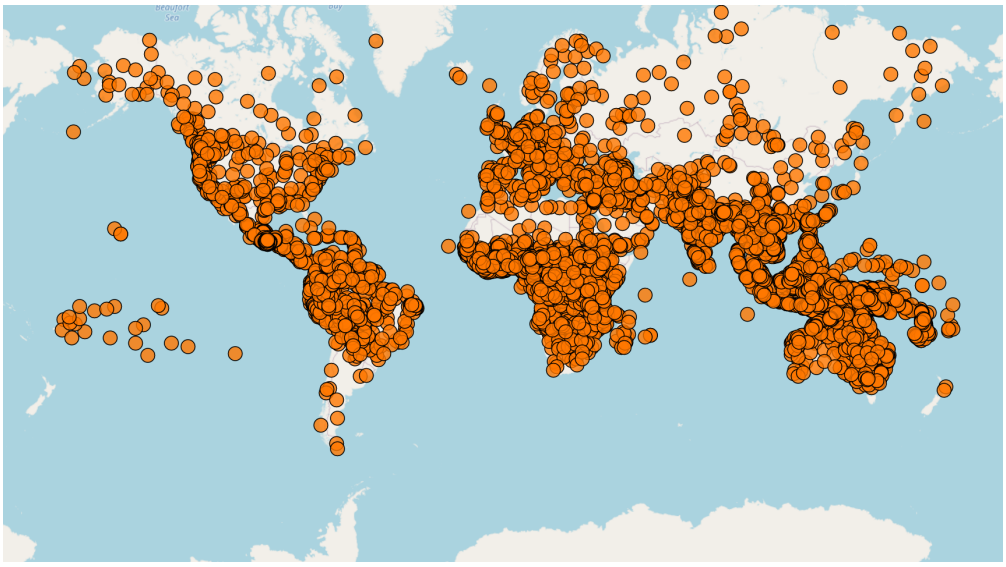


Figure 1.2: World Map showing all languages in Glottolog

Not only the number of available languages is limited, but even among the high-resource languages, it is not distributed geologically evenly across the world. Figure.1.2 shows the geological distribution of all 8000 languages in the world, each orange circle denotes a language. In contrast, Figure 1.3 shows the distribution of languages with pretrained models in ESPnet (Watanabe et al., 2018). It is clear the most of the available pretrained models are concentrated in the Europe and

East Asian areas, but it cover a few languages distributed in the American and African continents. Papua New Guinea is known as one of the most linguistically diverse countries in the world, it has 854 local languages but none of them are covered in Figure 1.3.

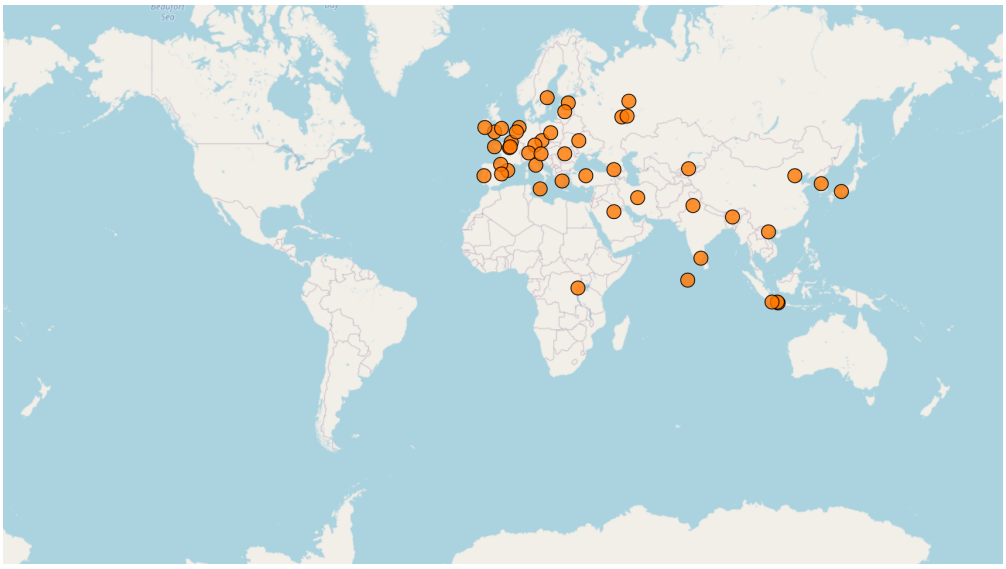


Figure 1.3: World Map showing all languages whose recipes are available in ESPnet

The goal of this thesis is to build speech recognition for languages with few audio datasets and/or text datasets. The main target of this thesis is the 2nd category (languages with text data available online) and the 3rd categories (languages with lexicon) in Figure.1.4. We propose an approach to build speech recognition systems for languages the 2nd category (around 2000 languages) and extends it to the 3rd category (around 6000 languages)

## 1.2 Background

Most speech recognition approaches can be classified into one of several groups depending on their data requirements. The most common group has access to the paired supervised dataset

$$\mathcal{D}_{\text{supervised}} = \{(X_i, Y_i)\}_{i=1}^{N} \tag{1.1}$$

, where $(X, Y)$ is a paired audio and text of an utterance. If the size $N$ of the dataset is large enough, various end-to-end models can be trained using CTC, ASG, seq2seq, RNN Transducer, and other objectives (Graves et al., 2006; Collobert et al., 2016; Graves et al., 2013; Sutskever et al., 2014). If the size is small, then it would be a low-resource speech recognition in which some acoustic knowledge should be transferred from high-resource languages (Li et al., 2019a; Xu et al.,

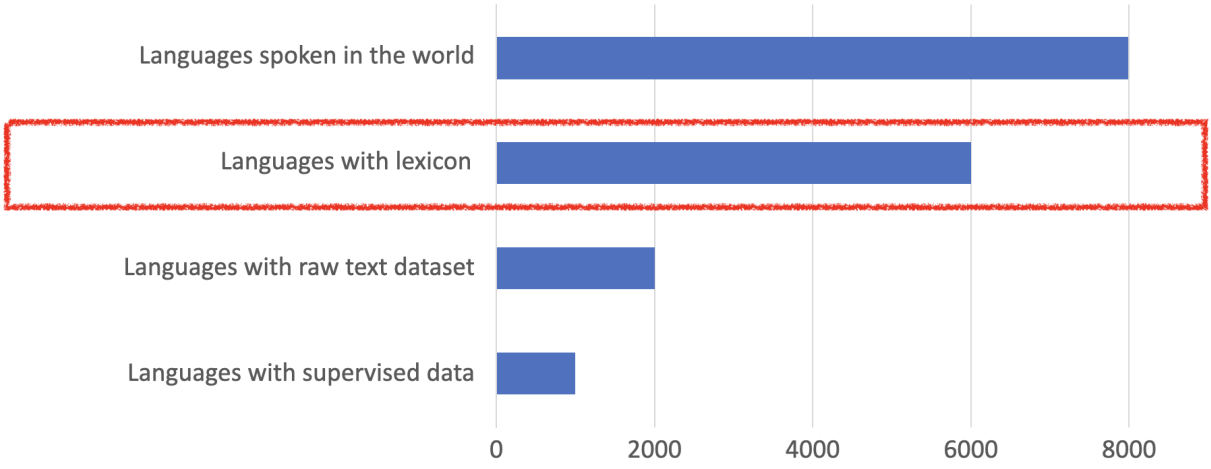



Figure 1.4: The scope of target languages addressed in this thesis

2020). Self-supervised training takes advantage of another large raw speech dataset $\{X_j\}$ to learn hidden representations of speech signals, those representations are useful to the supervised tasks and can reduce the amount of the paired dataset (Baevski et al., 2020; Hsu et al., 2021). The semi-supervised learning approach also leverages unlabeled speech datasets or text datasets to augment the supervision set (Veselý et al., 2017; Synnaeve et al., 2019; Rosenberg et al., 2019).

$$\mathcal{D}_{\text{semi\_supervised}} = (\{(X_i, Y_i)\}_{i=1}^{N}, \{X'_j\}_{j=1}^{J}) \tag{1.2}$$

In the semi-supervised learning, a supervised model is first trained using the paired supervised dataset $\{(X_i, Y_i)\}_{i=1}^{N}$ and then transcribes another unlabeled speech dataset $\{X'_j\}_{j=1}^{J}$ into a text dataset $\{\hat{Y}\}_{j=1}^{J}$. They augment the supervised dataset with pseudo-label dataset $\{(X'_j, \hat{Y}_j)\}_{j=1}^{J}$ to train a better supervised model.

Recently, unsupervised speech recognition attempts to target the dataset

$$\mathcal{D}_{\text{unsupervised}} = (\{X_i\}_{i=1}^{I}, \{Y_j\}_{j=1}^{J}) \tag{1.3}$$

, where we have access to an unlabeled raw audio set $\{X_i\}_{i=1}^{I}$ and a raw text dataset $\{Y_j\}_{j=1}^{J}$ (Baevski et al., 2021). The audio and text do not need to be aligned with each other and size of two datasets $I, J$ do not need to be same. A generator model is jointly trained with a discriminator model. The generator model attempts to translate audio into phonemes, while the discriminator model attempts to distinguish between phonemes transliterated from text and phonemes recognized from the generator. The disadvantage of this direction is that the model could only recognize phonemes instead of words and it requires a phonemizer (pronunciation model) for the target language, which would not be available for most languages. Another related direction is unsupervised speech unit discov-

ery (Chorowski et al., 2019; Tjandra et al., 2019), which is similar to the self-supervised learning approach and attempts to discover phone units from audios

$$\mathcal{D}_{\text{unit\_discovery}} = \{X_i\}_{i=1}^{I} \tag{1.4}$$

This group of approaches, however, cannot emit explicit phonemes or words as it does not have knowledge of the lexicon and language model for the target language.

In this thesis, we propose a new paradigm to focus on the text-only dataset

$$\mathcal{D}_{\text{proposed}} = \{Y_j\}_{j=1}^{J} \tag{1.5}$$

While all the previous groups require some amount of audio datasets $\{X_i\}$ (paired or unpaired) for the word recognition of the target language, we argue this requirement can be relaxed to some extent. In the minimalist setting, we only assume the lexicon for the target language as the text-only dataset.

## 1.3 Approach

### 1.3.1 Motivation

As mentioned in the first section, most of the languages do not have any training set (and even test set), it is impossible to create any end-to-end models directly. Instead of building the end-to-end models, we propose to model multiple linguistic units (e.g: phonemes, graphemes) explicitly and decompose the entire model into a sequence of separate models, each model represents a transformation from one linguistic unit to another linguistic unit. We briefly introduce all linguistic units or components we are using throughout this thesis.

1. **audio**: input speech into the pipeline. Our audio is typically encoded as a *WAV* format file with 1 channel, 16k Hz frequency and 16-bit precision.

2. **phonological feature (articulatory features)**: a set of discrete features characterizing how is each phone produced (i.e. manner of articulation) and where does it get produced (i.e. place of articulatory gestures).

3. **phone**: language-independent phone units. It is the *narrow transcription* of the speech signal which shows more phonetic detail by using more specific symbols.

4. **phoneme**: language-dependent contrastive phone units. It is the *broad transcription* of the speech signal which uses simpler and more abstract symbols. Different languages tend to use different sets of phonemes.

5. **grapheme**: language-dependent contrastive units in the writing systems. For example, we consider the Latin alphabets as the grapheme set for written English.

6. **text**: the final output of the pipeline.

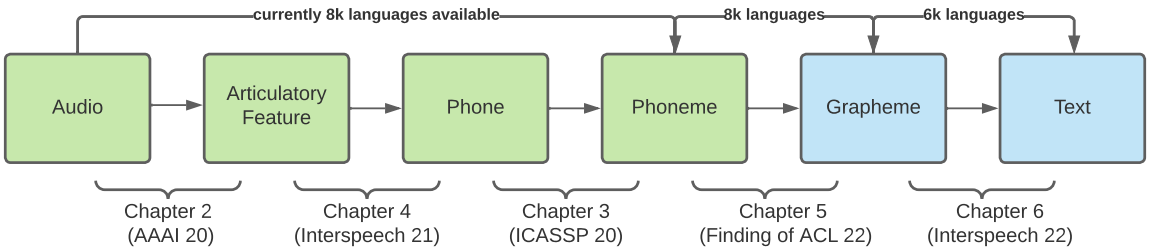



Figure 1.5: The overall architecture proposed in this thesis. The left four green components illustrate the acoustic model, which is covered in Part I. The right two components are corresponding to the language model, which is the main topic of the Part II. Every arrow between two components indicates the chapter and its published or submitted paper.

Figure 1.8 shows the overall architecture proposed in this thesis: the left part shown in green color is the acoustic model, in which we transform the raw audio into language dependent phonemes. The right part is the language model which translates the phoneme sequence into the text form. Within each model, we further decompose them into several submodules and represent multiple linguistic units (e.g: phonological features, phones) inside them.

This direction has many advantages over the end-to-end models:

1. some components are relatively less dependent on the training resources (e.g: texts are easier to obtain than audios)

2. some of them are already well-defined by linguists (e.g: phonetics, phonology), those domain knowledge can be incorporated into the model through some Bayesian frameworks.

3. Sharing information across languages is easier (e.g: English and German shares many common phones, we can use the English inventory as the German inventory in case the German inventory is not available)

Following this motivation, we divide our pipeline into the *acoustic model*, *pronunciation model* and *language model*. The joint probability over speech audio $X$ and speech text $Y$ can be factorized as

$$p_\theta(X, Y) = \sum_P p_{\text{am}}(X|P)p_{\text{pm}}(P|Y)p_{\text{lm}}(Y) \tag{1.6}$$

, where $P$ is the phoneme sequence corresponding to the text $Y$. The pronunciation model $p_{\text{pm}}$ is typically modeled as a deterministic function $\delta_{\text{pm}}$. In our pipeline, we assume that we have access to some text datasets or equivalent statistics for the target language as our main focus is the 3rd category in Figure.1.4. Using those datasets, we can directly create the language model $p_{\text{lm}}(Y)$. However, both the acoustic model and pronunciation model cannot be built from the text datasets and we need to approximate those models using zero-shot learning or transfer learning from other high resource languages, therefore we denote $\hat{p}_{\text{am}}, \hat{\delta}_{\text{pm}}$ for the approximated acoustic model and pronunciation model. The previous factorization can be approximated by

$$p_\theta(X, Y) \approx \hat{p}_{\text{am}}(X|\hat{P})p_{\text{lm}}(Y) \tag{1.7}$$

where $\hat{P} = \hat{\delta}_{\text{pm}}(Y)$ is the approximated phonemes. We shortly describe each of those three models in the following subsection. The details of the acoustic model are covered in Part I of this thesis, and the pronunciation model and the language model are the main focus of the Part II.

### 1.3.2 Acoustic Model

The acoustic model is to recognize phonemes or compute the distribution of phonemes conditioned on the input audio $p_{\text{am}}(P|X)$. The major hurdle of training acoustic models for low-resource languages is the lack of large supervised datasets because the acoustic model depends on the target language and phonemes $P$ are language-dependent units. However, as mentioned in the previous section, most of the languages in the world do not have those supervised datasets, therefore we cannot directly learn the acoustic model for those languages. This thesis proposes an architecture to solve this problem without relying on any audio dataset for the target language. Our proposed model attempts to further divide the acoustic model into two part as shown in Figure 1.6: the upper part is the universal phone recognition model, which is a language-independent model. We introduce it in Chapter 2. The lower part is the allophone mapping model, which is a language-dependent model. We cover this model in Chapter 3. Chapter 4 combines those two models into a single acoustic model.

This architecture attempts to decompose the phoneme recognition task into two subtasks: it first recognizes language-independent phone units from the audio (universal phone recognition task) , then the phone units are transformed into the language-dependent phonemes (allophone mapping task). Essentially, this architecture represents the acoustic model as follows:

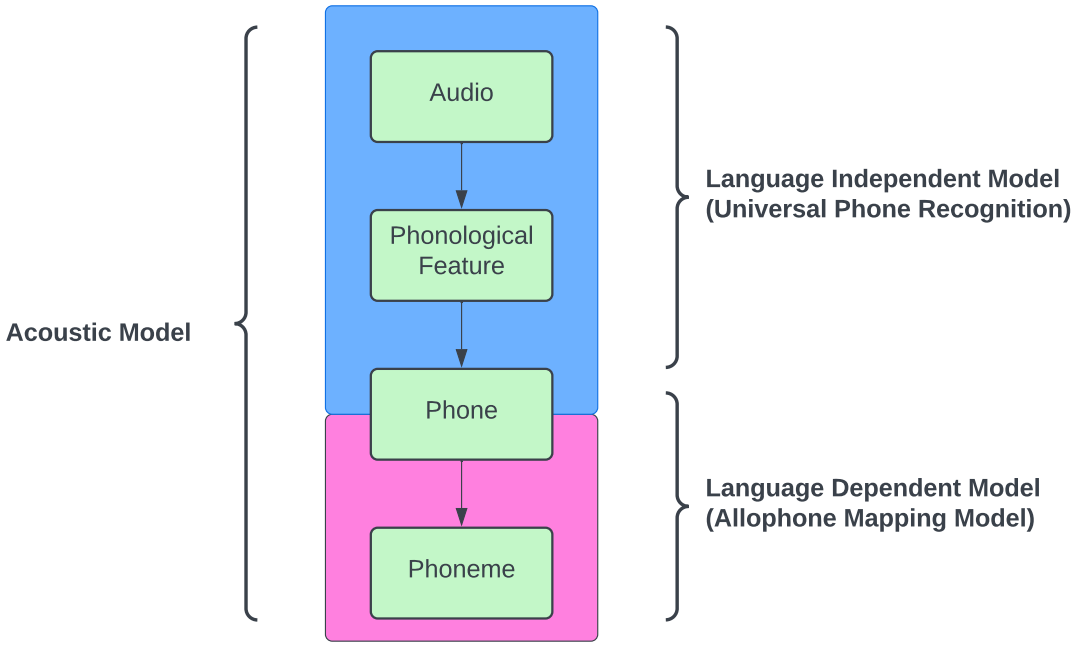



Figure 1.6: The acoustic model architecture. It consists of two parts: language-independent model and language-dependent model.

$$\hat{p}_{\text{am}}(P|X) = \sum_{Q} p_{\text{lang}}(P|Q)p_{\text{uni}}(Q|X) \tag{1.8}$$

, where $p_{\text{uni}}(Q|X)$ is a language-independent universal phone recognition model, recognizing physical-level phone units $Q$ from the speech audio $X$. The language-dependent architecture $p_{\text{lang}}(P|Q)$ is to encode how each physical phone should be mapped to a language-dependent phoneme. The relation between phones and phonemes is called an *allophone*, which is usually encoded as a 1-$n$ deterministic function annotated by phonologists for each language. The mapping is easier to obtain than the supervised dataset for low resource languages. We discuss this allophone mapping model in Chapter 3.

In contrast, the universal phone recognition model $p_{\text{uni}}(Q|X)$ does not have any dependency on the target language, therefore it can be trained using high-resource languages such as English and Mandarin. However, there exists one more issue with this model. The size of the phone inventory $|Q|$ is typically very large, we estimate there are $\sim$ 2000 distinct phones available in the PHOIBLE dataset (Moran and McCloy, 2019). Many phones in this dataset cannot be found in high-resource languages, therefore we cannot learn their representation directly. To tackle this

problem, we further decompose the phone into lower-level representations: *phonological features* or *articulatory features*. For example, Figure 1.7 shows the IPA chart of the most common basic vowels. From the chart, we can see the phone [i] can be characterized by the phonological features of *unrounded*, *close*, *front*, *vowel* . Those features can be extracted using some existing phonetic tools (Mortensen et al., 2016a) or be parsed automatically using the simple rule introduced in Chapter 2. We can first learn the representations of the phonological features, then use them to compose the representations of each phones. This transformation reduces the task of estimating representations over each phone into a much simpler task of estimating representations over each phonological feature because the feature inventory size is typically much smaller than the phone inventory size (for example, 20 vs 2000). More importantly, it enables us to recognize phones that are unseen in the training set. This helps us achieve zero-shot learning of most unseen phones. This idea is introduced in Chapter 2 and refined in Chapter 4.

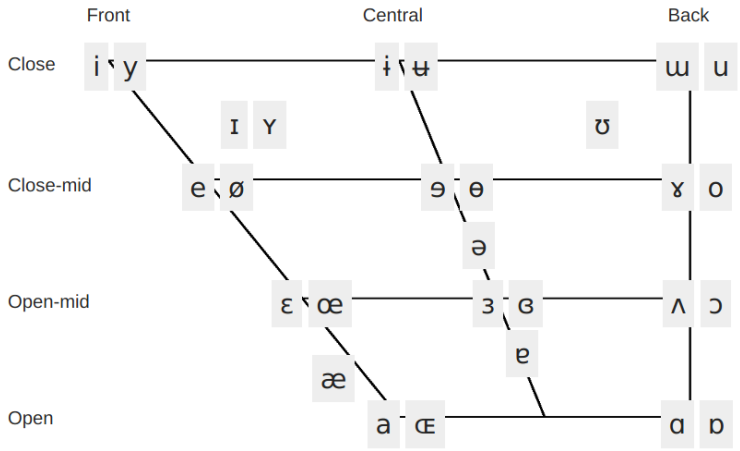



Figure 1.7: IPA chart of vowels.Where symbols appear in pairs, the one to the right represents a rounded vowel.

### 1.3.3 Pronunciation Model

The next component in the pipeline is the pronunciation model. We discuss it in Chapter 5. It is essentially a G2P (grapheme-to-phoneme) model that can predict the phoneme pronunciation given a grapheme sequence: $P = \delta_{\text{pm}}(Y)$. For high-resource languages, the G2P model can be either trained using a dictionary or be developed using rule-based systems (CMU, 2000; Mortensen et al., 2018). However, the majority of the languages do not have any accessible dictionaries or rules, therefore we consider an approximated pronunciation model $\hat{\delta}_{\text{pm}}$ instead. The easiest approximation model is to simply take a high-resource languages and apply its G2P model instead. For example, English is well-studied with respect to its G2P model and we consider it as one of our

baseline pronunciation model. The model, however, only considers the orthography rules in one language and might be inappropriate in other languages. A better option for approximation is to use a language that is similar to the target language. For example, it would be more appropriate to use German model to approximate English model than the Spanish model as German and English belonging to the same Germanic language family. This nearest model however, still suffers from the large variance issue as it depends on a monolingual model for the inference.

In this thesis, we propose a multilingual G2P model as our pronunciation model (Li et al., 2022b). For any target language $l_{\text{target}}$, this G2P model selects top-$k$ nearest languages: $l_{\text{topk}} \in \text{KNN}(l_{\text{target}})$ whose training set is available, then during the inference, it first propose $k$ hypothesis using each nearest language model $\delta_{l_{\text{topk}}}$, the models are ensembled by combining hypothesis into a lattice to emit the most-likely approximated sequence:

$$\hat{\delta}_{l_{\text{target}}} = \text{Ensemble}(\{\delta_{l_{\text{topk}}} | l_{\text{topk}} \in \text{KNN}(l_{\text{target}})\}) \tag{1.9}$$

The similarity metric between languages is defined to be the shortest path of two languages on the phylogenetic tree (i.e: language family tree). This approach enables us to approximate the pronunciation model for every language in Glottolog database (Nordhoff and Hammarström, 2011), which contains phylogenetic information about 7915 languages.

### 1.3.4 Language Model

The last component in the pipeline is the language model. We discuss it in Chapter 6. For high-resource languages, many modern architectures can be used to train the large language models such as BERT (Devlin et al., 2018), GPT-3 (Brown et al., 2020), Megatron-LM (Shoeybi et al., 2019). For low-resource languages, however, the text dataset is usually limited or not available as discussed in the previous section, therefore we cannot train such large language models. Instead, we rely on the classical n-gram models. In our language model, we first estimate the vocabulary $V = \{w_1, w_2, ..., w_{|V|}\}$ from the raw text dataset $\{Y_i\}$ when it is available. For each word $w_i \in V$, its pronunciation can be approximated using the pronunciation model and then this lexicon information can be encoded into a lexicon graph $L$. The text dataset also enables us to estimate the classical n-gram language model by counting n-grams statistics $C(w_1, ..., w_n)$. This n-gram language model can be then encoded into a grammar graph $G$. Composing the lexicon graph $L$ and the grammar graph $G$ as well as the CTC topology graph $H$ would generate a WFST-based language decoder HLG (Miao et al., 2015).

We realize that the text dataset requirement $\{Y_i\}$ can be further relaxed as the building blocks of the HLG graph only consist of the statistics $\{V, C\}$ estimated from the text dataset. For languages whose text dataset $\{Y_i\}$ is absent but $\{V, C\}$ is available, we can still proceed to build the decoder HLG. This is common for many languages in the internet: while only a few hundred

languages are recognized as being in use for web texts on the World Wide Web (Lewis, 2016), there exists several large databases collecting lexicon-related statistics for thousands of languages. For example, Crúbadán is a database consisting of vocabulary, bigrams, and character statistics for around 2000 languages (Scannell, 2007). Employing statistics from it, we build speech recognition systems for around 2000 languages. In the minimalist setting, we only assume the lexicon for the target language as the text-only dataset using the Panlex dataset (Kamholz et al., 2014). This further expands the number to 6185 languages.

## 1.4 Overview

We briefly discuss the organization of this thesis and main topics of each chapter in this subsection.

In the first part of this thesis, we discuss the acoustic model. In Chapter 2, we introduce the phonological features or articulatory features as the building blocks to represent high-level acoustic units such as phones and phonemes. we demonstrate that using discrete phonological features, we can efficiently share acoustic knowledge across languages. In Chapter 3, we introduce the concept of *allophone* and discuss how to use them to connect language-dependent phonemes and language-independent phones. In Chapter 4, we combine the ideas of Chapter 2 and Chapter 3. We propose the complete acoustic model pipeline by using a hierarchical structure.

In the second part of this thesis, I discuss the language model related topics. In Chapter 5, we introduce the pronunciation model used in this thesis. We demonstrate a multilingual grapheme-to-phoneme model which can be applied to any language in Glottolog. In Chapter 6, we build a language model using Crúbadán: an large online n-gram statistics for endangered languages. Finally, the acoustic model proposed in Part I, the pronunciation model proposed in Chapter 5 and the language model proposed in Chapter 6 are combined together to create a full speech pipeline. This enables us to build speech recognition models for around 6000 languages, which is the main focus in this thesis.

In the third part of this thesis, we introduce two datasets we used in the previous modeling part. In Chapter 7, we discuss the problem of *phoneme inventory estimation*. Two models are proposed to extend the language coverage by the PHOIBLE database. The estimated phoneme inventory are used in our pipeline (both the acoustic model and the pronunciation model) when the target language is not supported by PHOIBLE. In Chapter 8, a multilingual phonetic database is introduced to evaluate the acoustic model. This dataset is used to evaluate phone prediction performance for our acoustic model in Part I. In Chapter 9, we also discuss how to align speech and text pairs using the proposed acoustic model and pronunciation model.

## Contributions

This thesis addresses the problem of low-resource / zero-resource speech recognition and proposes a speech recognition pipeline for thousands of languages with the following key contributions:

1. Propose a multilingual acoustic model that can potentially perform phoneme recognition for around 8000 languages.
2. Introduce a multilingual pronunciation model (Grapheme-to-Phoneme conversion model) that can approximate G2P models without any supervised dataset for the target language.
3. Demonstrate that we can achieve speech recognition for around 6000 languages by combining acoustic model, pronunciation model, and language model
4. Develop several useful datasets and toolkits for low-resource speech recognition.

## Publications

This section lists my first-author publications directly related to this thesis.

1. Xinjian Li, Siddharth Dalmia, David R.Mortensen, Juncheng Li, Alan Black and Florian Metze Towards zero-shot learning for automatic phonemic transcription. 2020 Proceedings of the AAAI Conference on Artificial Intelligence. (Chapter 2) (Li et al., 2020b)
2. Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anas-tasopoulos, David R-Mortensen, Graham Neubig, Alan W Black, et al. 2020. Universal phone recognition with a multilingual allophone system. In ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (Chapter 3) (Li et al., 2020a)
3. Xinjian Li, Juncheng Li, Florian Metze, and Alan W Black. 2021. Hierarchical phone recognition with compositional phonetics. In Proc. Interspeech (Chapter 4) (Li et al., 2021a)
4. Xinjian Li, Florian Metze, David R Mortensen, Shinji Watanabe, and Alan W Black. 2022. Zero-shot learning for grapheme to phoneme conversion with language ensemble. Findings of ACL. (Chapter 5) (Li et al., 2022b)
5. Xinjian Li, Florian Metze, David R Mortensen, Alan W Black and Shinji Watanabe. 2022. ASR2K: Speech Recognition for Around 2000 Languages without Audio. Interspeech 2022. (Chapter 6)

6. Xinjian Li, Florian Metze, David R. Mortensen, Alan W. Black, Shinji Watanabe. LREC 2022 Phone Inventories and Recognition for Every Language. LREC (Chapter 7) (Li et al., 2022a)

7. Xinjian Li, David R. Mortensen, Florian Metze, and Alan W. Black. Multilingual phonetic dataset for low resource speech recognition In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (Chapter 8) (Li et al., 2021c)

8. Xinjian Li, Ondřej Klejch, Peter Bell, Alan W Black, Shinji Watanabe. Submitted to EMNLP 2023 (Chapter 9)

**Others**

This section lists my publications partially related to this thesis.

1. Xinjian Li, Siddharth Dalmia, Alan W Black, and Florian Metze. 2019b. Multilingual speech recognition with corpus relatedness sampling. Proc. Interspeech 2019 (Li et al., 2019a)

2. Mortensen, David, Xinjian Li, Patrick Littell, Alexis Michaud, Shruti Rijhwani, Antonios Anastasopoulos, Alan Black, Florian Metze, and Graham Neubig AlloVera: a multilingual allophone database. LREC 2020: 12th Language Resources and Evaluation Conference. 2020. (Mortensen et al., 2020)

3. Neubig, Graham, Shruti Rijhwani, Alexis Palmer, Jordan MacKenzie, Hilaria Cruz, Xinjian Li, Matthew Lee et al. A Summary of the First Workshop on Language Technology for Language Documentation and Revitalization. (Neubig et al., 2020)

4. Xinjian Li, Juncheng Li, Jiali Yao, Alan W. Black, and Florian Metze Phone Distribution Estimation for Low Resource Languages. ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021. (Li et al., 2021b)

5. Gupta, Akshat, Xinjian Li, Sai Krishna Rallabandi, and Alan W. Black. "Acoustics based intent recognition using discovered phonetic units for low resource languages." In ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7453-7457. IEEE, 2021. (Gupta et al., 2021)

6. Siminyu, Kathleen, Xinjian Li, Antonios Anastasopoulos, David Mortensen, Michael R. Marlo, and Graham Neubig (2021) Phoneme Recognition Through Fine Tuning of Phonetic Representations: A Case Study on Luhya Language Varieties. Proc. Interspeech 2021, 271-275 (Siminyu et al., 2021)

7. Mortensen, David R., Jordan Picone, Xinjian Li, and Kathleen Siminyu (2021) Tusom2021: A Phonetically Transcribed Speech Dataset from an Endangered Language for Universal Phone Recognition Experiments. Proc. Interspeech 2021, 3660-3664 (Mortensen et al., 2021)

8. Xinjian Li, Ye Jia, and Chung-Cheng Chiu: Textless direct speech-to-speech translation with discrete speech representation. ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023. (Li et al., 2023)

# Part I

# Acoustic Models

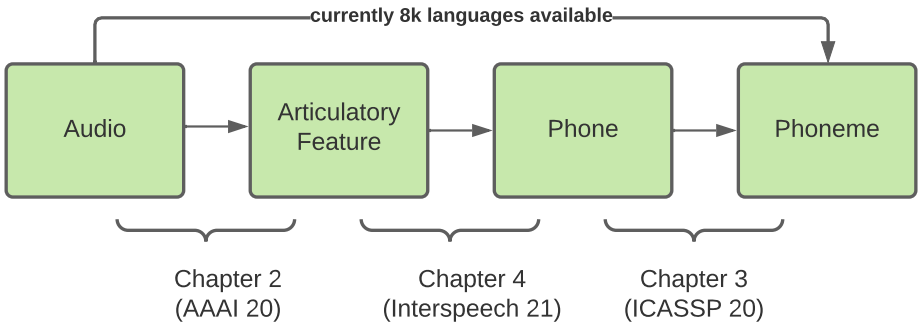



Figure 1.8: The acoustic model covered in Part I

# Chapter 2

# Zero-shot Learning using Phonological Features

## Summary

We start Part I of this thesis by considering the task of phonemic recognition or phonemic transcription. This task is to train an acoustic model $p_{\text{am}}(P|X)$ where $X$ is the speech audio and $P$ is the phoneme sequence or phoneme transcription. This task is useful for low-resource applications such as language documentation and language preservation. However, due to the lack of training sets, only a tiny fraction of languages have phonemic transcription models. Fortunately, multilingual acoustic modeling provides a solution given limited audio training data. A more challenging problem is to build phonemic transcribers for languages with zero training data. The difficulty of this task is that phoneme inventories $P$ are language dependent and they often differ between the training languages and the target language, making it infeasible to recognize unseen phonemes.

In this chapter, we address this problem by adopting the idea of zero-shot learning. In particular, we introduce the concept of the *phonological features* or *articulatory attributes*. These features are the lowest-level units used in this thesis. Those attributes enables us to recognize unseen phonemes in the target language without any training data. In our model, we decompose phonemes into corresponding phonological features or articulatory attributes such as *vowel* and *consonant*. Instead of predicting phonemes directly, we first predict distributions over articulatory attributes, and then compute phoneme distributions with a customized acoustic model. We evaluate our model by training it using 13 languages and testing it using 7 unseen languages. We find that it achieves 7.7% better phoneme error rate on average over a standard multilingual model. The acoustic model introduced in this chapter will be enhanced in Chapter 3 and Chapter 4.

Xinjian Li, Siddharth Dalmia, David R.Mortensen, Juncheng Li, Alan Black and Flo-

rian Metze Towards zero-shot learning for automatic phonemic transcription. Proceedings of the AAAI Conference on Artificial Intelligence.

## 2.1 Introduction

Over the last decade, automatic speech recognition (ASR) has achieved great successes in many rich-resourced languages such as English, French and Mandarin. On the other hand, speech resources are still sparse for the majority of other languages. They cannot thus benefit directly from recent technologies. As a result, there is an increasing interest in building speech processing systems for low-resource languages. In particular, phoneme transcription tools are useful for low-resource language documentation by improving workflow for linguists to analyze those languages (Adams et al., 2018; Michaud et al., 2018).

A more challenging task is to transcribe phonemes in the language with zero training data. This task has significant implications in documenting endangered languages and preserving the associated cultures (Gippert et al., 2006). This data setup has mainly been studied in the unsupervised speech processing field (Glass, 2012; Versteegh et al., 2015; Hermann and Goldwater, 2018), which typically uses an unsupervised technique to learn representations which can be used towards speech processing tasks.

However, those unsupervised approaches could not generate phonemes directly and there has been few works studying zero-shot learning for unseen phonemes transcription, which consist of learning an acoustic model without any audio data or text data for a given target language and unseen phonemes. In this work, we aim to solve this problem to transcribe unseen phonemes for unseen languages without considering any target data, audio or text.

The prediction of unseen objects has been studied for a long time in the computer vision field. For specific object classes such as *faces*, *vehicles* and *cats*, a significant number manually labeled data is usually available, but collecting sufficient data for every object human could recognize is impossible. Zero-shot learning attempts to solve this problem to classify unseen objects using mid-level side information. For example, *zebra* can be recognized by detecting attributes such as *stripped*, *black* and *white*. Inspired by approaches in computer vision research, we propose the Universal Phonemic Model (UPM) to apply zero-shot learning to acoustic modeling. In this model, we decompose the phoneme into its attributes and learn to predict a distribution over various articulatory attributes. For example, the phoneme /a/ can be decomposed into its attributes: *vowel*, *open*, *front* and *unrounded*. This can then be used to infer the unseen phonemes for the test language as the unseen phonemes can be decomposed into common attributes covered in the training phonemes.

Our approach is summarized in Figure 2.1. First, frames are extracted and a standard acoustic

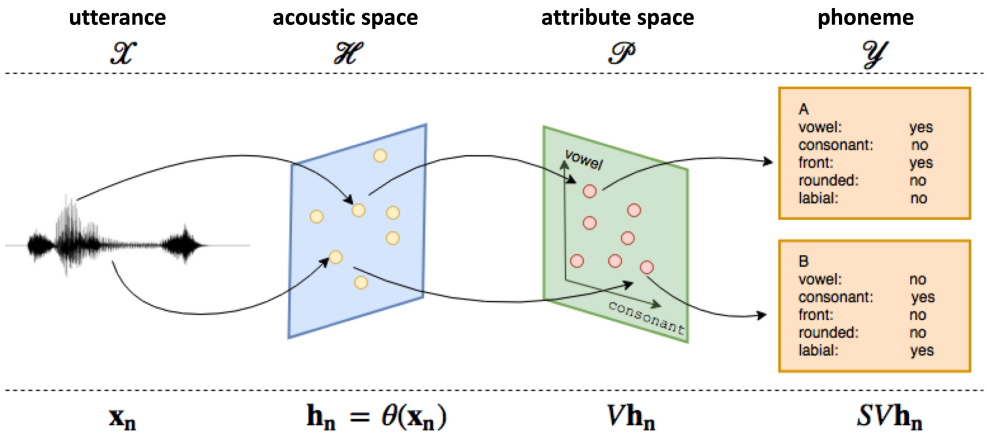



Figure 2.1: Illustration of the proposed zero-shot learning framework. Each utterance is first mapped into acoustic space (or hidden space) $\mathcal{H}$. Then we transform each point in the acoustic space into attribute space $\mathcal{P}$ with a linear transformation $V$. Finally phoneme distributions can be obtained by applying a signature matrix $S$

model is applied to map each frame into the acoustic space (or hidden space) $\mathcal{H}$. Next we transform it into the attribute space $\mathcal{P}$ which reflects the articulatory distribution of each frame (such as whether it indicates a *vowel* or a *consonant*). Then, we compute the distribution of phonemes for that frame using a predefined signature matrix $S$ which describes relationships between articulatory attributes and phonemes in each language.

To evaluate our UPM approach, we trained the model on 13 languages and tested it on another 7 languages. We also trained a multilingual acoustic model as a baseline for comparison. The result indicates that we consistently outperform the baseline multilingual model, and we achieve 7.7% improvements in phoneme error rate on average.

The main contributions of this chapter are as the followings:

1. We propose the Universal Phonemic Model (UPM) that can recognize unseen phonemes during training by incorporating knowledge from the phonetics/phonology domain.

2. We introduce a sequence prediction model to integrate a zero-shot learning framework for sequence prediction problem.

3. We show that our model is effective for 7 different languages, and our model gets 7.7% better phoneme error rate over the baseline on average.

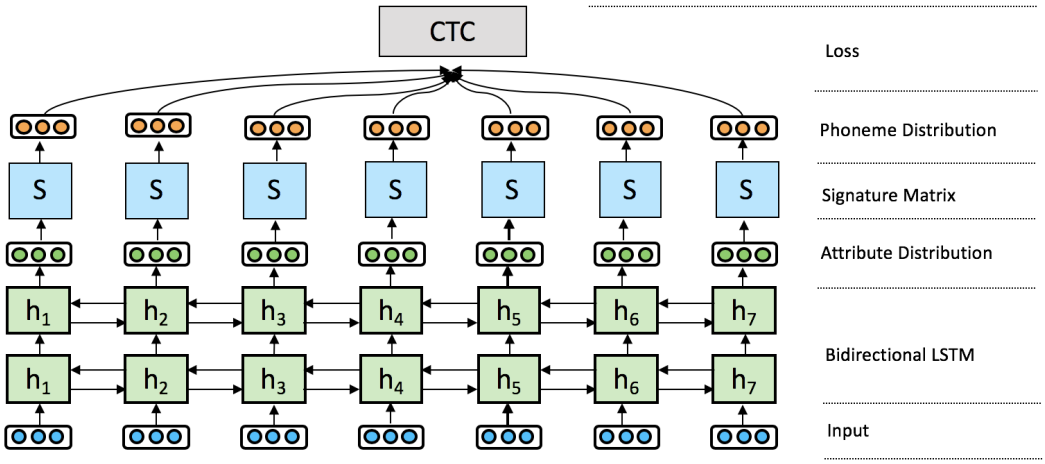



Figure 2.2: Illustration of the sequence model for zero-shot learning. The input layer is first processed with a Bidirectional LSTM acoustic model, and produces a distribution over articulatory attributes. Then it is transformed into a phoneme distribution by a language dependent signature matrix $S$

## 2.2 Approach

This section explains the details of our Universal Phonemic Model (UPM). In the first section, we describe how we constructed a proper set of articulatory attributes for acoustic modeling. Next, we demonstrate how to assign attributes to each phoneme by giving an algorithm to parse X-SAMPA format. Finally we show how we integrate the phonetic information into the sequence model with a CTC loss (Graves et al., 2006).

### 2.2.1 Articulatory Attributes

Unlike attributes in the computer vision field, attributes of phonemes are independent of the corpus and dataset, they are well investigated and defined in the domain of articulatory phonetics (Ladefoged and Johnson, 2014). Articulatory phonetics describes the mechanism of speech production such as the manner of articulation and placement of articulation, and it tends to describe phones using discrete features such as voiced, bilabial (made with the two lips) and fricative. These articulatory features have been shown to be useful in speech recognition (Kirchhoff, 1998; Stüker et al., 2003b; Müller et al., 2017a), and are a good choice for attributes for our purpose. We provide some categories of articulatory attributes below.

**Consonants**. Consonants are formed by obstructing the airstream through the vocal tract. They can be categorized in terms of the placement and the manner of this obstruction. The placements can be largely divided into three classes: *labial*, *coronal*, *dorsal*. Each of the class have more

fine-grained classes. The manners of articulation can be grouped into: *stop*, *fricative*, *approximant* etc.

**Vowel**. In the production of vowels, the airstream is relatively unobstructed. Each vowel sound can be specified by the positions of lips and tongue (Ladefoged and Johnson, 2014). For instance, the tongue is at its highest point in the front of the mouth for *front* vowels. Additionally, vowels can be characterized by properties such as whether the lips are rounding or not (*rounded*, *unrounded*).

**Diacritics**. Diacritics are small marks to modify vowels and consonants by attaching to them. For instance, *nasalization* marks a sound for which the velopharyngeal port is open and air can pass through the nose. To make the articulatory attribute set manageable, we assign attributes of diacritics to some existing consonants attributes if they share similar articulatory property. For example, *nasalization* is treated as the *nasal* attribute in consonants.

In addition to articulatory attributes mentioned above, we note that we also need to allocate an special attribute for blank in order to predict blank labels in CTC model, and backpropagate their gradients into the acoustic model. Thus, our articulatory attribute set $A_{phone}$ is defined as the union of these three domain attributes as well as the blank label,

$$A_{phone} = A_{consonants} \cup A_{vowels} \cup A_{diacritics} \cup \{blank\}$$

### Attribute Assignment

Next, we need to assign each phoneme with appropriate attributes. There are multiple approaches to retrieve articulatory attributes. The simplest one is to use tools to collect articulatory features for each phoneme (Mortensen et al., 2016a). However, those tools only provide coarse-grained phonological features but we expect more fine-grained and customized articulatory features. In this section, we propose a naive but useful approach for attribute assignment. We note that we use X-SAMPA format to denote each IPA in this work. X-SAMPA was devised to produce a computer-readable representation for IPA. Each IPA segment can be mapped to X-SAMPA with appropriate rule-based tools (Mortensen et al., 2018). For example, IPA /ə/ can be represented as /@/ in X-SAMPA.

The assignment can be formulated as the problem to construct an assignment function $f : P_{xsampa} \rightarrow 2^{A_{phone}}$ where the domain $P_{xsampa}$ is the set of all valid X-SAMPA phonemes, and the range $2^{A_{phone}}$ is a subset of articulatory attributes for each phoneme . The assignment function should map each phoneme into its corresponding subset of $A_{phone}$. To construct the function in the entire domain $P_{xsampa}$, we first manually map a small subset $P_{base} \subset P_{xsampa}$ and construct a restricted assignment function $f\big|_{P_{base}} : P_{base} \rightarrow 2^{A_{phone}}$. The mapping is customizable and has been verified with the IPA handbook (Decker et al., 1999). Then for every phoneme $p \in P_{xsampa}$,

**Algorithm 1:** G2P algorithm

**Data:** X-SAMPA representation of phoneme $p$
**Result:** Articulatory attribute set $A \subseteq A_{phone}$ for $p$
$A \leftarrow []$
**while** $p \notin P_{base}$ **do**
  find the longest suffix $p_s \in P_{base}$
  Add $f\big|_{P_{base}}(p_s)$ to $A$
  Remove suffix $p_s$ from $p$
**end**
Add $f\big|_{P_{base}}(p)$ to $A$

we continue to remove diacritics suffix from it until it could be found in $P_{base}$. For example, to recognize /ts_>/, we can first match the suffix, /_>/ as an *ejective*, and then recognize /ts/ as a consonant defined in $P_{base}$. The Algorithm 1 summarizes our approach.

### 2.2.2 Sequence model for zero-shot learning

Zero-shot learning has rarely been applied to speech sequence prediction problems. Zero-shot translation is an example of applying zero-shot learning to a different type of sequence problems(Johnson et al., 2017). In the standard settings, the zero-shot translation means that the target language pair is not in the training dataset. However, both languages should be already seen in other training pairs. In contrast, we assume a harder problem here: there is no available training audio or text for the target language at all.

In this section we describe a novel sequence model architecture for zero-shot learning. We adapt a modified ESZSL architecture from (Romera-Paredes and Torr, 2015). While the original architecture is devised to solve the classification problem with CNN(DECAF) features, our model aims to optimize a CTC loss over a sequence model as shown in Figure 2.2. We note our architecture is a general model, and it can also be used for other sequence prediction problems in zero-shot learning.

Given the training set $\{(\mathbf{x_n}, \mathbf{y_n}, \phi_n), n = 1...N\}$ where each input $\mathbf{x_n} \in \mathcal{X}$ is an utterance, $\phi_n$ is its language, and $\mathbf{y_n} \in \mathcal{Y}$ is the corresponding phoneme transcription. Suppose that $\mathbf{x_n} = (x_n^1, ..., x_n^T)$ is the input sequence where $x_n^t$ is the frame of time step $t$, and $T$ is the length of $\mathbf{x_n}$. Each frame $x_n^t$ is first projected into a feature vector $h_n^t \in \mathbb{R}^d$ in the hidden space $\mathcal{H}$ with a Bidirectional LSTM model.

$$h_n^t = \theta(x_n^t; W_{\text{LSTM}}) \tag{2.1}$$

| Language | Corpus Name | # Utterances | Language | Corpus Name | # Utterances |
|---|---|---|---|---|---|
| English | TED | 268k | Mandarin | Hkust | 197k |
| English | Switchboard | 251k | Mandarin | OpenSLR 18 | 13k |
| English | Librispeech | 281k | Mandarin | LDC98S73 | 36k |
| Amharic | OpenSLR 25 | 10k | Bengali | OpenSLR 37 | 196k |
| Cebuano | IARPA-babel301b-v2.0b | 43k | Dutch | Voxforge | 8k |
| Italian | Voxforge | 10k | Javanese | OpenSLR35 | 185k |
| Kazakh | IARPA-babel302b-v1.0a | 48k | Kurmanji | IARPA-babel205b-v1.0a | 46k |
| Lao | IARPA-babel203b-v3.1a | 66k | Turkish | IARPA-babel105b-v0.4 | 82k |
| Sinhala | openSLR52 | 185k | | | |
| German | Voxforge | 41k | Mongolian | IARPA-babel401b-v2.0b | 45k |
| Russian | Voxforge | 8k | Spanish | Callhome Hub4 | 31k |
| Swahili | OpenSLR 25 | 10k | Tagalog | IARPA-babel106b-v0.2g | 93k |
| Zulu | IARPA-babel206b-v0.1e | 60k | | | |

Table 2.1: Corpora of the training set and the test set used in the experiment. Both baseline model and proposed model are trained with 17 corpus across 13 languages, and tested on 7 corpus in 7 languages.

where $W_{\text{LSTM}}$ is the parameter of the Bidirectional LSTM model. We assume that our phoneme inventory of $\phi_n$ consists of $z$ phonemes in the training set, each of them having a signature of $a$ attributes constructed as mentioned above. We can first represent our attributes in a constant signature matrix $S \in \{0,1\}^{z \times a}$ of $\phi_n$. The $(i,j)$ cell in the signature matrix is 1 if the $i$-th phoneme has been assigned the $j$-th attribute, otherwise it is assigned to 0. We note that while the signature matrix is constructed automatically in this work, it can be refined by linguists using phonology in each language. Then, we transform $h_n^t$ into articulatory logits with the transformation matrix $V \in \mathbb{R}^{a \times d}$. Then it is further processed into the phoneme logits $l_n^t$ with $S$.

$$l_n^t = SVh_n^t \tag{2.2}$$

The logits $\mathbf{l_n} = (l_n^1, ..., l_n^T)$ are then combined with $\mathbf{y_n}$ to compute the CTC loss (Graves et al., 2006). Additionally, regularizing $V$ has been proved to be useful in the original ESZSL architecture (Romera-Paredes and Torr, 2015). Eventually our target is to minimize the following loss function:

$$\underset{V, W_{\text{LSTM}}}{\text{minimize}} \, \text{CTC}(\mathbf{x_n}, \mathbf{y_n}; V, W_{\text{LSTM}}) + \Omega(V) \tag{2.3}$$

where $\Omega(V)$ is an simple $\ell^2$ regularization. This objective can be easily optimized using stan-

| Language | # unseen phoneme | Baseline PER% | UPM PER% | Baseline Substitution% | UPM Substitution% |
|---|---|---|---|---|---|
| German | 2 | 68.0 | 64.9 | 51.9 | 46.9 |
| Mongolian | 18 | 87.8 | 77.5 | 44.1 | 35.8 |
| Russian | 19 | 74.5 | 54.4 | 63.5 | 34.5 |
| Swahili | 2 | 55.7 | 48.9 | 27.4 | 26.6 |
| Tagalog | 0 | 60.7 | 57.0 | 27.2 | 20.1 |
| Spanish | 2 | 48.6 | 44.4 | 31.0 | 26.2 |
| Zulu | 8 | 73.1 | 67.9 | 36.2 | 33.5 |
| Average | 7.3 | 66.9 | **59.2** | 40.2 | **31.9** |

Table 2.2: Phoneme error rate and phoneme substitution rate of the baseline, and our approach. Our model (UPM) outperforms the baseline for all languages, by 7.7% (absolute) in phoneme error rate, and 8.3% in phoneme substitution error rate.

dard gradient descent methods.

At the inference stage, we usually consider a new language $\phi_{test}$ with a new phoneme inventory. Suppose that the new inventory is composed of $z'$ phonemes, then we can automatically create a new signature matrix $S' \in \{0,1\}^{z' \times a}$, and estimate probability distribution of each phoneme $P_{acoustic}(p|x_n^t)$ from logits using $S'$ instead of $S$.

## 2.3 Experiments

### 2.3.1 Dataset

We prepare two datasets for this experiment. The training set consists of 17 corpora from 13 languages, and the test set is composed of corpora from 7 different languages. They are used by both our model and the baseline described later. Details regarding each corpus and each language are provided in Table 2.1.

We briefly describe our strategy of corpus selection in the experiment. To select the training corpus, the rich-resourced languages should be taken into account firstly to make sure the acoustic model can be fully trained. Therefore, we add three English corpora and three Mandarin corpora to the training set. Additionally, we expect both the baseline and our Universal Phonemic Model should be trained to recognize a variety of phonemes from different languages. Therefore we collect a number of corpora from different language families and diverse regions. Finally, we attempt to make the acoustic model robust to various channels and speech styles. For example, TED (Rousseau et al., 2012) is the conference style, Switchboard (Godfrey et al., 1992) is the spontaneous conversation style and Librispeech is the reading style (Panayotov et al., 2015). We note that 5 percent of the entire corpus was used as the validation set. The test corpora are selected

in a similar style. They are selected from a variety of languages: not only from rich-resourced languages, but also low-resourced languages with stable audio alignments and reliable g2p models.

### 2.3.2 Experimental Settings

We use the EESEN framework for the acoustic modeling (Miao et al., 2015). All the transcripts are transcribed into phonemes with Epitran (Mortensen et al., 2018). The input feature is 40 dimension high-resolution MFCCs, the encoder is a 5 layer Bidirectional LSTM model, each layer having 320 cells. The signature matrix is designed as we discussed above, different signature matrices are used for different languages. We train the acoustic model with stochastic gradient descent, using a learning rate of 0.005. In each iteration, we apply the uniform sampling (Li et al., 2019b): first randomly select a corpus from the entire training set, and then randomly choose one batch from that corpus.

Our baseline model is the multilingual acoustic model with a shared phoneme inventory. This type of architecture is one of the standard approaches in the multilingual ASR community (Tong et al., 2017; Vu and Schultz, 2013). In this architecture, all languages share a common acoustic model and a single output layer. The output layer is to predict phonemes in the universal phoneme inventory shared by all the training languages. In our experiment, the inventory consists of 131 distinct phonemes from 14 training languages. To compare the baseline with the proposed model, we also use the Bidirectional LSTM model as the encoder to compute phoneme distributions $P(p|x_n^t)$. Then we decode phonemes with greedy decoding as in our approach. We use the same configuration of LSTM architecture as well as the training criterion. As we focus on phonemic transcriptions in this work, we use phoneme error rate (PER) as the metric for evaluation.

### 2.3.3 Results

Our results are summarized in Table 2.2. As is shown, our approach consistently outperforms the baseline in terms of phoneme error rate. For example, the baseline achieves 55.7% phoneme error rate when evaluated with Swahili, and our approach obtains 48.9% in the same test set. For each language in our evaluation, we observe that we improve the phoneme error rate from 3.1% (German) to 20.1% (Russian) respectively. On average, the baseline has 66.9%, and our model gets 7.7 % better phoneme error rate.

The table also indicates the strong correlation between the number of unseen phonemes and the improvement in the phoneme error rate. For example, Russian achieves the largest improvement with our UPM: it improves significantly by 20.1% phoneme error rate. In our experiment, the Russian phoneme inventory has 48 phonemes in total out of which 19 of them are unseen during training. This suggests our model has a good generalization ability to adapt to languages whose

| Language | Baseline unseen PER% | UPM unseen PER% | Baseline seen PER% | UPM seen PER% |
|---|---|---|---|---|
| German | 100.0 | 100.0 | 63.9 | 61.9 |
| Mongolian | 100.0 | 91.9 | 86.8 | 78.6 |
| Russian | 100.0 | 96.1 | 69.5 | 51.7 |
| Swahili | 100.0 | 86.4 | 54.3 | 46.2 |
| Tagalog | N.A. | N.A. | 57.4 | 54.2 |
| Spanish | 100.0 | 58.0 | 45.2 | 41.7 |
| Zulu | 100.0 | 88.3 | 70.5 | 64.6 |
| Average | 100.0 | **89.8** | 64.2 | **57.0** |

Table 2.3: Phoneme error rate (%PER) of the seen phonemes and unseen phonemes in the baseline and our approach.

acoustic contexts are rarely known. On the other hand, every phoneme in the Tagalog inventory has been covered by other languages in the training set. Therefore, the number of its unseen phoneme is 0 and the corresponding 3.7% phoneme error rate improvement is relatively limited. Similarly, the least improved language is German, which improved from 68.0% to 64.9% because there are only two unseen phonemes in German. This fact can also be explained by the relationship between German and English. German comes under the West Germanic branch in the Indo-European language family like English. As English is the largest training set in this experiment, phonemes of English are well-trained in the baseline and should be generalizing well to German. Therefore it is hard for UPM to outperform by a large margin. Additionally, we find that the correlation between the number of unseen phonemes and phoneme error rates is relatively weak. For example, Tagalog has 12% higher phoneme error rate compared with Spanish, even its unseen phonemes are less than Spanish. This might be explained by the discrepancy of the phoneme distribution between the target language and training languages. For example, even though in principle all the phonemes of Tagalog have been covered in the training languages, their relative frequencies are not similar, which would affect the quality of the results.

To further investigate the reason for improvements for our model, we computed the (phoneme) substitution error rate, shown in the two right columns of Table 2.2. It goes down from 40.2% in the baseline to 31.9% in our model. The numbers show that we have 8.3% improvement in substitution error rate. This result suggests that our model is good at improving confusions between phonemes. However, it also indicates that our model is not able to improve addition and deletion errors.

To understand how the number of training languages contributes to the performance in the experiment, we train different models by changing the numbers of training languages: we train those models with 2, 6, 10, 14 languages. The first two languages are English and Mandarin which are corresponding to the 6 well resourced corpus in Table.2.1. The other 4, 8, 12 languages are

randomly selected from the remaining training languages.

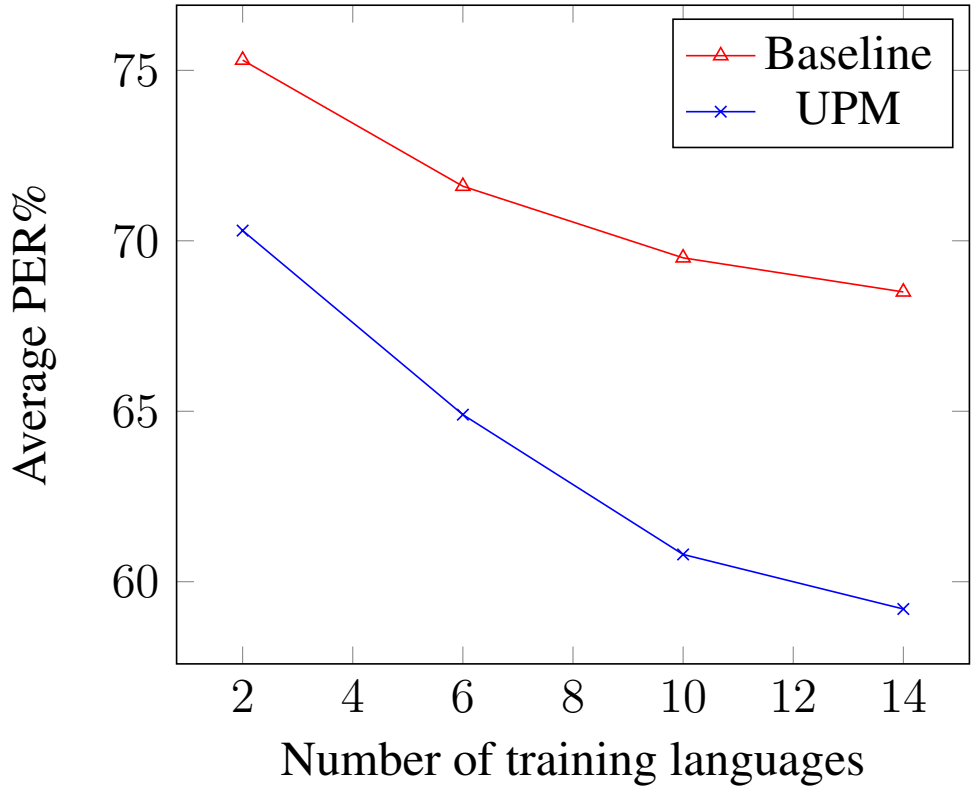



Figure 2.3: Illustration of the relationship between the number of training languages and the average phoneme error rate over 7 languages

Figure.2.3 demonstrates their performance: the red line (with triangular mark) and blue line (with cross mark) indicate the average PER of the baseline and UPM respectively. They suggest that increasing the number of training languages is helpful to reduce phoneme error rate for both models. For the baseline model, it indicates that the acoustic model get exposed to more diverse phonemes present in different languages. Therefore it learns to predict them with reduced error rates in the test set. Our UPM also improves by learning various acoustic contexts of broader articulatory attributes. The curves in Figure.2.3 show that UPM outperforms the baseline consistently with different training size. Additionally, the gap of phoneme error rate between the two models has increased when using more languages: the gap increased from 5.0 to 9.3. The results illustrate that our UPM is better at taking advantage of the diverse training languages. Our model can infer correlations between phonemes by using their shared articulatory attributes. This ability is helpful when a specific phoneme is rarely seen but its attributes have already been well-trained using other related phonemes. On the contrary, the baseline is not adapted well to those rare phonemes or

unseen phonemes. It fails to predict those phonemes when their training data are limited.

Finally, to highlight the ability of our model, we compute the phoneme error rate for each phoneme, then classify them into the seen group and unseen group based on whether the phoneme is available in the training set. To compute phoneme error rate in this case, we align the expected phonemes with the predicted phonemes using their edit distance, the phoneme error rate here denotes the correction rate for each expected phone. Table.2.3 demonstrates the results of both the baseline and UPM, it suggests the UPM outperforms the baseline on both groups. On average, UPM would predict 10.2 % better for the unseen groups and 7.2 % better for the seen groups. The average numbers demonstrate that our approach has the ability to predict unseen phonemes and could even be adapted better to seen groups. The table also shows the difficulty of the task and the weakness of our approach: we could not predict any unseen phonemes for German. The two unseen phonemes of German are /pf/ and /C/, but the frequencies of both phonemes are less than 0.5 % in the test set, which makes the model extremely unstable when predicting those phonemes. On the other hand, the Spanish improvement of unseen PER is extremely significant, which can also be explained by the unstable prediction over low frequency unseen phonemes. Additionally, the 89.8 error rate of unseen groups is still not practical in the real-world production systems.

## 2.4 Related Work

We briefly outline several areas of related works, and describe their connections and differences with this work. Zero-shot learning was first applied to recognize unseen objects during training in the computer vision field (Lampert et al., 2009; Palatucci et al., 2009; Socher et al., 2013). However those works rarely mention speech recognition.

Meanwhile there has been growing interests in zero-resource speech processing (Glass, 2012; Jansen et al., 2013), most of the work focusing on tasks like acoustic unit discovery, unsupervised segmentation and spoken term discovery (Heck et al., 2017). These models are useful for various extrinsic speech processing tasks like topic identification. However, the unsupervised concept cannot be directly grounded to actual phonemes, hence making it impracticable to do speech recognition or acoustic modeling. The usual intrinsic evaluations that these zero resource tasks are tested on is ABX discriminability task or the unsupervised word error rate which are good for quality estimates but not practical as they use an oracle or ground truth labels to assign cluster labels. In addition these approaches demands a modest size of audio corpus of targeting language (e.g: 2.5h to 40h). In contrast, our approach assumes no audio corpus and no text corpus for targeting languages. The idea of decomposing speech into concepts was also discussed by (Lake et al., 2014), where the authors propose a generative model to learn representations for spoken words which they then use to classify words with only one training sample available per word. Though this is

in the same line as the zero-resource speech processing papers, we feel the motivation behind the decomposition is very similar to this work.

Another group of researchers explore adaptation techniques for multilingual speech recognition, especially for low resource languages. In these multilingual settings, the hidden layers are either HMM or DNN models which are shared by multiple languages, and the output layer is either language specific phone set or a universal IPA-based phone set (Tong et al., 2017; Vu and Schultz, 2013; Thomas et al., 2010; Chen and Mak, 2015; Dalmia et al., 2018). However predictable phonemes are restricted to the phonemes in the training set, thus they fail to predict unseen phonemes in the test set. In contrast, our model can predict unseen phonemes by taking advantage of their articulatory attributes.

Articulatory features have been shown to be useful in speech recognition under several situation. Specifically, articulatory features has been used to improve robustness under noisy and reverberant environment (Kirchhoff, 1998), compensate for crosslingual variability (Stüker et al., 2003b), improve word error rate in multilingual models (Stüker et al., 2003a), be beneficial for low resource languages (Müller et al., 2016), detecting spoken words (Prabhavalkar et al., 2013), clustering phoneme-like units for unwritten languages (Müller et al., 2017a), recognizing unseen languages (Siniscalchi et al., 2011), developing phonological vocoder (Cernak and Garner, 2016). There are also some attempts to predict articulatory features or distributions for clinical usages (Jiao et al., 2017; Vásquez-Correa et al., 2019), but they do not provide a model to predict unseen phonemes.

We note that there are also several attempts to build acoustic models for unseen phonemes. For example, the authors in (Scharenborg et al., 2017) present an interesting method to predict unseen phonemes in Mboshi by mapping Dutch/Mboshi phonemes in the same space using an extrapolation approach. However starting phonemes used for extrapolation had to be manually assigned for every missing phoneme and every pair of languages. Compared with this work, our model proposes a much more generic algorithm to recognize unseen phonemes. Another previous work integrated articulatory attributes into the state-position based decision tree to predict unseen phones in their multilingual model (Knill et al., 2014), however the approach is limited to traditional HMM models and it is unclear how attributes are extracted and how it performs when predicting unseen phonemes.

## 2.5 Conclusion

In this work, we propose the Universal Phonemic Model to apply zero-shot learning to the automatic phonemic transcription task. Our experiment shows that it outperforms the baseline by 7.7 % phoneme error rate on average for 7 languages. While the performance of our approach is

still not enough for the real-world production systems, it paves the way to tackle zero-shot learning of speech recognition with a new framework.

# Chapter 3

# Universal Phone Recognition with Phones and Phonemes

## Summary

Multilingual models can improve language processing, particularly for low resource situations, by sharing parameters across languages. The model introduced in the previous chapter attempts to share information across languages by using the phonological features. This model and many previously proposed model generally ignore the difference between phonemes (sounds that can support lexical contrasts in a *particular* language) and their corresponding phones (the sounds that are actually spoken, which are language independent). This can lead to performance degradation when combining a variety of training languages, as identically annotated phonemes can actually correspond to several different underlying phonetic realizations.

In this chapter, we introduce the concept of *allophone* and propose a joint model of both language-independent phone $Q$ and language-dependent phoneme $P$ distributions. In multilingual ASR experiments over 11 languages, we find that this model improves testing performance by 2% phoneme error rate absolute in low-resource conditions. Additionally, because we are explicitly modeling language-independent phones, we can build a (nearly-)universal phone recognizer that, when combined with the PHOIBLE (Moran and McCloy, 2019) large, manually curated database of phone inventories, can be customized into 2,000 language dependent recognizers. Experiments on two low-resourced indigenous languages, Inuktitut and Tusom, show that our recognizer achieves phone accuracy improvements of more than 17%, moving a step closer to speech recognition for all languages in the world.[1]

[1]A web demo is available at `https://www.dictate.app`, the pretrained model is released at `https://github.com/xinjli/allosaurus`

Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anas-tasopoulos, David R-Mortensen, Graham Neubig, Alan W Black, et al. 2020. Universal phone recognition with a multilingual allophone system. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (Chapter 3)

## 3.1 Introduction

There is an increasing interest in building speech tools benefiting low-resource languages, specifically multilingual models that can improve low-resource recognition using rich resources available in other languages like English and Mandarin. One standard tool for recognition in low resource languages is multilingual acoustic modeling (Dalmia et al., 2018). Acoustic models are generally trained on parallel data of speech waveforms and *phoneme* transcriptions. Importantly, phonemes are perceptual units of sound that closely correlate with, but do not exactly correspond to the actual sounds that are spoken, *phones*. An example of this is shown in Figure 3.1, which demonstrates two English words that share the same phoneme /p/, but different in the actual phonetic realizations [p] and [p$^{h}$]. *Allophones*, the sets of phones that correspond to a particular phoneme, are language specific; distinctions that are important in some languages are not important in others.

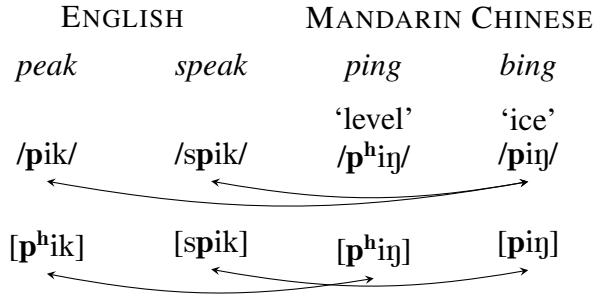



Figure 3.1: Words, phonemes (slashes), and phones (square brackets).

Most multilingual acoustic models simply use existing phoneme transcriptions as-is, taking the union of the phoneme sets to be shared by all training languages (Lin et al., 2009; Cohen et al., 1997; Schultz and Waibel, 2001, 1997; Li et al., 2020d). The assumption is reasonable under some circumstances as phoneme names are typically associated with their most common or least marked allophone. However, this is obviously an over-simplistic view: in Figure 3.1, for example, this would mean that all training in English would assign the phones [p] and [p$^{h}$] to phoneme /p/. This is detrimental if we want to recognize Mandarin Chinese, for instance, where the two phones are corresponding to two distinctive phonemes /p/ and /p$^{h}$/.

In this chapter, we propose a novel method for multilingual recognition based on phonetic annotation to tackle this problem: *Allosaurus* (**allo**phone **s**ystem of **au**tomatic **r**ecognition for **u**niversal **s**peech). Our method incorporates knowledge of phonology into the multilingual model through an *allophone layer*, which associates a universal narrow phone set with the phonemes that appear in the transcription of each language. Our model first computes the phone distribution using a standard ASR encoder, then the allophone layer maps the phone distribution into the phoneme distribution for each language. This model can be trained end-to-end using only standard phonemic transcriptions and an allophone list created by phoneticians. The allophone layer is first initialized with the allophone list, then is further optimized during the training process. We demonstrate that accounting for the phoneme-phone mismatch in this way improves the accuracy of multilingual acoustic modeling by 2.0% phoneme error rate in low-resource conditions.

Furthermore, the architecture simultaneously makes it possible to perform *universal phone recognition*. Previous approaches cannot perform phone recognition in a universal fashion as they depend on language-specific phonemes, as illustrated with the previous example of English not distinguishing /p/ and /p $^{h}$/ as required in Mandarin. In contrast, because our approach allows recognition of phones directly, it already has learned to make these fine-grained distinctions. Taking advantage of this fact, we incorporate a large phone inventory database collected by linguists, PHOIBLE (Moran and McCloy, 2019), and demonstrate that our phone recognizer can be customized to recognize over 2000 languages without any training data in the languages themselves. By evaluating the recognizer with completely unseen testing languages, we found that our recognizer achieves 17% better performance absolute compared with the traditional approach.

## 3.2 Related Work

While some recent work in multilingual ASR focuses on end-to-end models to directly predict graphemes (Watanabe et al., 2017; Toshniwal et al., 2018), most systems still depend on phonetically inspired acoustic models. Multilingual acoustic models fall into two groups. The first group, *shared phoneme* models, creates a shared phoneme inventory of all phonemes from all training languages (Lin et al., 2009; Cohen et al., 1997; Schultz and Waibel, 2001, 1997; Li et al., 2020d; Thompson et al., 2019). The second group, *private phoneme* models, treats phonemes from each language as completely different classes performs phoneme classification separately for each language (Huang et al., 2013; Dalmia et al., 2018; Li et al., 2019a). However, these two groups have their own respective drawbacks: the first group fails to consider the disconnect between the phonemes across languages while the second group completely ignores cross-lingual phonetic associations and is not applicable to recognition of new languages. In contrast, our approach solves both of these problems by taking into account allophones with phone-phoneme mappings.

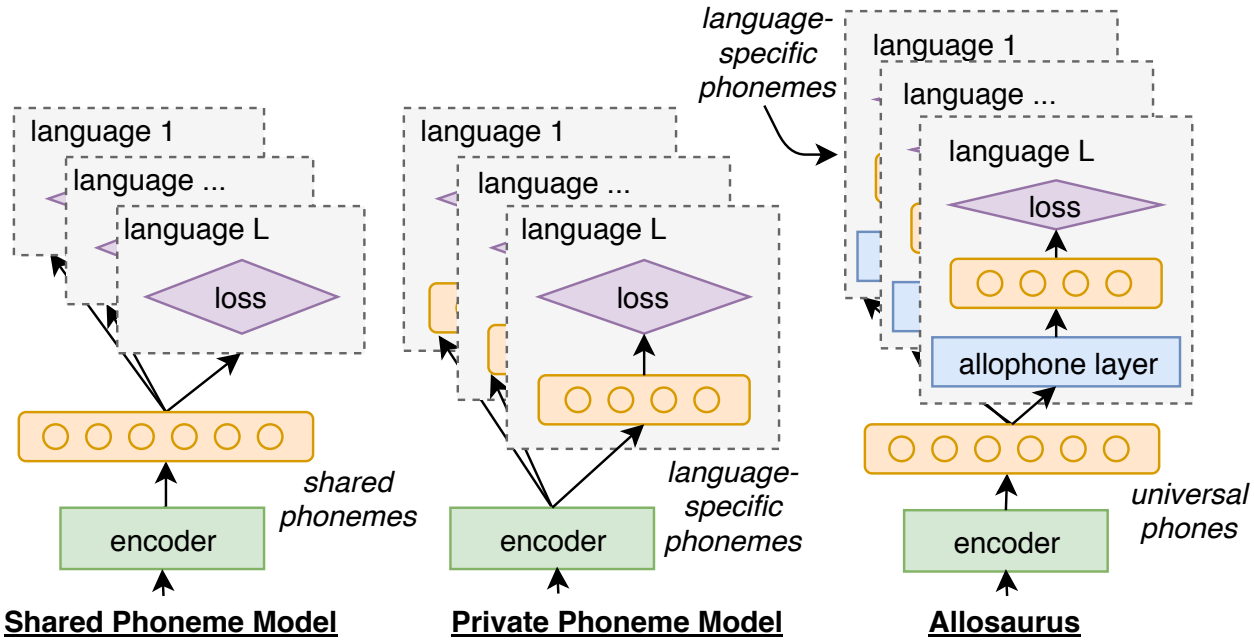



Figure 3.2: Traditional approaches predict phonemes directly, either for all languages (left) or separately for each language (middle). On the contrary, our approach (right) predicts over a shared phone inventory, then maps into language-specific phonemes with an allophone layer.

There have been some attempts to apply phone recognizers to low resource languages. For example, English recognizers have been applied to align transcription corpora of an endangered language (DiCanio et al., 2013), facilitate language documentation (Michaud et al., 2018), identify languages with language models (Matejka et al., 2005), and perform linguistic annotation (Neubig et al., 2018). However, these approaches depend heavily on training data in the language of interest and their specific phonemic transcriptions. Our approach, on the other hand, abstracts away the dependency to phonemes by applying the allophone transformations.

## 3.3 Approach

### 3.3.1 Phone-Phoneme Annotation

Suppose there are $|L|$ training languages, and each language $L_i$ has its own phoneme inventory $P_i$ which can be easily obtained by enumerating the phonemes appearing in its annotated training data. Most traditional multilingual approaches handle inventories at the phoneme level, and create a *shared phoneme inventory* $P_{\text{sha}}$ by taking union of the phoneme sets:

Table 3.1: Results of three models' phoneme error rate performance on 11 languages. The top-half shows the results trained with all training datasets. The bottom-half shows the low-resource results in which only 1k utterances are used for training from each dataset.

| | | Amh | Eng | Ger | Ita | Jap | Man | Rus | Spa | Tag | Tur | Vie | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Full** | **Shared Model** | 78.4 | 71.7 | 71.6 | 62.9 | 65.9 | 76.5 | 76.9 | 62.6 | 74.1 | 76.6 | 82.7 | 73.8 |
| | **Private Model** | 37.1 | 22.4 | 17.6 | 26.2 | 17.6 | 17.9 | 21.3 | 18.5 | 47.6 | 35.8 | 56.5 | 25.6 |
| | **Allosaurus** | 36.0 | 20.5 | 18.8 | 23.7 | 23.8 | 17.0 | 26.3 | 19.4 | 57.4 | 35.3 | 57.3 | 25.0 |
| **Low** | **Shared Model** | 80.4 | 73.3 | 74.3 | 72.2 | 77.1 | 83.0 | 83.2 | 72.8 | 84.8 | 84.4 | 84.5 | 78.4 |
| | **Private Model** | 55.4 | 50.6 | 41.9 | 31.6 | 36.8 | 37.0 | 47.9 | 36.7 | 62.3 | 54.5 | 73.6 | 43.8 |
| | **Allosaurus** | 54.8 | 47.0 | 41.5 | 37.4 | 40.5 | 33.4 | 45.0 | 35.9 | 70.1 | 53.6 | 72.5 | 41.8 |

$$P_{\text{sha}} = \bigcup_{1 \le i \le |L|} P_i \tag{3.1}$$

In contrast, our method distinguishes phonemes from their phone realizations. We have linguists annotate each phoneme $p \in P_i$ with its corresponding allophone set $Q_p^i$, where each phone $q \in Q_p^i$ is a realization of $p$ in language $L_i$.

Merging these sets for all languages, we obtain the *universal phone inventory* $Q_{\text{uni}}$.

$$Q_{\text{uni}} = \bigcup_{1 \le i \le |L|} \bigcup_{p \in P_i} Q_p^i \tag{3.2}$$

Additionally, we obtain a *signature matrix* $S^i = \{0,1\}^{|P_i| \times |Q_{\text{uni}}|}$ describing the association of phone and phonemes in each language $L_i$: Suppose the phoneme $p \in P_i$ has the row index $j$ where $1 \le j \le |P_i|$ , phone $q \in Q_{\text{uni}}$ has the column index $k$ where $1 \le k \le |Q_{\text{uni}}|$, if the $q$ is a realization of $p$, then $(j,k)$ cell of the $S^i$ has a value of 1, otherwise it is assigned 0.

### 3.3.2 Allophone Layer

As mentioned in Section 3.2, traditional multilingual models can be divided into two groups. The first group, *shared phoneme models* (Figure 3.2 left), predicts phoneme distributions over the shared phoneme inventory $P_{\text{sha}}$. The second group, *private phoneme models* (Figure 3.2 middle), on the other hand, shares a common encoder but computes distribution over private phoneme inventory $P_i$ for each language $L_i$. These approaches handle phonemes directly with no concept of underlying phones.

Table 3.2: Training corpora and size in utterances for each language. Models are trained and tested with 12 rich resource languages (top) and 2 low resource unseen languages (bottom).

| **Language** | **Corpora** | **Utt.** |
|---|---|---|
| English | voxforge, Tedlium (Rousseau et al., 2012), Switchboard (Godfrey et al., 1992) | 1148k |
| Japanese | Japanese CSJ (Maekawa, 2003) | 440k |
| Mandarin | Hkust (Liu et al., 2006), openSLR (Hui Bu, 2017; Dong Wang, 2015) | 377k |
| Tagalog | IARPA-babel106b-v0.2g | 93k |
| Turkish | IARPA-babel105b-v0.4 | 82k |
| Vietnamese | IARPA-babel107b-v0.7 | 79k |
| German | voxforge | 40k |
| Spanish | LDC2002S25 | 32k |
| Amharic | openSLR25 (Abate et al., 2005) | 10k |
| Italian | voxforge | 10k |
| Russian | voxforge | 8k |
| Inukitut | private | 1k |
| Tusom | private | 1k |

In contrast our proposed approach, *Allosaurus*, (Figure 3.2 right), comprises a language independent encoder and phone predictor, and a language dependent allophone layer and a loss function associated with each language. The encoder first produces the distribution $h \in \mathbb{R}^{|Q_{\text{uni}}|}$ over the universal phone inventory $Q_{\text{uni}}$, then the allophone layer transforms $h$ into phoneme distribution $g^i \in \mathbb{R}^{|P_i|}$ of each language. The allophone layer uses a trainable allophone matrix $W^i \in \mathbb{R}^{|P_i| \times |Q_{\text{uni}}|}$ to describe allophones in the similar way as $S^i$. The allophone matrix $W^i$ is first initialized with $S^i$, and is allowed to be optimized during the training process, but we add an L2 penalty to penalize divergence from the original signature matrix $S^i$. The allophone layer computes its logit distribution $g^i$ by finding the most likely allophone realization in $Q_{\text{uni}}$ with maxpooling.

$$g^i_j = \max(\{w^i_{j,k} \cdot h_k; 1 \leq k \leq |Q_{\text{uni}}|\}), \tag{3.3}$$

where $g^i_j \in \mathbb{R}$ is the logit of $j$-th phoneme in $g^i$ for language $L_i$, $w^i_{j,k} \in \mathbb{R}$ is the $(j,k)$ cell of the allophone matrix $W^i$, $h_k \in \mathbb{R}$ is the logit of $k$-th phone in $h$. Intuitively, if the $j$-th phoneme has the $k$-th phone as an allophone, $w^i_{j,k}$ would be near 1, otherwise $w^i_{j,k}$ would be near 0. Therefore, the phoneme logit of $g^i_j$ is decided by the largest allophone logit $h_k$. The phoneme distribution $g^i$ is further fed into the loss function. This method for phoneme prediction can be used with any underlying multilingual ASR system. Here we specifically optimize the parameters by minimizing CTC loss (Graves et al., 2006) for all training languages, with the addition of regularization of the allophone layer controlled by hyperparameter $\alpha$.

$$\mathcal{L} = \sum_{1 \leq i \leq |L|} (\mathcal{L}^i_{ctc} + \alpha W^i - {S^i}_2^2). \tag{3.4}$$

### 3.3.3 Universal Phone Recognition

Not only does the allophone layer abstract away from the language-specific phonemes, which contributes to the improvement in the multilingual acoustic modeling, the model also gives us the capability to predict universal phones themselves. This has rarely been attempted in previous work. By applying the greedy decoding strategy over the phone distribution $h$, we can obtain a phone sequence in which all phones $Q_{\text{uni}}$ in the training languages are candidates. When combined with a large training languages sets, our universal inventory is expected to cover most common narrow phones appearing in many languages in the world, which we show in the experiment section.

Furthermore, this recognition protocol can take into account phone inventories that have already been created for many languages in the world by linguists. For example, PHOIBLE (Moran and McCloy, 2019) is a database of phone inventories for more than 2000 languages and dialects, allowing our model to be applied to these languages with some degree of accuracy. If the phone inventory for language $L_i$ is $Q_i$, we can restrict the decoder to only produce phones in $Q_i \cap Q_{\text{uni}}$ by filtering out other phones. When the universal inventory $Q_{\text{uni}}$ covers most frequent phones in the world, we could expect that $Q_i \approx Q_i \cap Q_{\text{uni}}$.

## 3.4 Experiments

### 3.4.1 Settings

As we are interested in creating a large universal phone inventory, we select a phonetically diverse set of 11 training languages as summarized on the top of Table 4.1. We include corpora from a variety of speech domains to make our model robust (e.g., read speech, sponatenuous speech). 5% of the dataset is used as the test set, and the remaining data are used as the training set and the validation set. We also consider a low resource condition, where 1,000 random utterances are used from each corpus to train the model. As baselines, we compare with the previously-described *shared phoneme* and *private phoneme* models. All methods use the same encoder and features. Features are high-resolution 40 dimensional MFCCs extracted with Kaldi (Povey et al., 2011). The encoder is a 6-layer stacked bidirectional LSTM with hidden size of 1024 in each layer. The regularization hyperparameter $\alpha$ is set to 10. Phonemes for training languages are assigned using the grapheme-to-phoneme tool Epitran (Mortensen et al., 2018). For each phoneme in each

Table 3.3: Statistics of the phone coverage mean (standard deviation) of areas. Phone coverage of language $L_i$ is defined as $\frac{|P_{\text{uni}} \cap P_i|}{|P_i|}$

| Area | # Language | Shared | Allosaurus |
|---|---|---|---|
| Africa | 875 | 53% (13%) | 84% (11%) |
| America | 659 | 52% (14%) | 81% (13%) |
| Asia | 377 | 46% (15%) | 79% (13%) |
| Pacific | 152 | 59% (15%) | 87% (12%) |
| Europe | 92 | 35% (9.5%) | 69% (13%) |
| All | 2155 | 52% (15%) | 82% (13%) |

Table 3.4: Comparisons of phone error rates in two unseen languages

| | Inuktitut | Tusom |
|---|---|---|
| **Shared Phoneme PER** | 94.1 | 93.5 |
| **Best Private Phoneme PER** | 86.2 | 85.8 |
| **Allosaurus PER** | 84.1 | 77.3 |
| **Allosaurus+PHOIBLE PER** | 73.1 | 64.2 |

language, phoneticians create the allophone mappings. (Mortensen et al., 2020)[2]

We evaluate using phoneme error rate for the training languages. Furthermore, we select two languages not included in the training data: Inukitut and Tusom. These languages are indigenous languages with few training resources, representing a realistic scenario where our model is applied to entirely new languages, as may be done when ASR is used for documentation of endangered languages. The datasets of these two languages are transcribed with phones, and accordingly we use phone error rate rather than the phoneme error rate. While Allosaurus is able to predict phones in a natural way by decoding $h$, the two baselines could not predict phones directly. In this unseen language experiment, we assume phonemes predicted by the baselines correspond to phones of the same name.

### 3.4.2 Main Results

Table 3.1 demonstrates the performance of the baseline models and Allosaurus evaluated on 11 languages. The top half of the table summarizes the performance when trained with the full training

[2]its database is available at `https://github.com/dmort27/allovera`

Table 3.5: An English example from switchboard in which Allosaurus could distinguish [$p^h$] and [p] for phoneme /p/

| Model | Phones |
|---|---|
| **Utterance** | the quebec **people** that that **speak** french |
| **Annotation** | /ð ə k w ə b ɛ k **p** i p ə l .. s **p** i k f ɹ ɛ n ʧ/ |
| **Allosaurus** | [ð ə x o b ə k ɐ **$p^h$** i θ oː l .. s **p** ɹ ɪ k f ɹ ɛ n d] |

set. The results suggests both the private phoneme model and the Allosaurus model outperforms the shared phoneme model significantly. The results of the shared phoneme model can be explained by the disagreement of phoneme assignments across languages. In contrast, the private phoneme model handles this issue by using language specific phoneme layers. Our model also circumvents this issue by introducing the language-specific allophone layers. The bottom half of the Table 3.1 highlights the results when the training set of each language is limited as mentioned above. Unsurprisingly, limiting the amount of training data hurts accuracy across the board. While the private phoneme model and our model achieve similar results when using the full training set, our model outperforms the private phoneme model by 2.0% when training data is limited. This suggests that our model is better at sharing parameters across languages by using prior phonetic knowledge in this case, likely due to the fact that the private phoneme model needs to learn each phoneme predictor from scratch, while our model already has phone-phoneme mapping knowledge seeded by linguistically motivated annotations.

### 3.4.3 Universal Phone Recognition Results

In addition to the improvements over low resource settings, our model enables us to predict (nearly-)universal phone distributions. By merging phone inventories from all of our languages, we obtain a shared inventory of 187 phones. First, we assess how close this inventory gets to covering the languages registered in PHOIBLE. The Allosaurus column in Table 3.3 summarizes the phone coverage of our model, split into different geographic areas. The phone coverage in each cell represents the mean and standard deviation for each category. As the table suggests, our model has a promising phone coverage over all areas consistently. On average, it has 82% mean phone coverage and 12.8% standard deviation over all PHOIBLE languages. Furthermore, by comparing our model with the baseline model in which we merged all the phoneme inventories from the corpus as-is, we significantly improve the phone coverage by 30%. Additionally, the standard deviation shows that our model covers phones more consistently than the baseline model.

Next, we actually evaluate the model with respect to its ability to recognize phones. Table 3.5

Table 3.6: A qualitative example from Inuktitut dataset

| Model | Phones |
|---|---|
| **Ground Truth** | [i l i t s i lː i] |
| **Allosaurus** | [e l e p ʁ ɪ lː e] |
| **Allosaurus+PHOIBLE** | [i l i t i lː iː] |

shows a decoded English example. The utterance contains three English phonemes /p/ in word *people* and *speak*. The underlying allophones, however, are [$p^h$] and [p] as mentioned in Section 1. While the original English training set annotates those two words with the same phoneme /p/, Allosaurus is able to predict different allophones by leveraging knowledge from other languages (e.g: Mandarin). We also note that Allosaurus is still not perfect: it fails to recognize the second /p/ in "people."

Additionally we also investigate unseen languages on the Inuktitut and Tusom datasets. The results are summarized in the Table 3.4. As the result show, the shared phoneme model can hardly recognize any phonemes in these two languages, with more than 90.0% phone error rate on both datasets. Next, we try all 11 private phoneme models from the training datasets and use the one with the lowest phoneme error rate. Unsurprisingly, this also can not achieve satisfying results on both datasets, as none of our 11 languages is similar to Inuktitut and Tusom; they both have over 85.0% phone error rate. On the other hand, the proposed Allosaurus model achieves 84.1% phone error rate on Inuktitut and 77.3% phone error rate on Tusom, a significant drop. When combined with the PHOIBLE inventory, the error rates are further improved to 73.1% and 64.2% respectively, which shows 17% improvements on average over the shared phoneme baseline. Table 3.6 shows one qualitative example from Inuktitut data. It suggests that simply applying Allosaurus could capture some aspects of the target phonemes, but it still made many errors especially substitution errors between [e] and [i]. The reason is Allosaurus has a much broader phone search space (187 phones), it might be difficult to distinguish similar phones (e.g: both [e] and [i] are front vowels, but [e] is a close vowel and [i] is a close-mid vowel). We find those substitution errors account for the majority of errors in the test sets. Those confusing phones, however, might be solved when combined with an appropriate inventory such as PHOIBLE. The last row suggests that Allosaurus could fix those substitution errors as [e] does not exist in Inuktitut's inventory.

## 3.5 Conclusion

In this chapter, we propose *Allosaurus*, which considers the relationship between phones and phonemes in multilingual acoustic modeling. It improves significantly the phone recognition ac-

curacy over unseen languages by 17%.

# Chapter 4

# Acoustic Model: Hierarchical Multilingual Model

## Summary

There is growing interest in building phone recognition systems for low-resource languages as the majority of languages do not have any writing systems. Phone recognition systems proposed so far typically derive their phone inventory from the training languages, therefore the derived inventory could only cover a limited number of phones existing in the world. It fails to recognize unseen phones in low-resource or zero-resource languages.

Chapter 2 and Chapter 3 discuss two different multilingual acoustic model. In Chapter 2, each phoneme is represented using phonological features, in Chapter 3, each phoneme is represented using its corresponding phones (i.e. allophones). This chapter combines those two architectures into a single hierarchical model: we explicitly model three different entities in a hierarchical manner: *phoneme*, *phone*, and *phonological articulatory attributes*. In particular, we decompose phones into articulatory attributes and compute the phone embedding from the attribute embedding. The model would first predict the distribution over the phones using their embeddings, next, the language-independent phones are aggregated to the language-dependent phonemes and then optimized by the CTC loss. This compositional approach enables us to recognize phones even they do not appear in the training set. We evaluate our model on 47 unseen languages and find the proposed model outperforms baselines from Chapter 2 and Chapter 3 by 13.1% PER.

Xinjian Li, Juncheng Li, Florian Metze, and Alan W Black. 2021. Hierarchical phone recognition with compositional phonetics. In Proc. Interspeech

## 4.1 Introduction

With the development of deep neural networks, there is growing interest in applying deep neural network models to speech recognition (Amodei et al., 2016; Chiu et al., 2018; Chan et al., 2016). Those deep models, however, are restricted to languages with a large amount of training set such as English and Mandarin (Godfrey et al., 1992; Panayotov et al., 2015), therefore, they are not available for most languages in the world. Additionally, the majority of the languages in the world have never been written (Coulmas, 2013), as a result, the only accessible speech recognition systems are phone recognition systems. Many works have focused on developing phone recognition systems for low-resource languages (Schultz and Waibel, 1997; Li et al., 2020d; Thompson et al., 2019; Li et al., 2019a). However, most of them face the problem of the limited phone inventory. As the training languages typically consist of rich resource languages such as English and Mandarin, the training phone inventory usually consists of common phones available in European languages and East-Asian languages. This situation makes it hard to recognize unique phones in other language families. Another problem is the imbalanced phone distribution among the training set: some phones might appear frequently in many languages, but other phones might only occur in limited cases in one specific training language and therefore have much fewer training samples. This issue would cause the model to predict the first group more frequently and suppress the second group. Note that we distinguish the concept of *phone* and *phoneme* in this work (Ladefoged and Johnson, 2014): *phone* represents the physical speech sound, it is the language-independent unit shared by all languages. In contrast, *phoneme* is the language-dependent unit, it is the smallest unit to distinguish meaning in a specific language. Phones and phonemes are highly related to each other and one phoneme might correspond to multiple phones (those phones are referred to as the *allophones*). For example, the phoneme /p/ in English have two actual phonetic realizations (allophones) [p] and [$p^h$].

In this chapter, we propose a novel hierarchical model to tackle the two problems stated above. While most traditional works tend to consider each phone as the basic independent building block, we further decompose phones into their components: phonological articulatory attributes. For instance, the phone [a] can be characterized as a *open front unrounded vowel* where each word (e.g: *open*) can be seen as its attribute. We assign each attribute an *attribute embedding* to encode its information, then the *phone embedding* can be constructed by summing up its corresponding attribute embeddings. Those embedding would be fine-tuned during the training process. With those embeddings, we can build the recognition model as illustrated in Figure.9.1: the encoder (BLSTM) first receives the input features and generates hidden vectors. We take the inner product of the phone embedding and the hidden vector to compute the phone distributions. Then the phone distribution is mapped to phonemes in each language using the allophone mappings. Finally, the phoneme distribution is optimized by the loss (CTC) function. This approach enables us to

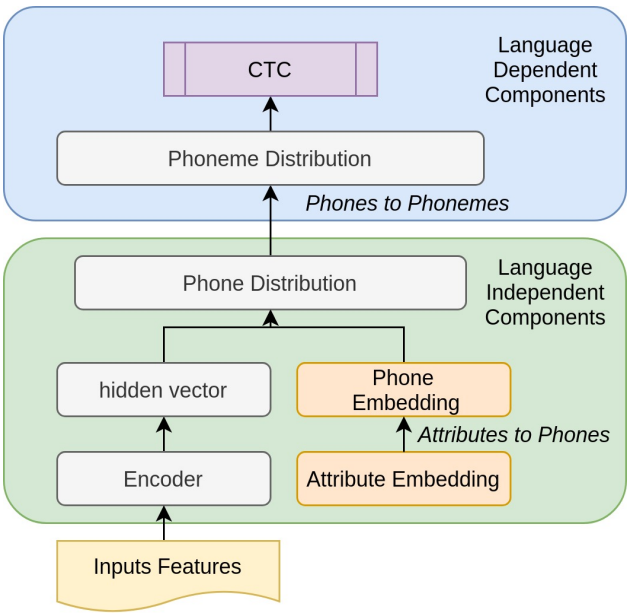



Figure 4.1: The architecture of the hierarchical model. We first compose the phone embeddings from their attribute embeddings. Then we compute the phone distributions using the embeddings and the hidden vector from the encoder, Next, the language-independent phones are transformed into language-dependent phonemes with the allophone mappings, which would finally be optimized by the loss (CTC) function.

solve the aforementioned two problems: first, the phones are no longer independent units, they are interconnected by articulatory attributes shared with each other. Even for a new phone which we have not encountered so far, we can decompose it into existing attributes and then compute its embedding as well. Therefore, this model has the ability to handle unseen phones. Furthermore, this model would suffer less from the imbalanced phone distribution problem as we are optimizing the attribute embeddings instead of the phones themselves: even the rare phones would be fully trained through their attributes shared with other frequent phones. We apply our models to 47 unseen languages and the results indicate that our model improves the average PER (Phone Error Rate) by 13.1%.

## 4.2 Related Work

Training speech recognition systems for low-resource languages remains a challenge due to the limited supervised training set. One common approach is to train multilingual models on languages with rich supervised resources and then transfer its knowledge to new languages (Huang et al., 2013; Heigold et al., 2013; Veselỳ et al., 2012). Another promising method proposed recently is to use the unsupervised approach to pretrain the encoder with a large amount of unsupervised dataset. The pretrained encoder can be fine-tuned to the target language with a limited size of training set (Schneider et al., 2019; Conneau et al., 2020; Rivière et al., 2020).

Despite the success of those models, they still rely on the supervised set for the target language and could not be applied to any unseen languages. In particular, the unseen language might contain unseen phones which are not available in the training languages. One solution is to use language-independent phones instead of the language-dependent phonemes or subwords (Li et al., 2020a). During the training phase, the model can first predict the distribution over the language-independent phones, then it transforms the distribution into the language-dependent phonemes to be optimized (as most of the training set is typically available in the form of phonemes). As the language-independent units are shared by all languages, this model can be applied to unseen languages without any training set for the target language. The only information required for such a model is the phone inventory, which is easy to obtain as PHOIBLE has published the inventory containing more than 2000 languages (Moran and McCloy, 2019). However, even this model cannot solve the issues of unseen phones as the available phone inventory is limited to the phones covered by the training languages. Additionally, it suffers from the problem of the imbalanced phone distribution we mentioned above.

One potential approach to overcome those two problems is to use phonological articulatory attributes. The articulatory attributes are well-defined by the linguists and most phones can be reduced to a list of discrete articulatory attributes (Ladefoged and Johnson, 2014). By learning

the representations over the articulatory attributes, we can associate any unseen phones with well-known attributes and therefore be available to use those phones during inference. Note that applying articulatory attributes to speech recognition tasks is not a new idea. To name a few, it has been applied to improve robustness under the noisy environment (Kirchhoff, 1998), improve performance for multilingual speech recognition (Müller et al., 2016), doing phoneme clustering for unwritten languages (Müller et al., 2017a). However, most works do not apply them to predict unseen phones. One work has applied a similar idea to recognize unseen phones as ours (Li et al., 2020c). This work, however, does not distinguish between phones and phonemes, it constructs the language-dependent phonemes directly from the articulatory attributes. We find this model would not be properly trained when the number of training language increases because more languages would bring more phone-phoneme inconsistencies.

## 4.3 Approach

### 4.3.1 Compositional Phonetics

In this work, we introduce the approach of *compositional phonetics*, where we decompose phones into a list of phonological articulatory attributes. Each attribute has been assigned a fixed length of embedding which we refer to as the *attribute embedding*, those embeddings are first randomly initialized and get fine-tuned together with other parameters during the training process. By using those attribute embeddings, each phone can also be assigned an embedding by linearly composing the embedding from their attributes. Formally, consider a set of phones $P$, for each phone $p \in P$, we could determine a list of its attributes $A_p$. For each attribute in the list $a \in A_p$, we could assign an attribute embedding $e_a \in \mathbb{R}^n$ where $n$ is the hidden size of the model. Then, the phone embedding $e_p \in \mathbb{R}^n$ can be computed by aggregating its attribute embeddings.

$$e_p = \sum_{a \in A_p} e_a \tag{4.1}$$

Suppose that the encoder computes the hidden vector $h \in \mathbb{R}^n$ for the current frame, we can obtain the logit $l_p$ for this phone $p$ by taking inner product

$$l_p = h^T e_p \tag{4.2}$$

Note that the embedding composition approach is not the only way to associate attributes and phones. A more simple idea used in (Li et al., 2020d) is to first compute the attribute logits $\mathbf{l}_A \in \mathbb{R}^{|A|}$ from the encoder, where $|A|$ is the size of entire attributes, then add up logits of corresponding attributes. We would refer to this model a linear model.

$$l_p = \sum_{a \in A_p} l_a \tag{4.3}$$

While the two approaches seem to be similar, we find that the embedding composition approach is more stable and typically leads to better performance. Our hypothesis is that the linear model encodes the hidden information with a small size of $|A|$, on the contrary, the embedding approach encodes the information with a much larger hidden size $n$ and thus has better expressive power (in our experiment, $n = 640, |A| = 23$). Additionally, the embedding approach enables us to have a better understanding of the model through their embedding spaces.

Notice that while the potential number of phones is very large, the number of articulatory attributes is significantly smaller. In our estimation, we find PHOIBLE has listed more than 2000 unique phones across all registered languages (Moran and McCloy, 2019), however, the articulatory phonological attributes are well-defined and we only consider 22 unique attributes (+1 ctc attribute) in this work. Even if a particular phone does not exist in the training set, we can still do the inference as we can easily compose its embedding from the known attribute embeddings.

### 4.3.2 Allophone Layer

We review the idea of the allophone layer from Chapter 3. The allophone layer is to transform the language-independent phone distributions into the language-dependent phoneme distributions. For the allophone layer, we follow the architecture proposed in the previous work (Li et al., 2020a). Suppose the current language is $L$, and its phoneme inventory is $Q_L$. For each phoneme in the inventory $q \in Q_L$, it has multiple allophones corresponding to it. Suppose the allophone set for $q$ is $P_q$. Then each phone $p \in P_q$ is an allophone for $q$. The allophone layer computes the phoneme logits by selecting the max logits among its allophones.

$$l_q = \max\{l_p | p \in P_q\} \tag{4.4}$$

Finally, the phoneme distributions are fed into the CTC loss to be optimized (Graves et al., 2006). CTC loss is selected as it has the conditional independence assumption, which reduces the dependency to the language modelings of the training languages, and thus make it easier to predict unseen phone sequence patterns.

## 4.4 Experiments

### 4.4.1 Settings

In this section, we describe our experiment in this work. We select 11 training languages as described in Table.4.1. Those languages are selected as they have large training sets and their phonology is well understood. We use Epitran to convert text into phoneme for each utterance in the text (Mortensen et al., 2018). Each phoneme might correspond to several phones, those mapping rules are provided by Allovera (Mortensen et al., 2020). Finally, we extract discrete phonological articulatory attributes from each narrow phone by using Panphon (Mortensen et al., 2016b). The tool supports 22 distinct features, we create two different attributes from each feature by considering whether that feature exists or not. For instance, `+syllabic` means it is a syllabic phone, `-syllabic` means it is not.

For the testing languages, we use a recently proposed dataset (Li et al., 2021c). The dataset contains many small corpora from around 100 languages. Each corpus is phonetically annotated by linguists and manually aligned. We sort all corpus by their size and extract corpus whose size of utterances is larger than 50. The number of unique languages in this subset is 47, their ISO-639 id are abk, ady, afn, afr, agx, ajp, apc, ape, apw, asm, azb, bam, cbv, cpn, dan, ell, fin, guj, hau, haw, heb, hil, hin, hrv, hun, hye, ibb, ilo, isl, kan, kea, khm, klu, knn, lad, lav, lit, lug, mlt, mya, nan, nld, pam, pes, prs, wuu, yue.

For the evaluation, we compare 4 different acoustic models. The first one is the English phone recognition model which is a standard LSTM model trained using only English training sets. This model is used as a baseline to contrast language-dependent models and language-independent models. The second model is the Allosaurus model (Li et al., 2020a) whose architecture has an allophone layer mapping between phones and phonemes, it does not model any articulatory attributes and thus each phone is considered independent from each other. Those two models are open-sourced and available on Github.[1]. The other two models are hierarchical models we propose in this work. One hierarchical model is using a simple linear model mapping articulatory distributions into phone distributions. The other model is the main model we discuss in the previous section where we compose phone embeddings from the attribute embeddings and apply those embeddings to estimate distributions. All 4 models are using the same input feature and same encoder architecture: 40 dimension MFCCs and 5 layer bidirectional LSTM with 640 hidden size, the loss function are all CTC loss. The English model connects the encoder directly to the loss function, the Allosaurus model has an allophone layer between the encoder and loss function, the hierarchical models have the aforementioned compositional architecture.

[1] eng2102 and uni2005 from https://github.com/xinjli/allosaurus

Table 4.1: Training corpora and size in utterances for each language. Models are trained with 11 rich resource languages

| Language | Corpora | Utt. |
|---|---|---|
| English | voxforge, Tedlium (Rousseau et al., 2012), Switchboard (Godfrey et al., 1992) | 1148k |
| Japanese | Japanese CSJ (Maekawa, 2003) | 440k |
| Mandarin | Hkust (Liu et al., 2006), openSLR (Hui Bu, 2017; Dong Wang, 2015) | 377k |
| Tagalog | IARPA-babel106b-v0.2g | 93k |
| Turkish | IARPA-babel105b-v0.4 | 82k |
| Vietnamese | IARPA-babel107b-v0.7 | 79k |
| German | voxforge | 40k |
| Spanish | LDC2002S25 | 32k |
| Amharic | openSLR25 (Abate et al., 2005) | 10k |
| Italian | voxforge | 10k |
| Russian | voxforge | 8k |

### 4.4.2 Results

Table.4.2 shows the main results of our experiment. For each model, we evaluate it across all 47 languages and take the average of their PER (phone error rate). In addition, we also show the percentage of errors of addition, deletion, and substitution. The table indicates that the English model has 72% PER, which is the worst phone error rate among all models. The result is expected as the English model could only recognize phones available in English but is not able to recognize any unseen phones in our testing languages. This also explains the high substitution error rate in English as it typically replaces unknown phones with English phones during inference. The Allosaurus model performs better than the English model as it is a language-independent model and could cover a larger phone inventory. It improves the substitution error rate from 45.6% to 37.8%. Both hierarchical models perform significantly better than the Allosaurus model. The linear model has 57.6% PER and the compositional model has 51.2% PER.

To have a better understanding of performance across languages, Figure.4.2 shows the box plot of the 4 models. It is clear from the figure that each model has a very large variance: some languages perform better and other languages perform worse. By investigating the performance of each language, we find languages with better recording environments tend to obtain better scores, and languages with many background noise tend to score worse. We also compute the correlations across 4 models as shown in Figure.4.3. It demonstrates that Allosaurus model and both hierarchical models are highly correlated, but the English model is much less related. This is because the three models are language-independent models but the English model is language-dependent.

Next, we investigate the most common errors of the embedding model. Table.4.3 shows the

Table 4.2: Average Performance of 47 testing languages for each model. The proposed Hierarchical model using embedding approach performs best. PER is the phone error rate, Add, Del, Sub denotes the addition, deletion and substitution errors. All numbers are shown in %

| Model | PER | Add | Del | Sub |
|---|---|---|---|---|
| English model | 72.0 | 11.2 | 15.2 | 45.6 |
| Allosaurus model | 64.3 | 7.86 | 18.6 | 37.8 |
| Hierarchical (linear) | 57.6 | 7.87 | 13.6 | 36.1 |
| Hierarchical (embedding) | 51.2 | 3.4 | 18.9 | 28.8 |

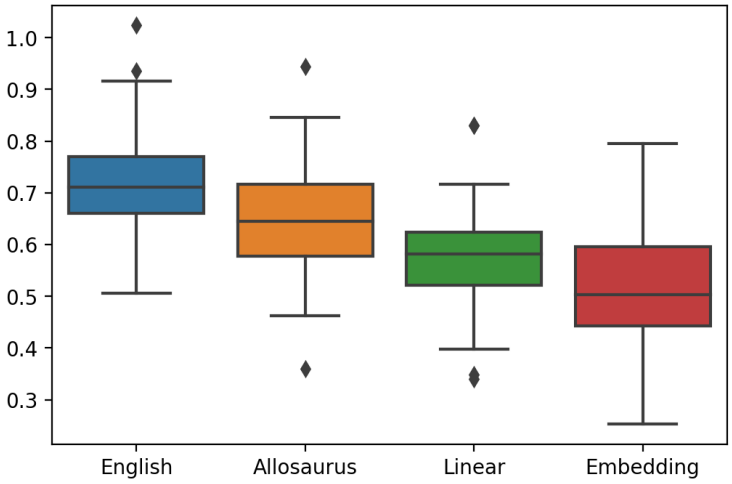



Figure 4.2: The boxplot of performance distribution across all 47 languages for each model

top 3 errors and their occurrences across the dataset. The statistics indicate that the most common error is the deletion of phone [s]. Our hypothesis is that our model might have some difficulties in recognizing unvoiced sounds. For example, [s] is an unvoiced fricative consonant and [t] is an unvoiced plosive consonant. We find those unvoiced sounds typically have some characteristic patterns in the high frequency regions of spectrograms. However, our training set contains many 8k frequency audios and therefore the resolution of our model is restricted to 4k due to the Nyquist sampling theorem. Those deletion errors might be overcome by using high resolution audio corpus in the future. Another major errors come from the substitution errors, they have longer tails than the other two errors. The table suggests that most common substitution errors come from ambiguous vowels.

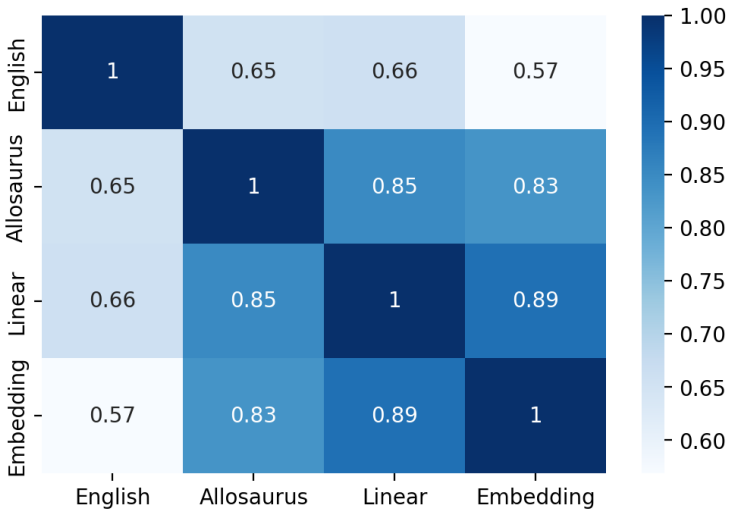



Figure 4.3: Performance correlation between 4 models

Table 4.3: Most frequent errors in the Hierarchical model (embedding), the left side in the tuple is the error and the right side is its total occurrences in the test set. In the substitution row, the phone on the left side is the reference and the phone on the right side is the hypothesis

| Types | Most Common Errors |
|---|---|
| Add | ([i], 104), ([a], 53), ([m], 47) |
| Del | ([s], 247), ([a], 238), ([t], 221) |
| Sub | ([a] -> [], 122), ([u] -> [o], 109), ([a] -> [ɑ], 104) |

### 4.4.3 Analysis of Embeddings

During the training process, we also obtain the embeddings of both articulatory attributes and phones. The attribute embeddings do not have much patterns in them as they are mostly independently from each other. However, the phone embeddings have several interesting patterns. Figure.4.4 shows the embeddings of English phones. The embeddings originally have 640 dimension and get reduced to 2 dimension by PCA. There are several interesting things we can observe in the figure. First, there are a couple of clusters in the graph. The easiest one to identify is the vowel cluster at the right bottom corner. We have vowels such as [a], [o], [u] clustered together. This provides another reason for the substitution error: the embeddings of those phones are near to each other, therefore it is easy to confuse them with each other. On the top of the figure, we have the plosive velar group: [k] and [g]. [ng] near them is another velar consonant. Furthermore, we could find several word2vec like relations (e.g: king - queen = man - woman) in the figure. For

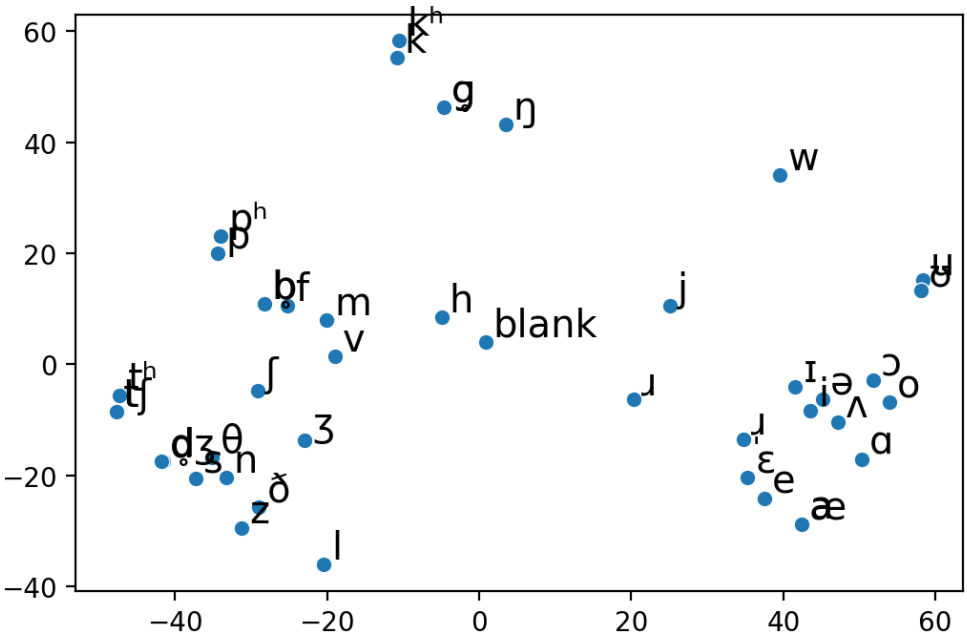



Figure 4.4: PCA projected embeddings for all phones available in English. The embeddings are from the Hierarchical (embedding) model.

example, for voiced and unvoiced sounds,

$$e([k]) - e([g]) = e([p]) - e([b]) \tag{4.5}$$

Similarly, for aspirated and unaspirated sounds, we find following relations:

$$e([p^{\mathrm{h}}]) - e([p]) = e([k^{\mathrm{h}}]) - e([k]) \tag{4.6}$$

## 4.5 Conclusion

In this chapter, we propose the hierarchical model for low-resource phone recognition where we explicitly model three different entities: phoneme, phone and articulatory attributes. We test the model on 47 unseen languages and the result demonstrates that our approach achieves 13.1% PER better than the baseline model. The model will be integrated into the Allosaurus repository for more researchers to explore phone recognition systems.

# Part II

# Language Models

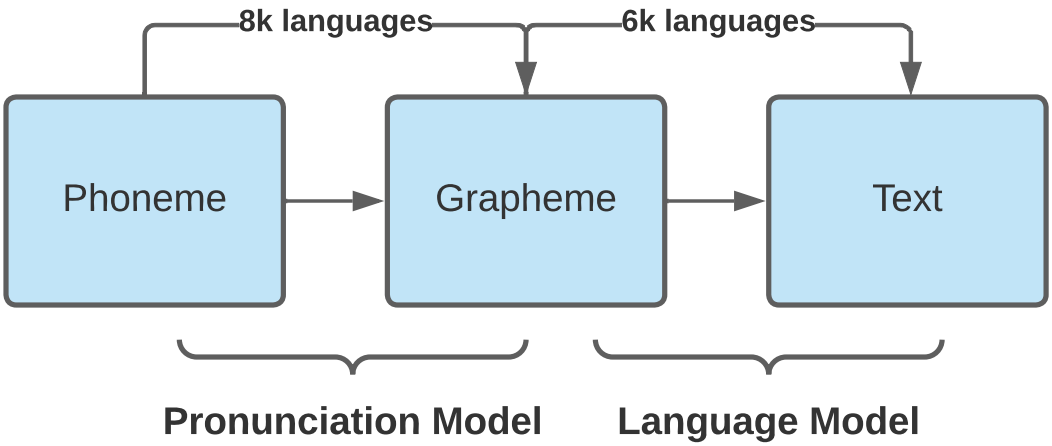



Figure 4.5: The language model covered in Part II

# Chapter 5

# Pronunciation Model: Grapheme to Phoneme Conversion

## Summary

Part I of the thesis covers the acoustic model, which converts the speech audio $X$ into the phoneme transcription $P$. For languages without any writing systems, the phoneme transcription can be the final outputs of the pipeline. If the target language has some writing systems, we need to continue to convert phoneme transcriptions into word sequences. To achieve this conversion, we should establish the connection between phonemes and graphemes for the target language. In particular, we are interested in creating a language-dependent model $P = \delta(Y)$ where $Y$ is a text and $P$ is its phoneme pronunciation. This is the Grapheme-to-Phoneme (G2P) conversion task and it has many applications in NLP and speech fields. Most existing work focuses heavily on languages with abundant training datasets, which limits the scope of target languages to less than 100 languages.

This chapter attempts to apply zero-shot learning to approximate G2P models for all low-resource and endangered languages in Glottolog (about 8k languages). For any unseen target language, we first build the phylogenetic tree (i.e. language family tree) to identify top-$k$ nearest languages for which we have training sets. Then we run models of those languages to obtain a hypothesis set, which we combine into a confusion network to propose a most likely hypothesis as an approximation to the target language. We test our approach on over 600 unseen languages and demonstrate it significantly outperforms baselines.

Xinjian Li, Florian Metze, David R Mortensen, Shinji Watanabe, and Alan W Black. 2022. Zero-shot learning for grapheme to phoneme conversion with language ensemble. Findings of ACL.

| Language | Grapheme | Phoneme |
|---|---|---|
| English | hello | /hlʊ/ |
| Mandarin | 你好 | /nixɑʊ/ |
| French | bonjour | /bɔʒuʁ/ |
| German | hallo | /halo/ |
| Japanese | こんにちは | /konnichiwa/ |
| Spanish | hola | /ola/ |

Table 5.1: A small sample of G2P examples from high-resource languages in our training set.

## 5.1 Introduction

Grapheme-to-Phoneme (G2P) plays a crucial role in many NLP tasks. In particular, it is used heavily in many speech-related tasks such as speech recognition and speech synthesis (Arık et al., 2017; Miao et al., 2015). Even in the latest end-to-end systems, it still has a strong impact on the speech performance (Hayashi et al., 2021). Typically, the G2P task is language-dependent—many language-specific factors affect the G2P process such as the general characteristics of scripts (Ager, 2008), phonotactic constraints (Hayes and Wilson, 2008) and other orthography factors (Frost and Katz, 1992). For example, in Table 5.1, Mandarin and Japanese are not using the Latin script, therefore they cannot share their G2P models with English. As a consequence, to develop a G2P model, we need either to create a training set for the target language, like (CMU, 2000), or to ask linguists to explicitly define a set of orthographic rules to map from graphemes to phonemes (Mortensen et al., 2018). Both approaches have achieved success for high-resource languages; however, they can only account for a small number of the world's languages. The majority still do not have access to G2P due to limited training resources. A good G2P model would be beneficial to many speech tasks in low-resource languages (Li et al., 2020a,c; Yan et al., 2021)

In this work, we attempt to tackle this challenging problem by using the language ensemble approach. Our approach allows us to propose an approximated G2P baseline to all languages present in the GlottoLog database: around 8000 of them (Nordhoff and Hammarström, 2011). The main insight of our approach is that we can approximate the G2P model of an unseen language using those of related languages because languages related to the target language should have similar orthographic rules (of both the context-free and context-dependent type). For example, a native speaker of English (a Germanic language) is likely to make accurate guesses about how a text in German (another Germanic language) would be pronounced. In Table 5.1, both German and English pronounce the "h" grapheme explicitly, but Spanish (a Romance language) does not share the same property. We define the similarity between languages as the shortest distance between

two languages in the phylogenetic tree (i.e. language family tree). We first build models for the subset of languages (training languages) where we have a large enough training set (e.g., Italian, Spanish, etc.). Then, for each unseen language (e.g., Catalan), we first find the top-$k$ nearest training languages (like Italian, Spanish, etc.) and use those languages' G2P models to generate $k$ hypotheses. Finally, we ensemble the G2P outputs by building a confusion network and discover the most-likely sequence as an approximation to the target language.

In our experiments, we build a large dataset from Wiktionary in which we use 260 languages as the training languages and test our approach on 600 unseen languages. We apply our approach to 3 different architectures: a joint-sequence n-gram model (Novak et al., 2016), an LSTM sequence-to-sequence model (Rao et al., 2015), and a transformer-based sequence-to-sequence model (Peters et al., 2017). Using any of the architectures, our approach outperforms all baselines by more than 5% PER (phoneme error rate).

The main contributions of this work are as follows:

1. A novel approach to approximate target language G2P models using the nearest languages in a phylogenetic tree

2. An approach to ensemble predictions from multiple outputs using confusion networks.

3. A demonstration that our approach achieves significantly better performance than baselines when testing on 600 unseen languages.

## 5.2 Related Work

Traditionally, a G2P component is built using rule-based models. For example, the phonological constraints can be incorporated into context-sensitive grammars and implemented using finite-state transducers (Kaplan and Kay, 1994). However, designing the rules requires many hours from linguists and can be prohibitive for low-resource languages if they have deep orthographies[1].

Statistical models overcome this problem by learning the rules automatically. Typically, there are two steps in building such a model: first, the sequence of phonemes and graphemes are aligned to each other, then another prediction model is built on top of the alignment. The alignment model is typically done using Expectation and Maximization (Ristad and Yianilos, 1998; Jiampojamarn and Kondrak, 2010). The prediction model can be done using neural networks (Sejnowski and Rosenberg, 1987), decision trees (Black et al., 1998), joint-sequence models (Bisani and Ney, 2008) and WFST-based n-gram models (Novak et al., 2016). More recently, deep neural networks have been applied to the G2P task. Various architectures have been explored, for example,

[1]Orthographies in which the relationship between graphemes and phonemes has been obscured by history or is otherwise complicated.

RNNs (Rao et al., 2015; Yao and Zweig, 2015; Lee et al., 2020), CNNs (Yolchuyeva et al., 2019) and Transformers (Yolchuyeva et al., 2020).

Traditionally, each G2P model was typically built for one high-resource language. Recently, many researchers have started to focus on low-resource G2P models. One related work adapts high-resource language models to low-resource language models by measuring similarity between languages and phonemes (Deri and Knight, 2016). This previous work creates a new training set for every low-resource language by adapting the training set from the top-3 nearest languages. However, there are several issues with this approach. First, it has to prepare separate training sets and n-gram models for every testing language, which is quite computationally expensive. It also suffers from the limited training set problem even after merging top-3 languages because the vocabulary size of most training languages are less than 100, which is insufficient to train any stable neural models. In contrast, we only prepare one unified training set and one unified model in our neural approach, which circumvents these problems. Additionally, the testing languages and training languages are mixed in this work, therefore the performance on unseen languages is not clear. Only a limited number of papers so far focus on developing G2P models for unseen languages. The most common strategy is to drop the target language information and make predictions using a shared multilingual model (Peters et al., 2017; Bleyan et al., 2019). This is one of our baseline (the global language model) in this work.

## 5.3 Approach

In this section, we describe our zero-shot learning approach. We first introduce three G2P models to be used for supervised learning and covering high-resource languages. Next, we define the language similarity and language families. Finally, we explain how to ensemble nearest languages models to predict G2P for an unseen language.

### 5.3.1 Monolingual Model

In this section, we introduce our monolingual G2P models: a joint n-gram model based on WFSTs, two neural models based on sequence-to-sequence LSTMs, and transformer models. We select those models as they are the three baseline models used in the SIGMORPHON Multilingual G2P task (Gorman et al., 2020). These models are trained for every training language and then used as building blocks to approximate G2P models for unseen testing languages.

The joint n-gram model is a standard monolingual G2P model (Novak et al., 2016). For each training language, the dataset is first aligned using Expectation Maximization, then an n-gram

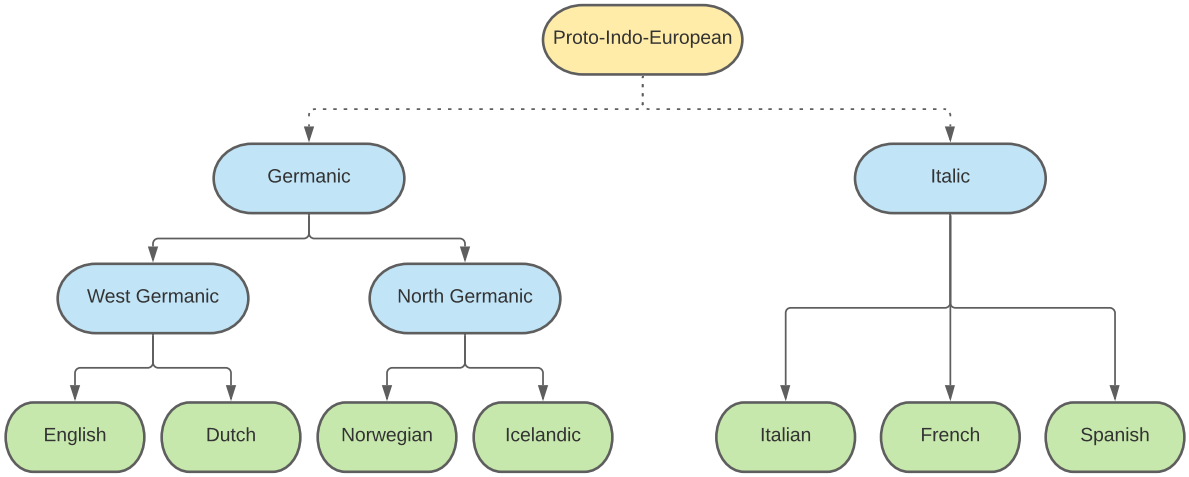



Figure 5.1: Illustraction of a partial phylogenetic tree (i.e. language family tree). The subtree has Proto-Indo-European as the root of the family (there also exists many other root language families). The Germanic branch and Italic branch can be derived (not directly though) from the Proto-Indo-European, they are further divided into the modern languages we are using today. This information can help us compute the similarity between languages.

model is built using a WFST[2]. The neural model is a standard sequence-to-sequence model. We tried two common architectures: bidirectional LSTM and transformer. Unlike the n-gram model, the neural model is trained by combining all training sets into one large dataset. To distinguish different languages, a ISO 639-3 language ID is attached to the input sequence, for example, we attach the "<eng>" to "hello", so the input sequence is "<eng> h e l l o". This approach was explored in previous work (Peters et al., 2017). It allows the parameters to be shared across different languages. Even language with a limited training set could benefit from other high-resource languages.

### 5.3.2 Phylogenetic Tree and Nearest Languages

The model discussed in the previous subsection could predict phonemes for any training language, however, it cannot deal with any unseen languages. Our main contribution in this work is to select the highly related languages and then effectively combine those models to approximate the target language. In this subsection, we introduce the concept of the nearest language in terms of the phylogenetic tree (i.e. language family tree), then we explain how we ensemble nearest languages.

There are many metrics to measure the distance between languages from different perspectives

[2]https://github.com/AdolfVonKleist/Phonetisaurus

(Dryer and Haspelmath, 2013; Littell et al., 2017). In this work, we only consider the phylogenetic tree (i.e., language family tree) to measure the distance between languages. This is because the phylogenetic information is available for a larger portion of languages than any of the other bases of linguistic distance or similarity. Glottolog provides us with language family information for around 8000 languages (Nordhoff and Hammarström, 2011).

In Figure 7.1, we write a subtree of the entire phylogenetic tree, in particular, it illustrates two major branches of the linguistic *Stammbaum*: the Germanic and Italic. Both of them are children of the Proto-Indo-European (PIE) node. The tree also indicates that English and Dutch are closely related languages and that Norwegian and Icelandic are closely related languages. To measure the distance between any pair of languages, we can compute the length of the shortest path between the two languages. In our example, the English/Dutch pair has a distance 2, and the English/Norwegian pair has a distance of 4. The shortest path can be computed efficiently by using Lowest Common Ancestor (LCA).

$$d(l_1, l_2) = H(l_1) + H(l_2) - H(LCA(l_1, l_2)) \tag{5.1}$$

where $d(l_1, l_2)$ is the distance between language $l_1$ and $l_2$, $H$ compute the height of a node in the tree. This time complexity is $O(log(M))$ where $M$ is the max height of the phylogenetic tree (Cormen et al., 2009). Suppose the entire language set is $L$ and training languages are $T \subset L$, we could compute the $k$ nearest languages for every language $l \in L$, those languages would allow us to ensemble models.

Note that the original tree structure in Glottolog groups languages into separate top-level families, therefore languages belonging to different top-level families do not have any direct path among them. To connect all languages, we add a root node and set all top-level languages as its direct children. There are also several assumptions in our approach that might not be correct: for example, we assume languages belonging to the same family should share similar orthography, however, this is not always the case. They are also influenced by non-linguistic aspects such as political factors and cultural factors. Additionally, we assume each language is only using one script, but some languages are actually written in multiple scripts. For example, Uzbek is written with a Perso-Arabic, Cyrillic, and Latin script. Despite all those limitations, information on language families provides a reasonable starting point.

### 5.3.3 Model Ensemble

After obtaining the nearest languages and the monolingual model for each of the training languages, we can use those models to approximate the target model. In particular, we are interested in combining prediction outputs from different models to create a single prediction output. If the models are one of the local prediction models (i.e: for each grapheme, we decide whether

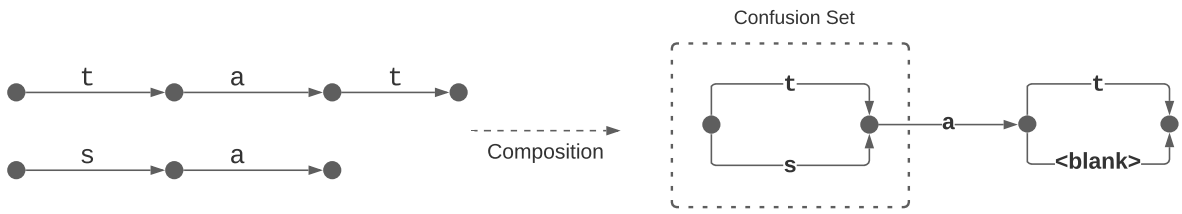



Figure 5.2: An illustration of an actual ensemble example from our dataset. The input is 'that' from Old Dutch (odt), its top-2 nearest language in our training set are Dutch (nld) and Middle Dutch (dum). The left-hand side denotes two hypotheses generated from those two languages, from which we compose into a confusion network. The composed confusion network has three confusion sets, which would vote '/t a t/' as a final prediction.

to generate a phoneme and which phoneme to generate) (Sejnowski and Rosenberg, 1987; Black et al., 1998), the ensemble task is simple. As we made one phoneme prediction at every grapheme position, we can use the voting to decide the most likely phoneme.

$$[\hat{p}] = \text{argmax}_{[p]} \sum_{i} \mathbf{1}([p] = [p]_i) \tag{5.2}$$

However, for the more general sequence-to-sequence neural model, it is more complicated. Different models would predict outputs with variable sequences, therefore voting at each position would be meaningless. For example, suppose two phoneme sequences "/helo/" and "/elo/" are generated from "hello" using two different languages. It is difficult to average /h/ and /e/ as they are corresponding to different graphemes. To solve this problem, we use a robust approach to ensemble outputs with variable lengths. Our approach is similar to the ROVER system (Fiscus, 1997), which is a commonly used approach to combine multiple speech outputs into one output. It has been applied to combine phoneme sequence (Schlippe et al., 2014), but only under the monolingual scenario in which they combine different models to improve the performance. This work focus on combining multilingual outputs and modifying the standard word-based network to consider the phonological structure.

One actual example from our dataset is illustrated in Figure 5.2. First, we build one *confusion network* (or *lattice*) per language in our nearest language set. The raw confusion network represents a single hypothesis using a directed graph whose edge corresponds to a single phoneme from the hypothesis[3]. When we compose multiple confusion networks into one confusion network, there

[3]We can also generate n-best hypotheses from each model and build confusion networks, however, we only consider the top-1 hypothesis in this work for simplicity. N-best hypotheses might be a future work

would typically be more than one edge connecting two nodes. The set of edges connecting two contiguous nodes is typically referred to as the *confusion set* (or *correspondence set*) (Fiscus, 1997; Mangu et al., 2000). For example, the first confusion set from the right network in Figure 5.2 is $\{/t/, /s/\}$. The goal of our ensemble approach is to compose all confusion networks into a single network, and then pick up the best hypothesis from the composed network.

Unlike the original work in which hypotheses are composed without any specific order, we iteratively compose the network using the nearest order: we first compose the nearest and second nearest confusion network into a single network, then further merge the third nearest network into it. In each composition step, we align two networks by computing the similarity between pairs of confusion sets. While the standard network computes the similarity step using the exact matching metric, we relax this exact matching scheme and use a more coarse matching strategy by considering the phonological distance structure. In particular, we use the *phonologically-equivalent class*, which collapses similar sounds into a small number of classes (Mortensen et al., 2016b). This means we could easier match $/a/, /o/$ (vowel pairs) than $/a/, /s/$ (vowel, consonant pairs). After composing all confusion networks into one network, the most likely phoneme sequence can be generated from the final network. To generate the sequence, we pick up 1 phoneme per confusion set and concatenate them together. The phoneme in each confusion set is selected using the voting scheme. When there are multiple candidates with equal votes, we break the tie by selecting the candidate generated from the nearest language. Algorithm 1 summarizes the entire steps in our approach.

## 5.4 Experiments

In this section, we show the experiment results on our G2P models. First, we introduce the main datasets we used to build our model, next we describe our baseline models and G2P architectures we use in our experiments. Finally, we demonstrate that the proposed ensemble approach outperforms those baseline models in different architectures.

### 5.4.1 Data

The main training/testing dataset we used is the Wiktionary website. Wiktionary is a large multilingual website containing lexicon information for many languages, including many low-resource languages. One previous work has prepared a dataset using Wiktionary (Deri and Knight, 2016), but the testing languages and training languages are mixed together in this dataset: many testing languages are also available as training languages. To demonstrate our approach on unseen languages, we create a new dataset using the latest Wiktionary. First, we download a dump file from

**Algorithm 2:** G2P algorithm

**Data:** $input, lang$ (Grapheme sequence and its language)
**Result:** $output$ (ensembled phoneme sequence)
$klangs \leftarrow KNearestLanguage(lang)$
$hyps \leftarrow []$
**for** $klang \in klangs$ **do**
  $hyp \leftarrow G2P(input, klang)$ ; `/* Generate hypothesis for every nearest language */`
  $hyps.append(hyp)$
**end**
$x \leftarrow ConfusionNetwork()$
**for** $hyp \in hyps$ **do**
  $n \leftarrow ConfusionNetwork(hyp)$
  $a \leftarrow align(x, n)$
  $x \leftarrow composite(x, n, a)$
**end**
$output \leftarrow []$
**for** $cs \in x$ **do**
  $p \leftarrow vote(cs)$ ; `/* vote 1 phoneme per confusion set */`
  $output.append(p)$
**end**

the website and extract all words with pronunciation information[4]. We group all words by their languages, which gives us 972 languages in total. However, not all languages yield a similar number of training data. Figure 5.3 shows the log-scaled histogram of language counts for different vocabulary sizes. Only 1 language: English, has more than 400k vocabulary items. Most of the languages are concentrated in the lowest histogram bar. In our dataset, we find that the majority of the language have less than 100 vocabulary items. Therefore, the model needs to be able to handle low-resource training scenarios.

Next, most languages from Wiktionary can be assigned an ISO 639-3 ID, which can be identified in our phylogenetic tree. As mentioned in the previous section, our phylogenetic tree is built using the Glottolog database (Nordhoff and Hammarström, 2011), which contains phylogenetic information about 7915 languages. We split all languages into training languages or testing languages depending on the vocabulary size: we consider the language to be a training language if the vocabulary size is above a predefined threshold, otherwise, it is classified as a testing language. Typically, there is a trade-off when selecting the threshold: making the threshold lower would increase the number of training languages and make it easier to find the nearest languages, however

[4] `https://github.com/tatuylonen/wiktextract`

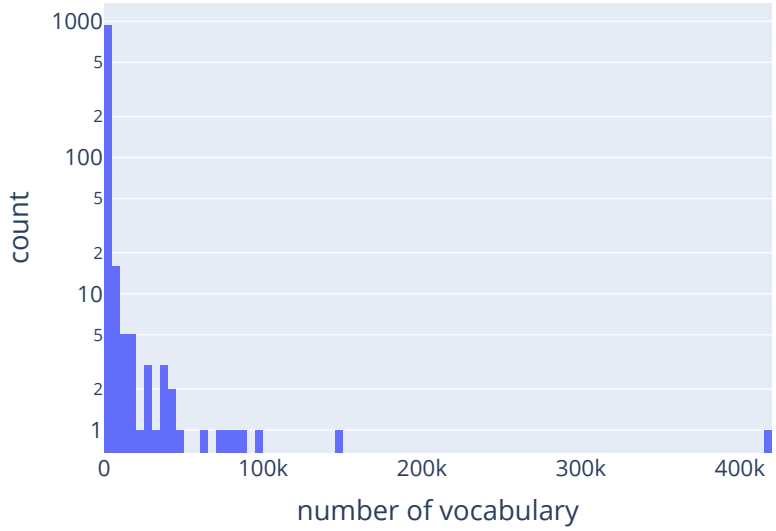



Figure 5.3: Log-scaled histograms of the count of languages grouped by the vocabulary size available in Wiktionary. The language with over 400k vocabulary is English, however, most languages are low-resource languages for which we have less that 100 Wiktionary entries.

lower threshold make the training process more difficult due to the number of limited vocabulary, additionally, it would reduce the number of testing languages. In our experiment, the threshold is set to 50 by following the previous work (Deri and Knight, 2016), and the statistics of both training datasets and test datasets are shown in Table 5.2. We have 269 training languages and 605 testing languages. Most of the training languages have a large vocabulary size but the testing languages have only 8 vocabulary items per language on average. The number of distinct graphemes is 9082 and the number of phonemes is 416. The grapheme number is much larger than the phoneme one because many languages are using non-Latin scripts, for example, there are around 4000 distinct Chinese characters in our grapheme set. We train both the n-gram model and neural models using only the training languages, and then test them on the testing languages, which are not seen during the training process. The evaluation is done using the average PER (phoneme error rate) across all testing languages.

### 5.4.2 Baselines

In our experiments, we consider three different baseline models: the **fixed language model**, which is a model trained using the English dataset. The **global language model** is a shared model mixing all training sets, it ignores the target language id during inference, this was explored in the previous

| Dataset | # Languages | # Vocabulary |
|---|---|---|
| Training set | 269 | 1,672,444 |
| Testing set | 605 | 4,796 |
| All | 874 | 1,677,240 |

Table 5.2: Statistics of the Wiktionary dataset we used in the experiment. 269 languages are used for training and 605 languages are used for testing.

| | N-gram Model | | | | LSTM Model | | | | Transformer Model | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PER | Add | Del | Sub | PER | Add | Del | Sub | PER | Add | Del | Sub |
| Fixed Model | 76.0 | 4.52 | 9.39 | 62.1 | 78.1 | 4.53 | 20.4 | 53.2 | 78.5 | 3.2 | 19.0 | 56.2 |
| Global Model | 70.4 | 6.89 | 9.86 | 53.6 | 72.8 | 3.4 | 29.0 | 43.4 | 74.2 | 2.9 | 20.6 | 50.8 |
| Nearest Model | 68.4 | 4.51 | 12.4 | 51.5 | 43.8 | 12.1 | 4.0 | 27.6 | 45.4 | 15.8 | 3.6 | 26.1 |
| Ensemble Model | **55.0** | 0.56 | 23.6 | 30.9 | **35.7** | 10.0 | 3.4 | 22.2 | **39.8** | 13.9 | 3.1 | 22.8 |

Table 5.3: Experiment Results of the our approach. It compares our ensemble model with three baselines: Fixed Model, Global Model and Nearest Model. The comparison is performed under three different architectures: N-gram model, LSTM model, Transformer Model. In all settings, the proposed model outperforms baselines.

work (Peters et al., 2017). The **nearest language model** can be seen as a special case of our proposed model: we compute the most similar language to the target language and run inference using that language's model instead. For each of the baseline models, we investigate three different architectures: N-gram, LSTM, and transformer architecture. We use OpenNMT-py[5] for our neural models. The LSTM architecture is using the framework's default configuration: 2 standard LSTM layers for both encoder and an attention-based decoder, each layer has 500 hidden size. This model is optimized with 1.0 learning rate using SGD optimizer. The transformer model uses the framework's WMT sample configuration[6]: we have 6 layers for both the encoder and decoder with 500 attention and feedforward size. The mode has a positional encoding layer and is using 8 heads in self-attention. The optimizer is Adam with learning rate 2.0 and 8000 steps for warmup. Both neural models are trained with 20k steps. In our ensemble model, we use the top-10 languages $(k = 10)$ in our main experiment.

[5] `https://github.com/OpenNMT/OpenNMT-py`

[6] `https://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model`


Figure 5.4: The effect of using different number of nearest languages when ensembling models. It shows that we reach the best performance when we use the top-10 languages to ensemble outputs.

### 5.4.3 Results

Table 5.3 shows our experiment results. For each of the G2P architecture (N-gram Model, LSTM Model, Transformer Model), we demonstrate our ensemble model's results as well as 3 baselines. The leftmost architecture shows the N-gram Model result: the fixed language model performs 76% PER, The global language model get 70%, which is better than the fixed language model. the nearest language model further improves it to 68%. While all those models perform poorly, the reason for their poor performance is different from each other: the fixed language model is only trained with the English dataset, therefore it cannot handle orthography rules in other languages. The global language model suffers from the inconsistency of the training set: the same grapheme might map to different phonemes in different languages, therefore it cannot learn consistent rules across all languages. Recall the grapheme "h" have different pronunciations in English and Spanish. Finally, the nearest language model has the problem that the nearest language might be a low-resource language. As we mention in the previous section, most languages have few training vocabularies, even we restrict the training languages to have more than 50 vocabularies, the large proportion of languages still have 50 to 100 vocabularies, which might be insufficient to train a good model. Additionally, depending on a single language might have a large variance. The proposed ensemble model solves those issues to some extent: it relies on more than 1 language when predicting for the target language: even 1 language is a low-resource language, other

languages might be able to compensate for that low-resource language. Additionally, introducing more language also reduces the variance. The proposed model significantly improves the PER to 55.0%.

Table 5.3 also demonstrates the performance of two neural models: the LSTM model and the transformer model. Interestingly, the neural model's performance does not perform better than the n-gram model when using a fixed language, even slightly worse than it. It is because the neural model further overfits the English dataset and could not capture orthography rules in other languages. The global model has the same trend, which again fails to fit each language. However, the nearest language model significantly reduces the error rate by almost 30%. Unlike the N-gram architecture, whose models of different languages are trained using a separate dataset, the neural model uses the shared architecture, and only distinguishes different languages by a language tag. This allows efficient parameter sharing between low-resource languages. Ensembling the model further reduces the error rate by more than 5%. In our experiment, the LSTM model and the transformer model have similar trends in their performance, but the LSTM model has a better performance than the transformer's one. The reason might be that there are far more hyperparameters to be tuned in the transformer model and the default sample configuration provided by the framework might not be optimal. As the main contribution of this work is to propose a general approach to ensemble languages rather than exploring different neural architectures, we only focus on how to ensemble models of different languages in this work.

### 5.4.4 Ensemble Analysis

It would be interesting to compare the number of languages when ensembling languages. Figure 5.4 demonstrates the influence of the number of languages from the LSTM model. PER drops quickly when we start ensembling models, it reaches the bottom when the number of nearest languages is 10, then starts to increase very slowly. We observe that there exists a bias-variance trade-off when changing the number of languages. When the number is relatively small, the prediction relies heavily on each language, therefore causing high variance when predicting for the target language. Increasing the number of languages could alleviate the variance problem, but using a large number of languages would decrease the accuracy as the selected languages are no longer close to the target language, which introduces more bias to the model.

To further understand the behavior of the model, we also show curves of Addition, Deletion, and Substitution in Figure 5.4. It indicates that after we start ensembling the model (from 2), the addition is increasing while the deletion is decreasing in general, the substitution decreases first and remains relatively flat later. The opposite trend of addition and deletion can be explained by the ensembling approach: when we introduce a new hypothesis into the model, it is probable some phonemes might not be aligned to the existing confusion set in the confusion network, to

| Errors | Most Common Errors |
|---|---|
| Add | /a/, /k/, /u/, /i/, /n/, /o/ |
| Del | /a/, /i/, //, /e/, /j/, /u/ |
| Sub | (/a/, /o/), (/o/, /u/), (/r/, /l/), (/t/, /d/) |
| Add | /a/, /i/, /k/, /u/, /s/, /o/, |
| Del | /a/,//, /i/, /e/, /u/ , /j/ |
| Sub | (/r/, /l/),(/a:/, /a/), (/i:/, /i/), (//, /e/) |

Table 5.4: Most frequent errors in the LSTM model. The top half shows the errors in the nearest model, the bottom-half shows the errors when using 10 languages

incorporate these new phonemes into the network, we need to create new confusion set, which would lead to more phoneme emissions. More phonemes would also contribute to decreasing the deletion rate as well. Therefore, that curve of PER is very similar to the curve of the substitution error (as the addition and deletion almost cancel each other). Not only does the ensemble model improve the substitution error quantitatively, it also improves the errors qualitatively: Table 5.4 shows the most frequent errors made by the nearest language model and the top-10 ensemble model. It indicates the most frequent substitution errors (/a/, /o/) and (/o/, /u/) are replaced by (/a/, /a:/) and (/i/, /i:/). We find latter errors are much closer to each other (they have phonological distances of 1, while the former errors have larger distances), therefore they are much better errors than the first two pairs qualitatively.

## 5.5 Limitations

While we get reasonable performance in our testing languages, we acknowledge that there are several limitations in our approach: first, both of our training languages and testing languages are limited to languages available in Wiktionary. The full Glottolog Phylogenetic Tree has 110 top-level branches in total, however, our dataset only spans 40 branches. Therefore if we want to apply our approach to unseen languages in the remaining 70 branches, we have to depend on unrelated languages to build our ensemble model, which might lead to worse performance. Second, as our approach heavily depends on Glottolog and Wiktionary, if the language is not available in the Glottolog database or the vocabulary quality in Wiktionary is not good enough, then our approach cannot be applied to it. Finally, many of the 8k languages do not have orthographies, therefore it might be difficult or meaningless to evaluate the G2P performance for them.

## 5.6 Conclusion

In this chapter, we propose a zero-shot learning method to approximate G2P models for 8k languages in the world. We use the phylogenetic tree to measure the distance between languages and combine multilingual outputs. We test our approach on 600 unseen languages and demonstrate it significantly outperforms baselines.

# Chapter 6

# Language Model: Speech Recognition for 2000 Languages


## Summary

Most recent speech recognition models rely on large supervised datasets, which are unavailable for many low-resource languages. In this chapter, we conclude the entire speech pipeline proposed in this thesis. We present a speech recognition pipeline that does not require any audio for the target language. The only assumption is that we have access to raw text datasets or a set of n-gram statistics. Our speech pipeline consists of three components: acoustic, pronunciation, and language models. Unlike the standard pipeline, our acoustic and pronunciation models use multilingual models without any supervision.

The acoustic model is the model proposed in Part I, especially in Chapter 4, the pronunciation model is the model introduced in the previous Chapter 5. In this chapter, we propose the language model, which is built using n-gram statistics or the raw text dataset. We build speech recognition for 1909 languages by combining it with Crúbadán: a large endangered languages n-gram database. Furthermore, we test our approach on 129 languages across two datasets: Common Voice and CMU Wilderness dataset. We achieve 50% CER and 74% WER on the Wilderness dataset with Crúbadán statistics only and improve them to 45% CER and 69% WER when using 10000 raw text utterances.



ASR2K: Speech Recognition for Around 2000 Languages without Audio Interspeech 2022

## 6.1 Introduction

Recently, the performance of speech recognition has witnessed rapid improvement due to modern architectures (Gulati et al., 2020; Karita et al., 2019; Watanabe et al., 2018). Those models typically require thousands of hours of training data for the target language. However, there are around 8000 languages in the world (Lewis, 2016), the majority of which do not have any audio or text datasets. There have been some attempts to reduce the size of the training set by using pretrained features from self-supervised learning models (Baevski et al., 2020; Hsu et al., 2021). However, such models still rely on a small amount of paired supervised data for word recognition. More recently, inspired by the recent success of unsupervised machine translation (Conneau et al., 2017; Artetxe et al., 2018), there is some work applying the unsupervised approach to speech recognition as well (Baevski et al., 2021). Those models apply adversarial learning to automatically learn a mapping between audio representations and phoneme units. They can learn a phoneme recognition model using an unlabeled audio dataset and a text dataset.

Despite the success of those recent approaches, all of these models rely on some audio datasets of the target language (labeled or unlabeled), which significantly restricts the scope of target languages. In this work, we investigate whether we can develop speech recognition systems without requiring any audio dataset or pronunciation lexicon for the target language. The only assumption is the existence of some monolingual text or a set of n-gram statistics for the target language. Our proposed method consists of three components: acoustic, pronunciation, and language models. Both acoustic and pronunciation models can be trained using supervised datasets from high-resource languages, and then applied to the target language by taking advantage of some linguistic knowledge. Both models can be applied in a zero-shot learning fashion without any supervision. Finally, we use the raw texts or n-gram statistics to create a language model, which is then combined with the pronunciation model to create a WFST decoder. To analyze our pipeline more efficiently with small test sets, we also propose an approach to decompose the observed errors into acoustic/pronunciation model errors and language model errors.

We apply our approach to 1909 languages using Crúbadán: a large endangered languages n-gram database and then test our approach on 129 languages from the Common Voice (34 languages) and CMU Wilderness dataset (95 languages) (Ardila et al., 2019; Black, 2019). On the Wilderness dataset, we achieve 50% CER (character error rate) and 74% WER (word error rate) respectively when using Crúbadán's statistics only, and improve them to 45% CER and 69% WER by using 10000 raw text utterances. As far as we know, this is the first attempt to build speech recognition for thousands of languages without audio.

## 6.2 Related Work

Most speech recognition approaches can be classified into one of several groups depending on their data requirements. The most common group has access to the paired supervised dataset $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{N}$ where $(X, Y)$ is a paired audio and text of an utterance. If the size $N$ of the dataset is large enough, various end-to-end models can be trained using CTC, ASG, seq2seq, RNN Transducer, and other objectives (Graves et al., 2006; Collobert et al., 2016; Graves et al., 2013; Sutskever et al., 2014). If the size is small, then it would be a low-resource speech recognition in which some acoustic knowledge should be transferred from high-resource languages (Li et al., 2019a; Xu et al., 2020). Self-supervised training takes advantage of another large raw speech dataset $\{X_j\}$ to learn hidden representations of speech signals, those representations are useful to the supervised tasks and can reduce the amount of the paired dataset (Baevski et al., 2020; Hsu et al., 2021). The semi-supervised learning approach also leverages unlabeled speech datasets or text datasets to augment the supervision set (Veselý et al., 2017; Synnaeve et al., 2019; Rosenberg et al., 2019).

Recently, unsupervised speech recognition attempts to target the dataset $\mathcal{D} = (\{X_i\}_{i=1}^{I}, \{Y_j\}_{j=1}^{J})$ where we have access to an unlabeled raw audio set $\{X_i\}_{i=1}^{I}$ and a raw text dataset $\{Y_j\}_{j=1}^{J}$ (Baevski et al., 2021). The audio and text do not need to be aligned with each other. A generator model is jointly trained with a discriminator model. The generator model attempts to translate audio into phonemes, while the discriminator model attempts to distinguish between phonemes transliterated from text and phonemes recognized from the generator. The disadvantage of this direction is that the model could only recognize phonemes instead of words and it requires a phonemizer (pronunciation model) for the target language, which would not be available for most languages. Another related direction is unsupervised speech unit discovery (Chorowski et al., 2019; Tjandra et al., 2019), which is similar to the self-supervised learning approach and attempts to discover phone units from audios $\mathcal{D} = \{X_i\}_{i=1}^{I}$. This group of approaches, however, cannot emit explicit phonemes or words as it does not have knowledge of the lexicon and language model for the target language.

In this work, we propose a new paradigm to focus on the text-only dataset $\mathcal{D} = \{Y_j\}_{j=1}^{J}$. While all the previous groups require some amount of audio datasets $\{X_i\}$ (paired or unpaired) for the word recognition of the target language, we argue this requirement can be relaxed to some extent.

## 6.3 Model

Our speech pipeline is divided into the *acoustic model*, *pronunciation model* and *language model*. The joint probability over speech audio $X$ and speech text $Y$ can be factorized as

$$p_\theta(X, Y) = \sum_P p_{\text{am}}(X|P)p_{\text{pm}}(P|Y)p_{\text{lm}}(Y) \tag{6.1}$$

, where $P$ is the phoneme sequence corresponding to the text $Y$. The pronunciation model $p_{\text{pm}}$ is typically modeled as a deterministic function $\delta_{\text{pm}}$. In our pipeline, only the language model can be estimated from the text, both the acoustic model and pronunciation model are approximated using zero-shot learning or transfer learning from other high resource languages, therefore we denote $\hat{p}_{\text{am}}, \hat{\delta}_{\text{pm}}$ for the approximated acoustic model and pronunciation model. The previous factorization can be approximated by

$$p_\theta(X, Y) \approx \hat{p}_{\text{am}}(X|\hat{P})p_{\text{lm}}(Y) \tag{6.2}$$

, where $\hat{P} = \hat{\delta}_{\text{pm}}(Y)$ is the approximated phonemes.

### 6.3.1 Acoustic Model

We briefly review the acoustic model introduced in the Part I. The acoustic model should be able to recognize phonemes of the target languages even when the languages are unseen in the training set. We follow a direction of recently proposed allophone-based multilingual architectures (Li et al., 2020a, 2021a). This direction attempts to recognize phonemes of an unseen language using language-independent phone representations and their mappings to the language-dependent phonemes. Essentially, those architectures attempt to represent the acoustic model as follows:

$$\hat{p}_{\text{am}}(P|X) = \sum_Q p_{\text{lang}}(P|Q)p_{\text{uni}}(Q|X) \tag{6.3}$$

, where $p_{\text{uni}}(Q|X)$ is a language-independent universal phone recognition model, recognizing physical-level phone units $Q$ from the speech audio $X$. The allophone architecture $p_{\text{lang}}(P|Q)$ is to encode how each physical phone should be mapped to a language-dependent phoneme. The relation between phones and phonemes is called an *allophone*, which is usually encoded as a 1-$n$ deterministic function annotated by phonologists for each language. The mapping is easier to obtain than the supervised dataset for low resource languages. We rely on Allovera and PHOIBLE datasets for allophone mapping of more than 2000 languages (Mortensen et al., 2020; Moran and McCloy, 2019). The other model $p_{\text{uni}}(Q|X)$ does not have any dependency on the target language, therefore it can be trained using high-resource languages such as English and Mandarin. The CTC objective is used to train this acoustic model (Graves et al., 2006). The conditional independence assumption in CTC prevents the model from biasing too much towards one specific language model (e.g: English), therefore it can be easier to apply to other low-resource languages. In our

experiment, we observe the originally proposed model (Li et al., 2021a), is not very robust when recognizing audios from different domains. To further improve the model, instead of using the standard filterbank features, we use self-supervised learning (SSL) features as our frontend feature extraction (Baevski et al., 2020; Hsu et al., 2021; Conneau et al., 2020).

### 6.3.2 Pronunciation Model

We also shortly revisit the pronunciation model covered in Chapter 5. The pronunciation model is essentially a G2P (grapheme-to-phoneme) model that can predict the phoneme pronunciation given a grapheme sequence: $P = \delta_{\text{pm}}(Y)$. For high-resource languages, the G2P model can be either trained using a dictionary or be developed using rule-based systems (CMU, 2000; Mortensen et al., 2018). However, the majority of the languages do not have any accessible dictionaries or rules, therefore we consider an approximated pronunciation model $\hat{\delta}_{\text{pm}}$ instead. In particular, we apply a recently proposed multilingual G2P model as our pronunciation model (Li et al., 2022b). For any target language $l_{\text{target}}$, this G2P model selects top-$k$ nearest languages: $l_{\text{topk}} \in \text{KNN}(l_{\text{target}})$ whose training set is available, then during the inference, it first propose $k$ hypothesis using each nearest language model $\delta_{l_{\text{topk}}}$, the models are ensembled by combining hypothesis into a lattice to emit the most-likely approximated sequence:

$$\hat{\delta}_{l_{\text{target}}} = \text{Ensemble}(\{\delta_{l_{\text{topk}}} | l_{\text{topk}} \in \text{KNN}(l_{\text{target}})\}) \tag{6.4}$$

The similarity metric between languages is defined to be the shortest path of two languages on the phylogenetic tree (i.e: language family tree). This approach enables us to approximate the pronunciation model for every language in Glottolog database (Nordhoff and Hammarström, 2011), which contains phylogenetic information about 7915 languages.

### 6.3.3 Language Model

For the language model, we first estimate the vocabulary $V = \{w_1, w_2, ..., w_{|V|}\}$ from the raw text dataset $\{Y_i\}$. For each word $w_i \in V$, its pronunciation can be approximated using the pronunciation model and then this lexicon information can be encoded into a lexicon graph $L$. The text dataset also enables us to estimate the classical n-gram language model by counting n-grams statistics $C(w_1, ..., w_n)$. This n-gram language model can be then encoded into a grammar graph $G$. Composing the lexicon graph $L$ and the grammar graph $G$ as well as the CTC topology graph $H$ would generate a WFST-based language decoder HLG (Miao et al., 2015).

We realize that the text dataset requirement $\{Y_i\}$ can be further relaxed as the building blocks of the HLG graph only consist of the statistics $\{V, C\}$ estimated from the text dataset. For languages whose text dataset $\{Y_i\}$ is absent but $\{V, C\}$ is available, we can still proceed to build the decoder

HLG. This is common for many languages in the internet: while only a few hundred languages are recognized as being in use for web texts on the World Wide Web (Lewis, 2016), there exists several large databases collecting lexicon-related statistics for thousands of languages. For example, Crúbadán is a database consisting of vocabulary, bigrams, and character-trigrams statistics for around 2000 languages (Scannell, 2007). Employing statistics from it, we build speech recognition systems for around 1909 languages.

We further push the boundaries by eliminating the dependency on $C$. Our endeavor focuses on building a language model utilizing only the vocabulary $V$, given that lexicon information is more readily available compared to raw text information. Expanding the range of languages, we leverage the Panlex database (Kamholz et al., 2014)—a lexicon-only resource that provides information for 5981 languages. Adopting a naive uniform distribution over the vocabulary for each language, we construct a unigram language model. Combined with the languages already present in the Crúbadán database, we effectively cover 6185 unique languages from the 7915 listed in the Glottolog database.

### 6.3.4 Error Decomposition

Since the acoustic, pronunciation models are approximated models, it is helpful to understand how the approximation would impact our results. As the final observed errors also contain the language model errors, we propose a framework to decompose the observed errors $\epsilon_{\text{observed}}$ into language model errors and other errors. To achieve this, in addition to the experiment using the approximated models, we conduct a new set of experiments using the *oracle* acoustic and pronunciation models (i.e. the acoustic and pronunciation model that achieves perfect performance), such that any recognition errors in this new experiment should be attributed to the language model $\epsilon_{\text{lm}}$. The gap between the observed error $\epsilon_{\text{observed}}$ and the oracle error $\epsilon_{\text{lm}}$ should correspond to the errors made by the approximated acoustic and pronunciation model $\epsilon_{\text{am/pm}}$. In other words, the observed errors can be decomposed as follows:

$$\epsilon_{\text{observed}} = \epsilon_{\text{am/pm}} + \epsilon_{\text{lm}} \tag{6.5}$$

To estimate the oracle error $\epsilon_{\text{lm}}$, every testing utterance is first converted to the phoneme sequence using our pronunciation model, the phoneme sequence is then augmented with the CTC blank labels by inserting blank labels "⟨blk⟩" between every pair of phonemes. (e.g: "a b" is converted to "⟨blk⟩ a ⟨blk⟩ b ⟨blk⟩"). Next, the augmented sequence is converted to CTC logits by giving an extremely high probability to each phoneme (including blank) for every timestep. Finally, the logits is fed into the decoder HLG to be decoded. We obtain the oracle error $\epsilon_{\text{lm}}$ by comparing it against the expected word sequence. The achieved error rate is the oracle error rate, as we assume the pronunciation model is perfect: pronunciation in logits is perfectly consistent with

the pronunciation in the HLG decoder. The acoustic model is perfect as well: it assigns extremely high probability to the "correct" phoneme.

## 6.4 Experiments

For the acoustic model, we tried 4 different models, one from the previous literature and the newly proposed SSL-based models (Li et al., 2021a). All the models are trained using cmn, deu, eng, fra, ita, rus, tur, vie languages from the Common Voice dataset (Ardila et al., 2019). In the SSL-based model, we tried three different self-supervised learning features: HuBERT, wav2vec2, and XLSR (Baevski et al., 2020; Hsu et al., 2021; Conneau et al., 2020). All the features are extracted using s3prl framework (wen Yang et al., 2021). For every SSL model, the features from the last hidden layer were used. Two layers of transformers are appended on top of the pretrained features, which are then connected with the multilingual architecture $p_{\text{lang}}(P|Q)$ as proposed in the original literature (Li et al., 2021a). The transformer layer has a 768 hidden size and 4 multi-attention heads. Other parameters follow the original literature (Li et al., 2021a). For the pronunciation model, we use the multilingual model proposed in the previous literature and its implementation (Li et al., 2022b) [1]. For the language model, we first download the complete dataset from Crúbadán's website (Scannell, 2007), which results in 1909 languages after cleaning. Each language consists of several files: unigrams, bigrams, web urls for the target language, and character trigrams. The most relevant files are unigrams (vocabulary) and bigrams. We provide statistics in Table 6.1. The same set of information can also be extracted from raw text sets $\{Y_i\}$ if we have access to them. For the WFST decoder, we use the k2 library and adapt its icefall recipe[2]. We build trigram models from texts and bigram models from Crúbadán. During the decoding, we set the search beam size to be 20, output beam size to be 8, min and max active states to be 30 and 10000. For the lexicon-only experiment, we build a unigram language model only using the vocabulary $\{V\}$.

Table 6.1: Descriptive statistics for distinct unigrams and bigram for 1909 languages from Crúbadán database.

| | mean | std | 25% | median | 75% |
|---|---|---|---|---|---|
| unigram | 10870 | 14012 | 837 | 5149 | 14761 |
| bigram | 29383 | 22087 | 2504 | 42996 | 50000 |

To test our approach on unseen languages, we use the 34 languages from Common Voice dataset (denoted by CV) and 95 languages from CMU Wilderness corpus (denoted by WN) (Black,

[1] https://github.com/xinjli/transphone
[2] https://github.com/k2-fsa/icefall

2019). For the Common Voice languages, we select the subset of languages whose text size is larger than 1000. Any languages seen in the acoustic model are excluded (i.e: cmn, deu, eng, fra, ita, rus,tur, vie). 95 languages from Wilderness are selected based on the top-100 MCD score, which measures the alignment qualities. 5 languages are excluded due to duplications and preprocess failure (i.e: gag, xsb, nah, may, pxm).

### 6.4.1 Results

We first evaluate the acoustic model using PER (phoneme error rate). Note that our PER is only an approximation of the actual PER as the expected phoneme sequence relies on the pronunciation model, which is only an approximation. However, it reveals many useful insights into the acoustic models. Table 6.2 shows the performance across 4 models. The baseline acoustic model has around 50% PER and half of the errors are deletion errors (e.g: /a/, /i/ are our most deleted phonemes). We find the main causes of deletion errors are the domain mismatch and language mismatch. To improve the robustness, we employ the SSL-based models, which decreases the error rate by 5%. Most of the improvement is from the deletion reduction. We find the XLSR model, which is a multilingually pretrained model, performs the best and we use it as the main model in the pipeline.

Table 6.2: Average results (%) of the acoustic model on all test languages. PER is the phoneme error rate, Ins, Del, Sub are Insertion, Deletion and Substitution Error. CV and WN denote Common Voice and Wilderness datasets.

| Acoustic Model | PER | | Ins | Del | Sub |
|---|---|---|---|---|---|
| | CV | WN | | | |
| Baseline (Li et al., 2021a) | 51.7 | 49.2 | 1.02 | 30.2 | 19.7 |
| SSL (HuBERT) (Hsu et al., 2021) | 49.7 | 44.3 | 1.15 | 23.8 | 20.8 |
| SSL (wav2vec2) (Baevski et al., 2020) | 49.8 | 43.4 | 1.37 | 25.8 | 18.1 |
| SSL (XLSR) (Conneau et al., 2020) | **47.8** | **42.1** | 1.49 | 24.7 | 19.2 |

Table 6.3: Average Performance (%) of the lexicon-based language model on all testing languages under different resource conditions. CER, WER denotes character error rate and word error rate.

| Language Model | CER | | WER | |
|---|---|---|---|---|
| | CV | WN | CV | WN |
| Random Word Model | 102.1 | 114.4 | 100.0 | 100.0 |
| Most Frequent Character Model | 86.8 | 81.2 | 99.4 | 99.4 |
| Panlex Model | 54.5 | 52.1 | 98.3 | 79.0 |

Table 6.4: Average Performance (%) of the n-gram based language model on all testing languages under different resource conditions. CER, WER denotes character error rate and word error rate.

| Language Model | CER | | WER | |
|---|---|---|---|---|
| | CV | WN | CV | WN |
| Crúbadán Model | 65.5 | 50.2 | 92.4 | 74.5 |
| Text Model (1k utterances) | 55.3 | 50.8 | 84.6 | 76.9 |
| Text Model (5k utterances) | 51.3 | 47.0 | 80.2 | 72.2 |
| Text Model (10k utterances) | **50.9** | **44.9** | **79.0** | **69.2** |
| Text Model (10k utterances, 2023) | **39.8** | **37.2** | **69.7** | **59.1** |

Table 6.5: A Welsh example from the Common Voice dataset. The top two rows are the hypothesis (HYP) and reference (REF) phonemes, the bottom two rows are the hypothesis and reference words. Deleted phonemes and words are highlighted.

| Model | Sentence |
|---|---|
| HYP | kɔpθχi:ðɛrpənvənkəsfɔnəkənharəχ |
| REF | kɔpe:aθjɔi:jχi:ðɛrpənvi:nɛkɛsfo:nənkənharaχ |
| HYP | gobeithio ch dderbyn yn gyson cynharach |
| REF | gobeithio i chi dderbyn fy neges ffôn yn gynharach |

Table 6.3 presents the performance of the Panlex-based language model and several lexicon-based baselines (with XLSR as the acoustic model). Initially, we examine the random word model, where words are randomly chosen from the lexicon to match the reference text's length. As anticipated, this baseline model fails to produce any meaningful predictions, resulting in a CER and WER that are approximately 100% or higher. Subsequently, we employ the repetition of the most frequent character from each language as the prediction outcome, achieving approximately 15% correct character predictions, albeit without word recognition capability. In contrast, the employment of lexicon information from Panlex in the model substantially reduces the CER to nearly 50%, demonstrating the utility of incorporating a lexicon. Nonetheless, the lexicon-based model still struggles with word recognition, owing to the absence of frequency information.

Table 6.4 shows the language model performance (using XLSR as the acoustic model). First, we try n-gram statistics from the Crúbadán without using any text dataset. It shows that Crúbadán captures some character-level information even without any text dataset: it achieves 65% and 50% CER on two datasets. The Crúbadán WER of the Wilderness languages is also very promising under this condition: 74.5%. Next, we use 1k, 5k, 10k text utterances from the training set to train the model without Crúbadán. As the training text datasets are in the same domain as the

test dataset, this improves the performance significantly. With 10k text dataset, we achieve 51% and 45% CER respectively. While we omit the result in this table, we also investigate the effect of combining Crúbadán and text language models together. However, it does not improve the performance because there is a domain mismatch between two models. The text-only language model shown in Table.6.4 performs the best.

To understand the language model errors, we compute the insertion, deletion, and substitution errors. We find the dominant errors are deletion and substitution. By comparing the most common word errors and phoneme errors, we observe that the phoneme errors have been propagated into the word errors: the previous deletion of phonemes /a/, /i/ caused deletions of the entire words, especially of some short words (e.g: *na*, *ni*). The substitution error also suffers from the missing phonemes issue. For example, the most substituted pair is (*charirca*, *carica*), it is clear that our model failed to recognize several consonants. Table 6.5 shows a typical example from the pipeline. The acoustic model tends to recognize fewer phones from the audio, those phoneme deletions propagate to the language model and lead to the word deletions.

### 6.4.2 Error Decomposition Analysis

Next, we apply the error decomposition framework to our results in Figure 6.1. The figure shows the trend of how the CER/WER responds to the size of the training text dataset. Each blue circle point on the top region represents an observed error $\epsilon_{\text{observed}}$ from the Common Voice corpus and each orange square point on the bottom region is an oracle error $\epsilon_{\text{lm}}$ with our framework. It shows that both errors tend to decrease as the size of the text dataset increases, however, the oracle error has a much sharper decreasing slop than the observed one. As we mentioned in the previous section, the oracle error shows the errors from the language model and the gap between the two errors is the error from the acoustic and pronunciation model. Based on this assumption, the figure indicates that $30 \sim 40\%$ word errors are from the language model and $40 \sim 50\%$ word errors are from the acoustic model and pronunciation model; most of the character errors are caused by the acoustic model and pronunciation model.

### 6.4.3 Language Analysis

We can also interpret the results from the linguistic perspective and discuss several limitations of the pipeline. First, we find the phonology of the target language has a crucial impact on the PER performance. Since our acoustic model is trained using high-resource languages (most of them, Indo-European) and then applied to the target language, phonemes that are not common in Indo-European languages should be difficult to recognize. For example, non-pulmonic consonants are common in some languages (e.g: implosive consonants are widespread in Sub-Saharan Africa)

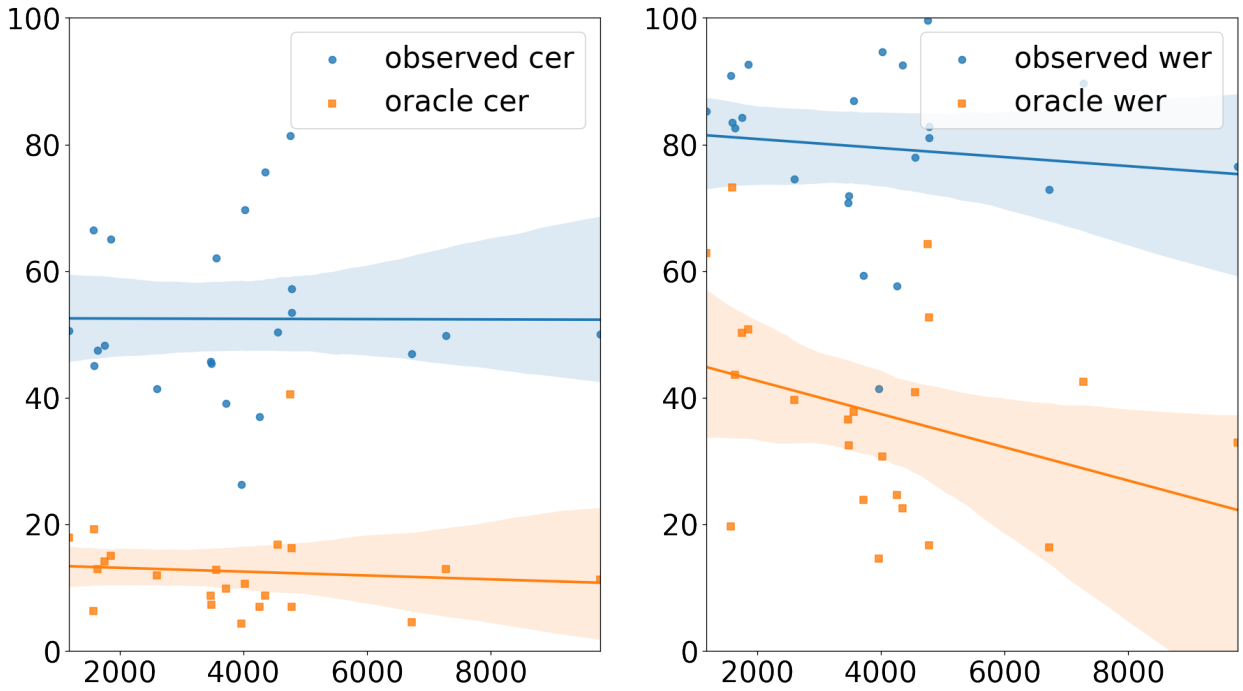



Figure 6.1: The trend of CER (left) and WER (right) using different sizes of training text. The horizontal axis represents the size of the text dataset. The vertical axis is the error. Each blue circle point denotes an observed error $\epsilon_{\text{observed}}$ from a particular language in the Common Voice corpus and each orange square point shows the oracle error $\epsilon_{\text{lm}}$. An OLS estimator is applied to all sets of points.

but are not typical phonemes in high-resource languages. Another example is the tonal language, we find the Sochiapam Chinantec language displays bad performance: 73% PER, 75.9% CER, and 96.5% WER. This language is a tonal language with 7 different tones. The acoustic model is trained without tonal information and fails to distinguish tonal contrasts (Mandarin Chinese, a tonal language, is included in the acoustic training set, but the tonal information was not used during the training). Orthography depth is another important factor for acoustic performance. The pronunciation model tends to fail more frequently when the language has a deeper orthography (i.e. the rules to map graphemes to phonemes are complicated). For instance, the Swedish language has deep orthography, which makes the PER (67%) significantly worse than the average PER. Furthermore, if the writing system of the target language is unknown to the pronunciation model, then the model cannot infer its pronunciation. In our dataset, the Maldivian language is written in the Thaana script, which is mostly unknown to the pronunciation model. The error rates are 80% PER, 81% CER and 99% WER. Finally, we observe that some languages have relatively small gaps between CER and WER, and others have larger gaps. For example, the Tai Dam language has an

error rate gap of less than 20%. On the other hand, in the closely related Northern Thai language, we observe a gap of around 40%. We find the length of a typical word is the main cause: the average token length in Tai Dam is 3.48 characters (e.g., *choi*), but Northern Thai has an average token length of 7.04 (e.g.: *we-machi-warbogwad-e-nandi*). We find there is a strong correlation between the gap and the length of word ($r = 0.8015$, in our experiment).

## 6.5 Conclusion

In this chapter, we propose a speech recognition pipeline using raw text or n-gram statistics, and we apply it to around 2000 languages. Our training scripts will be released for more researchers to explore this direction.[3]

[3]our code will be available at `https://github.com/xinjli/asr2k`

# Part III

# Datasets and Applications

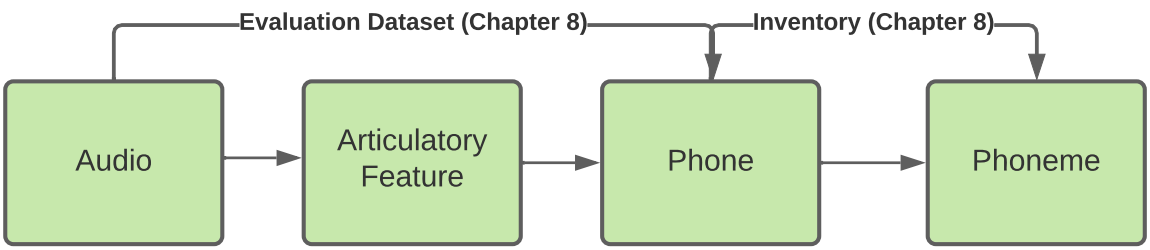



Figure 6.2: The datasets covered in Part III

# Chapter 7

# Phoneme Inventory: Phoneme Inventory Estimation for Every Language

## Summary

In Part I, we discuss the multilingual acoustic model that can be applied to unseen languages. Unlike the traditional acoustic model, it can be applied to languages without any supervised data. The only requirement for the target language is its *phoneme inventory*. However, even the largest collection of phone inventory only covers about 2000 languages, which is only 1/4 of the total number of languages in the world. A majority of the remaining languages are endangered. To extend the phoneme recognition to all 8000 languages, we need to solve the task of *phoneme inventory estimation*, which is to identify the phoneme inventory for unseen languages. It is also a crucial component in language documentation and the preservation of endangered languages.

In this chapter, we attempt to tackle this problem by estimating the phone inventory for any language listed in Glottolog, which contains phylogenetic information regarding 8000 languages. In particular, we propose one probabilistic model and one non-probabilistic model, both using phylogenetic trees ("language family trees") to measure the distance between languages. We show that our best model outperforms baseline models by 6.5 F1. Furthermore, we demonstrate that, with the proposed inventories, the phone recognition model can be customized for every language in the set, which improved the PER (phone error rate) in phone recognition by 25%.

Xinjian Li, Florian Metze, David R. Mortensen, Alan W. Black, Shinji Watanabe.
LREC 2022 Phone Inventories and Recognition for Every Language

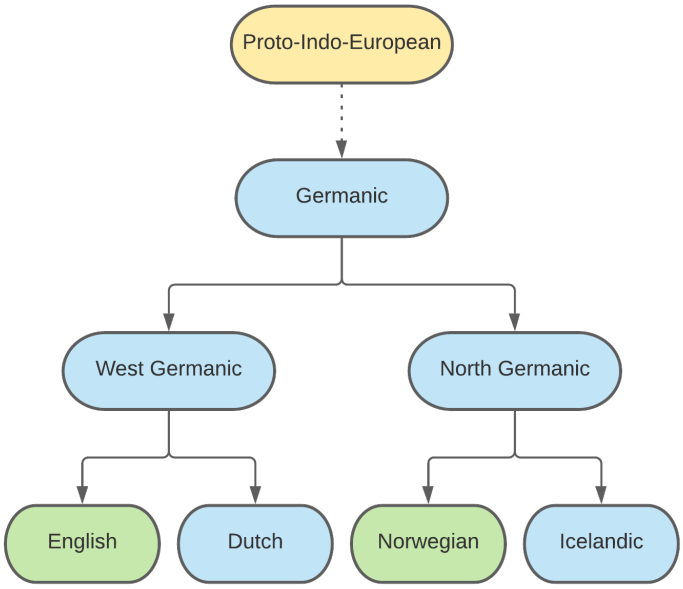



Figure 7.1: Illustration of a subtree sample from the Germanic branch a phylogenetic tree. We derive the testing inventory for Dutch and Icelandic using the training inventory from English and Norwegian

## 7.1 Introduction

A fundamental aspect of the description or documentation of any language is establishing its phone inventory (Bird and Simons, 2003; Michaud et al., 2018). This is a necessary prerequisite to further phonetic and phonological analysis (including transcribing text, discovering allophonic patterns, and developing an orthography), these are foundations upon which other facets of linguistic description can be built. Traditionally, phone inventories have been discovered by field linguists using a mixture of audio, visual, and lexical tools to arrive at a set of sounds sufficient to characterize the phonetics of the language. The largest collection of phone inventory aggregated so far is the PHOIBLE dataset (Moran and McCloy, 2019), which is a collection of phone inventories from over 2000 languages. However, there are around 8000 languages in the world, for most of which no documented phone inventory exists. Unfortunately, those languages are typically endangered (Nettle et al., 2000). Language preservation projects typically target languages in this category. Field linguists starting work on a new language will benefit from knowing, in approximate terms, what the phone inventory of that language is like.

In this work, we attempt to solve this problem by estimating the phone inventory for any language listed in Glottolog (Nordhoff and Hammarström, 2011), which contains around 8000 lan-

guages. In particular, we take advantage of the phylogenetic trees from Glottolog (as this information is available for almost every language in the world)[1]. We propose two approaches that exploit this tree structure: First, we impose a probabilistic structure on the phylogenetic tree ("language family tree"), where each child node is expected to have a similar phone distribution to its parent. Next, we introduce *nearest language ensemble* approach, in which we compute the nearest neighbor languages for any unseen target language and we ensemble the phone inventory from those nearest languages as the inventory for the target language. Note there are other features such as geographical coordinates to derive closeness between languages. These features, however, are not easy to model for non-leaf nodes in our approach (e.g: it might not make sense to assign a specific coordinate to the Indo-European family node). As a result, we only consider the simple tree structure in this work. We apply our approach to 77 languages, whose inventories are excluded from our training set. This experiment shows that our approach achieves an F1 score of 65.9, which is 6.5 points better than the best baseline model. Finally, we demonstrate that, using the proposed phone inventories, we enable a recently proposed phone recognizer to recognize all 8000 languages (Li et al., 2021a)(Chapter 4). Our results show that with the hypothesized phone inventories, we achieve 64.2% PER (phone error rate), which is 25% better than the original model (Li et al., 2020a) (Chapter 3). To the best of our knowledge, this is the first speech recognition system that has been successfully customized for almost every known language known to comparative linguistics.

## 7.2 Related Work

Compiling the phonemic/phonetic inventory for a single target language is typically an important task in phonetic and phonological analysis (Hayes, 2011). However, not all languages in the world are equally well-researched. For example, much phonological research has focused on richly resourced languages (therefore they usually have well-defined phone inventories (International Phonetic Association et al., 1999)), while other, low-resource languages have historically received less attention. Recently, there have been several unsupervised models proposed that are meant to discover linguistic units for unwritten languages (Varadarajan et al., 2008; Müller et al., 2017b; Dunbar et al., 2019, 2020), those models typically require the raw speech recordings for discovery, whose resources are limited for most languages (Black, 2019).

While most traditional phonetic research has been focused on a single language or a few languages, there have been several attempts to compile large databases to collect many phone inventories of a diversity of languages. PHOIBLE (PHOnetics Information Base and Lexicon) is a

[1]Since linguists do not always agree upon the phylogenetic groupings of languages—especially of poorly-studied languages—the trees from Glottolog are necessarily imperfect. However, they usually represent state-of-the-art classifications and are thus useful for our experiments here.

phonological inventory database which contains inventory information of more than 2000 distinct languages (Moran and McCloy, 2019), each phone also has been assigned distinctive phonological features (Jakobson et al., 1951; Chomsky and Halle, 1968). Another large database compiled by Merritt Ruhlen is the Ruhlen Database (Creanza et al., 2015). It contains not only the phonological information for each language, but also a wealth of extra-linguistic information (e.g: number of speakers and the geographical location of each language). While both projects have successfully collected many sound inventories, the majority of the inventories of the world's languages remain undocumented. To address this problem, this work attempts to give a reasonable approximation of each phone inventory for every language registered in Glottolog.

## 7.3 Approach

In this section, we introduce our two proposed approaches: Bayesian Network Estimation and Nearest Languages Ensemble. Before that, though, we propose two baselines and setup notations used in this work.

### 7.3.1 Baseline

Assume a set of training languages is $L$. For every training language $l \in L$, we have access to its phone inventory $\Sigma_l$. The simplest inventory estimation model uses the inventory $\hat{\Sigma}_{\text{fixed}}$ from a fixed language, for example Tagalog: $\Sigma_{\text{<tgl>}}$. This is because Tagalog's inventory has a reputation for typicality. Note that not every well-known language can be a good baseline. The English inventory $\Sigma_{\text{<eng>}}$, for example, is atypical: it includes some very rare phones like [θ] and [ð] but lacks (depending on analysis and dialect) some very common phones like [a], [e], and [o]. This *Fixed Inventory*, however, only covers phones from a single language; therefore it fails to include common phones in other languages and has low recall. Another possible baseline would be to the use the entire phone inventory available from all training languages:

$$\Sigma = \bigcup_{l \in L} \Sigma_l \tag{7.1}$$

This is a default inventory used in some phone recognition works (Li et al., 2021a, 2020a). This naïve approach should improve recall but it includes far more phones that any individual language and most of them are, invariably, false positives. To improve the precision, we sort all phones by the number of times they appear in our training languages and only keep the top-$n$ most frequent phones based on the following statistics. This inventory baseline is the *Global Inventory* $\hat{\Sigma}_{\text{global}}$

$$\sum_{l \in L} \mathbb{1}([p] \in \Sigma_l) \tag{7.2}$$

### 7.3.2 Bayesian Network Estimation

The global inventory reflects the overall trend of phones across all languages, but it does not capture the local similarity between languages. We propose to exploit a phylogenetic tree to capture the local relations between languages (based on the insight that languages that are phylogenetically close also have similar phone inventories). Our first model is to impose a probabilistic structure to the tree. In particular, we consider the tree to be a *Bayesian Network* (i.e: a directed probabilistic graphical model). For each node in the tree, a multinomial distribution over the entire inventory $\Sigma$ is assigned. We assume that the inventory of the child node is drawn from its parent's multinomial distribution. Formally, suppose we have a parent node $r$ and its child $l$ where the child $l$ is one of our training languages. We can model the probability of drawing the child inventory using $r$'s multinomial distribution:

$$\text{Prob}(\Sigma_l|\Theta_r) = \frac{|\Sigma_l|!}{\prod_i (x_i!)} \prod_{i \in \Sigma_l} \theta_i^{x_i} \tag{7.3}$$

where $\Theta_r = \{\theta_1^r, ..., \theta_\Sigma^r\}$ is the parameter of parent node $r$, and each parameter $\theta_i^r$ is the probability to draw the $i$-th phone from all available phones $\Sigma$, and $x_i$ is the indicator function whether the $i$-th phone is contained in the child $l$'s inventory. The parameter $\Theta_r$ can be inferred using *Maxmimum Likelihood Estimation* (MLE). After obtaining the parameter $\Theta_r$ of the parent node $r$, we could construct the phone inventory $\hat{\Sigma}_{\text{bayes}}$ for the parent node by selecting phones with the top-$n$ highest probability. This is equivalent to select the top-$n$ phones which have the highest counts in children of $r$.

$$\sum_{l \in \text{Children(r)}} \mathbb{1}([p] \in \Sigma_l) \tag{7.4}$$

### 7.3.3 Nearest Language Ensemble

The Bayesian Network model can infer parent's inventory using its children information, however, it cannot take advantage of information from other close nodes (e.g: sibling nodes). To fix this issue, our second model is to use the nearest languages to approximate the inventory of the target language. The metric to define distance between languages is the length of the shortest path between any two languages in the phylogenetic tree. The shortest path between any two language nodes can be efficiently computed with *Lowest Common Ancester* (LCA) whose time complexity

is $O(log(H))$ where $H$ is the height of the phylogenetic tree (Cormen et al., 2009). For a target language, suppose we find the top-$k$ nearest languages $L_k$, then we first count the appearance of each phone $[p]$:

$$\sum_{l \in L_k} \mathbb{1}([p] \in \Sigma_l) \tag{7.5}$$

Then we could select the top-$n$ phones $\hat{\Sigma}_{\text{nearest}}$ using these counts. For example in Figure 7.1, suppose that our training languages are English and Norwegian, and we would like to estimate the inventory for Dutch, when we use the top-1 nearest language, only English would be selected and we could simply copy the English inventory to the Dutch inventory, when we use $k = 2$, we would identify English and Norwegian as the nearest languages, and average them using counts.

## 7.4 Universal Phone Recognition

The hypothesized phone inventories pave the way for many new applications. Notably, they allow us to create phone recognition systems for (almost) every language in the world. In this section, we first introduce the acoustic model we use in this work, then we explain how to apply the estimated inventory for the recognition task.

### 7.4.1 Architecture

We closely follow the architecture described in the previous work (Li et al., 2021a): The architecture has a hierarchical structure which is illustrated in Figure 9.1. We model three different units explicitly: *phonemes*, *phones* and *phonological attribute*. Phonemes are typically language-dependent units, whereas phones are language-independent units. Phonological attributes or articulatory attributes are a set of discrete properties to characterize each phone. The set of phones corresponding to one phoneme in a particular language is called the *allophones* of the phoneme, which is annotated by phonologists. We use an annotated dataset to map between phone and phonemes (Mortensen et al., 2020). Similarly, each phone can also be decomposed into a set of attributes. The correspondence is also well-studied by linguists and we use tools to extract attributes for each phone (Mortensen et al., 2016b). During the training process, the encoder would first encode each frame of the audio into a hidden vector, from which we could obtain the distributions of phones in each frame using their attributes. Each phone distribution is further transformed into phoneme distribution and optimized by the CTC loss function. In this work, the encoder is a 12-layer transformer-based encoder whose hidden size is 640 and multi-head attention size is 4. The feature is the 40-dimension filter bank. We train the model using eng, cmn, deu, fra, ita, rus, tur,

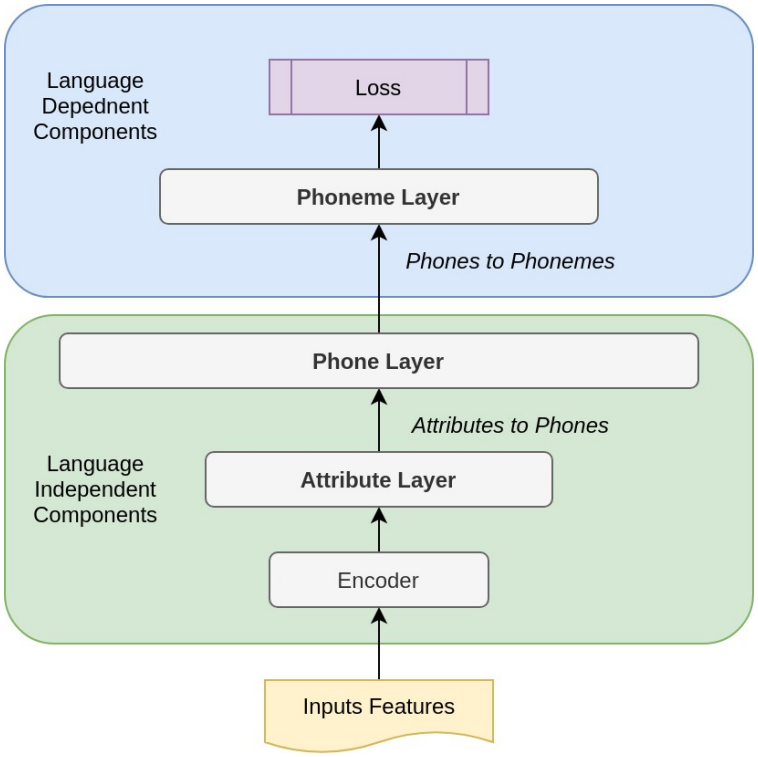



Figure 7.2: The architecture of the phone recognition model. We first compose the phone representations using their phonological attributes. Then we compute the phone distributions using the hidden vector from the encoder, Next, the language-independent phones are transformed into language-dependent phonemes with the allophone mappings, which are finally optimized by the loss (CTC) function

vie languages from Common Voice corpus (Ardila et al., 2019). Our trained model is available online. [2]

### 7.4.2 Inference

As the lower part in Figure 9.1 is language-independent, we can apply the trained model to any unseen languages whose inventory is accessible: if both phoneme and phone inventory are available, we can plug those inventories into the model and run the inference. If only the phone inventory is available, we approximate the phoneme set with its phone set, assuming each phone is mapped to the same phoneme.

Even the phone inventory, however, is not always available for every language. For languages whose phone inventory is absent, an approximated phone inventory should be used instead. In pre-

[2]Interspeech21 model at `https://github.com/xinjli/allosaurus`

vious works, the inventory was chosen to be the global inventory $\hat{\Sigma}_{\text{global}}$: all the available training phones to make their prediction (Li et al., 2020a, 2021a). This naïve approach, however, has the low precision problem because the set of all training phones is too large. We demonstrate that, employing the hypothesized inventories $\hat{\Sigma}_{\text{bayes}}, \hat{\Sigma}_{\text{nearest}}$ introduced in the previous section, we can improve the phone recognition accuracy.

## 7.5 Experiments

In this section, we demonstrate our experimental results for both phone inventory evaluation and phone recognition. As mentioned in the previous section, we first build the phylogenetic tree using Glottolog (Nordhoff and Hammarström, 2011). The tree contains 7915 languages, where there are 43 top-level language families. We further create a root language node which possesses all top-level languages as its children, therefore all the languages are connected and can be reached from a single root node. Most leaf nodes can be identified with ISO 693-3 language ID while most non-leaf nodes have Glottolog IDs attached to them. Next, we use the PHOIBLE as our training phone inventory. PHOIBLE contains 2100 languages, and 2091 of them can be mapped to one of the leaf node in the Glottolog-based tree.

For every unseen node in the tree (leaf or non-leaf), we estimate its phone inventory using our proposed models. For each model, we specify the size of inventory to be $n = 40$, which is a typical size of the phone inventories in our training set. To evaluate the model, we select 77 languages as the unseen testing languages and take them out of our training languages. The languages are selected from a recently proposed multilingual phone dataset (Li et al., 2021c), in which we can identify 77 out of 95 languages in our tree. For every testing language, we evaluate both their inventory coverage (using the F1 score) and the phone recognition accuracy (using phone error rate) as an extrinsic task. The ISO 693-3 id of testing languages are abk, ace, ady, afn, afr, aka, asm, azb, bam, bem, ben, bfd, bfq, bin, brv, bsq, cbv, ces, cha, cpn, dag, dan, deg, dyo, efi, ell, ema, eus, ewe, ffm, fin, fub, gaa, gla, guj, hak, hau, haw, heb, hil, hin, hrv, hun, hye, ibb, ibo, idu, ilo, isl, kan, kea, khm, klu, knn, kri, kub, kye, lad, led, lgq, lit, lkt, lug, mak, mal, mlt, mya, nan, njm, nld, ozm, pam, pes, run, tzm, wuu, yue.

### 7.5.1 Phone Inventory Evaluation

Table 7.1 shows the statistics for the four models: the fixed language inventory has 51.1 F1 with 48.8 precision and 57.5 recall. As mentioned in the previous section, the Tagalog inventory contains many cross-linguistically common phones, which makes the recall much higher than the precision. We found it interesting to investigate which commonly used languages perform better

| Model | F1 | Prec | Rec |
|---|---|---|---|
| Fixed Inventory ($\hat{\Sigma}_{\text{fixed}}$) | 51.1 | 48.7 | 57.5 |
| Global Inventory ($\hat{\Sigma}_{\text{global}}$) | 59.4 | 58.1 | 64.8 |
| Bayesian Network ($\hat{\Sigma}_{\text{bayes}}$) | 61.2 | 60.0 | 66.7 |
| Nearest Neighbor ($\hat{\Sigma}_{\text{nearest}}$) | **65.9** | 71.3 | 64.7 |

Table 7.1: F1, precision and recall for 77 testing languages and each model. The two models on top are the baseline models and the two on the bottom are the proposed models.

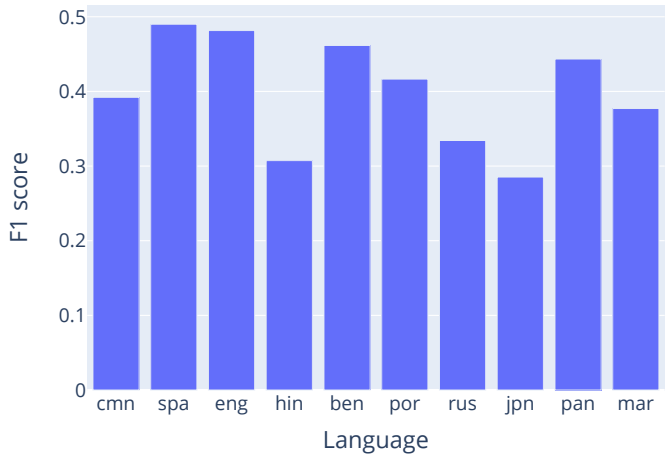



Figure 7.3: Comparison of inventory evaluation using different fixed language. Spanish has the highest F1 score among the top-10 languages ranked by the population.

in this regard. We evaluate the top ten languages, ranked by the population of first language speakers (Lewis, 2009). Figure 7.3 indicates that Romance branch from the Indo-European language family tends to have relatively high scores, but none of them outperforms the Tagalog inventory. While the fixed language inventory can capture 50% of the inventory, it only consists of the inventory from an individual language and fails to reflect the global properties of all languages. On the contrary, the global inventory baseline is built using statistics from all training languages, which improves the F1 score by 8 points. Our experiment shows that selecting the most frequent phones is essential for the global baseline. We also consider another global inventory baseline which consists of all basic phones available in the IPA table (without diacritics and modifiers). This model only achieves a 27.2 F1 score: it captures most phones in every language (high recall), but it generates many false positives, which significantly decreases the precision.

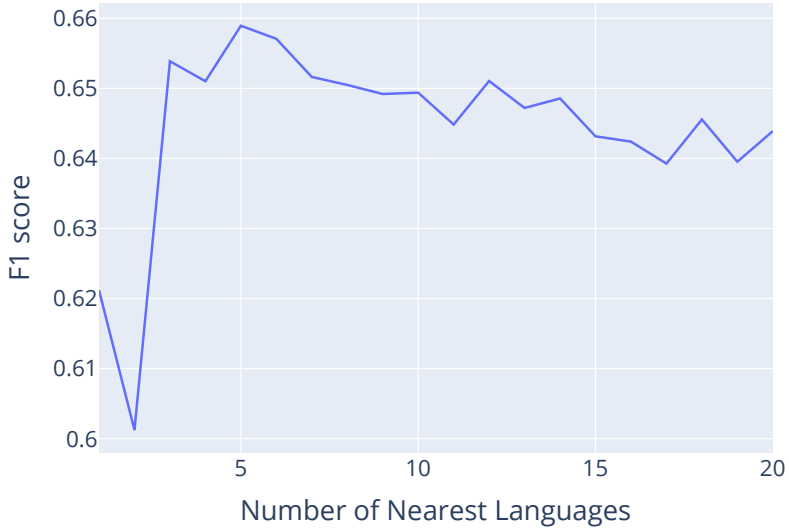



Figure 7.4: Comparison of performance when using different number of nearest neighbors

To further incorporate information from local language branches, we propose the Bayesian Network model and Nearest Neighbor model. Table 7.1 shows that they further improve the F1 score by 1.8 and 6.5 points respectively. Despite the simplicity of the nearest neighbor model, it outperforms the Bayesian Network model by 4.7 point. This is because the nearest neighbor model can capture more languages than the Bayesian Network Model. Suppose we would like to estimate the inventory of West Germanic (as in Figure 7.1). The Bayesian model will only rely on the training languages among its children: English alone. On the other hand, the nearest neighbor model can take advantage of training languages in other branches: Norwegian. This gives the nearest neighbor model more information when deciding the inventory, which significantly improve its precision from 60.0 to 71.3. Next, we investigate the effect of using different number of nearest neighbors. Figure 7.4 is a line plot showing the result of using different number of nearest neighbors ($k$). We observe a bias-variance trend in our experiment. When $k = 1$, we simply search for the nearest language and use that language to approximate the target language. This suffers from large variance as it only uses one language's inventory. Increasing $k$ reduces the variance by averaging over $k$ nearest languages. However, increasing $k$ too much also hurts the performance as the additional languages are far from the target language and introduce bias into the inventory instead.

### 7.5.2 Universal Phone Recognition

Finally, we report the results of the extrinsic task in Table 7.2. The original phone recognition models propose to use the union of all available phones when the inventory is not available. This approach, again, suffers from the low-precision problem and only achieves 89.2% PER. In contrast, all 4 models introduced in this work (including the two baselines) improve the PER by more than 20%. the nearest neighbor model again achieves the highest performance of 64.2%. The gap between 4 models, however, is smaller than the inventory evaluation. This is because phones are not uniformly distributed in utterances (Li et al., 2021b), and frequent phones typically have already been captured by the global inventory (as we select them based on the sorted order). The major F1 improvement of Bayesian Network and Nearest Neighbor approach comes from the identification of other rare phones, therefore the improvement is reduced in this task. Despite the small gap between the 4 proposed models, we show that using a proper inventory could significantly improve the PER.

| Model | PER | Add | Del | Sub |
|---|---|---|---|---|
| Default Inventory ($\Sigma$) | 89.2 | 3.8 | 16.2 | 69.1 |
| Fixed Inventory ($\hat{\Sigma}_{\text{fixed}}$) | 67.4 | 3.6 | 15.4 | 48.2 |
| Global Inventory ($\hat{\Sigma}_{\text{global}}$) | 65.3 | 3.4 | 15.2 | 46.7 |
| Bayesian Network ($\hat{\Sigma}_{\text{bayes}}$) | 64.6 | 2.5 | 20.1 | 41.8 |
| Nearest Neighbor ($\hat{\Sigma}_{\text{nearest}}$) | **64.2** | 3.2 | 16.4 | 44.6 |

Table 7.2: Statistics of the universal phone recognition task. Lower PER (phone error rate) indicates better performance. Add, Del, Sub are Addition, Deletion and Substitution errors

## 7.6 Limitations

While we get reasonable performance in our testing languages, we acknowledge that there are several limitations in our approach: first, our approach heavily depends on Glottolog, if the language is not available in the Glottolog database, then our approach cannot be applied to it. Second, if the target language does not have any training languages near it (e.g: it is the only language in its branch), then the approximation might not be accurate.

## 7.7 Conclusion

In this work, we propose multiple approaches to estimate phone inventories for unseen languages. By using the knowledge derived from phylogenetic trees, we demonstrate that they significantly improves the inventory quality over competitive baselines and boost performance in a phone recognition task. This work also paves the way for appling speech recognition technology to (almost) every language in the world. All the phone inventories of 7915 languages would be released to enable more researchers to explore them in future research.

# Chapter 8

# Dataset: Multilingual Phonetic Dataset for Low Resource Speech Recognition


## Summary

In Part I of this thesis, we discuss the multilingual acoustic model, in particular, we introduce a language-independent phone recognition model, which is able to recognize phones of low-resource languages. However, most speech recognition datasets so far only focus on high-resource languages, there are very few datasets available for low-resource languages, especially datasets with detailed phone annotation.

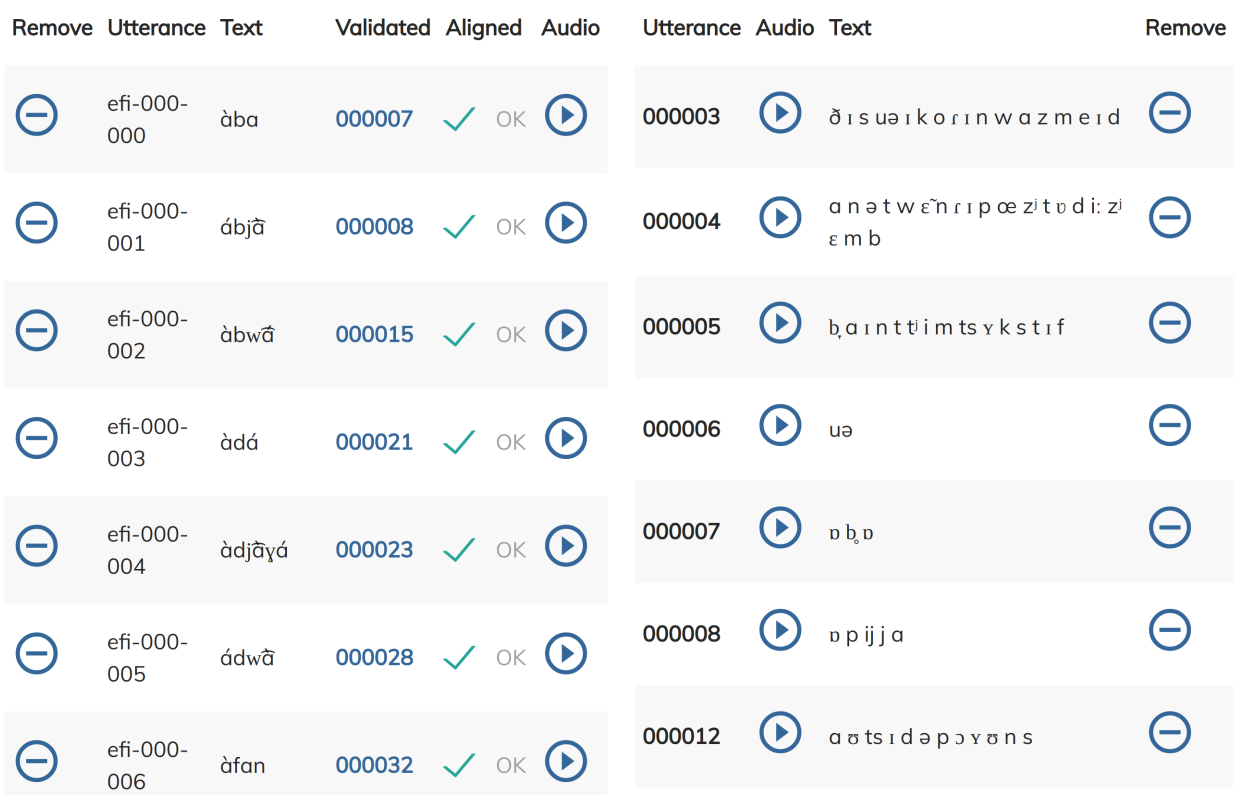
In this chapter, we present a large multilingual phonetic dataset, which is preprocessed and aligned from the UCLA phonetic dataset. The dataset contains around 100 low-resource languages and 7000 utterances in total. This dataset would provide an ideal training/evaluation set for universal phone recognition. This dataset is used as an evaluation dataset in Chapter 4.



Xinjian Li; David R. Mortensen; Florian Metze; Alan W Black. ICASSP 2021 Multilingual Phonetic Dataset for Low Resource Speech Recognition


## 8.1 Introduction

Recently, speech recognition communities have made significant progress towards building deep neural networks for speech recognition by taking advantage of huge volumes of training data and high-quality test sets (Amodei et al., 2016; Xiong et al., 2018). While high-resource languages such as English and Mandarin have been able to benefit from the newly developed technology (Godfrey et al., 1992; Cieri et al., 2004), most of the languages in the world are low-resource languages



| Remove | Utterance | Text | Validated | Aligned | Audio |
|---|---|---|---|---|---|
| ⊖ | efi-000-000 | àba | 000007 | ✓ OK | ▶ |
| ⊖ | efi-000-001 | ábjə̃ | 000008 | ✓ OK | ▶ |
| ⊖ | efi-000-002 | àbwə̃ | 000015 | ✓ OK | ▶ |
| ⊖ | efi-000-003 | àdá | 000021 | ✓ OK | ▶ |
| ⊖ | efi-000-004 | àdjə̃ɣá | 000023 | ✓ OK | ▶ |
| ⊖ | efi-000-005 | ádwə̃ | 000028 | ✓ OK | ▶ |
| ⊖ | efi-000-006 | àfan | 000032 | ✓ OK | ▶ |

| Utterance | Audio | Text | Remove |
|---|---|---|---|
| 000003 | ▶ | ð ɪ s u ə ɪ k o ɾ ɪ n w ɑ z m e ɪ d | ⊖ |
| 000004 | ▶ | ɑ n ə t w ɛ ̃n ɾ ɪ p œ zʲ t ʊ d iː zʲ ɛ m b | ⊖ |
| 000005 | ▶ | b̥ ɑ ɪ n t tʲ i m ts ʏ k s t ɪ f | ⊖ |
| 000006 | ▶ | u ə | ⊖ |
| 000007 | ▶ | ɒ b̥ ɒ | ⊖ |
| 000008 | ▶ | ɒ p ij j ɑ | ⊖ |
| 000012 | ▶ | ɑ ʊ ts ɪ d ə p ɔ ʏ ʊ n s | ⊖ |

Figure 8.1: A alignment sample from the dataset where the left Table shows the annotated phones/utterances extracted from the website, the table on the right side is the segmented audio chunks and the recognized phones. Two tables are first aligned automatically with phonetic features distances and then fixed manually.

lacking large sets of training data or even small test sets. More importantly, many languages do not have standardized orthographies; speech datasets fully annotated with phonetic transcriptions are the only means of building speech technologies for them. Unfortunately, phonetically-annotated data sets are also largely limited to high-resource languages (Garofolo et al., 1993). Additionally, the annotated data is usually monolingual corpus with a limited phone inventory. Ideally, a well-annotated dataset should contain a large number of languages and have a rich phone inventory. This would be useful not only to train the recognition system for the target language but also benefit to build/evaluate any language-independent universal phone recognizers as introduced in Part I of this thesis (Li et al., 2020c).

In this chapter, we introduce a large multilingual phonetic dataset, which is derived from the online UCLA phonetics archive (Ladefoged et al., 2009)[1]. The online phonetics archive contains

[1]http://archive.phonetics.ucla.edu/

a large amount of speech data collected by field linguists. For each language, there are typically a variety of materials available including the audio recordings in WAV format, transcribed word lists, information about the native speakers, etc. The total number of languages is around 300, and most of them are low-resource languages (most of which have less than 1 million native speakers). However, for each subset of the data, the archive only includes a large audio file and a table of transcriptions, along with some other information. No alignments are provided, which poses a challenging problem to create the aligned phonetic dataset.

In this work, we tackle this problem by using two-step alignment: in the first step, we segment the transcriptions and audio files into small utterances. All audio files are transcribed into phones with the recognizer proposed in Chapter 3 (Li et al., 2020a). Every transcribed utterance is aligned automatically with the recognized utterances by measuring the phone feature distance. Next, all aligned utterances are manually validated and corrected by human experts. Additionally, during the second step, we implement several simple but effective strategies to speed up alignment correction. The prototype dataset contains around 100 languages and 7000 utterances, it will be distributed to benefit future work[2]. Note that the number of utterances and languages might change in the final version.

## 8.2 Related Work

Previously, many multilingual datasets have been created from different sources such as audiobooks, broadcast news, and online recordings. These include the Babel database (Harper, 2011), TUNDRA corpus (Stan et al., 2013), Voxforge collections (Voxforge.org) and common voice dataset (Ardila et al., 2020). While those datasets sometimes cover more than 10 languages, the target languages are typically high-resource languages, whereas low-resource languages are rarely included. More recently, a dataset has been prepared for a much broader group of languages (Black, 2019). However, the dataset is automatically aligned and alignment quality differs among languages. Additionally, the transcription is in the orthographic form where the phonetic transcriptions are not available. In this chapter, we prepare a low-resource language dataset with fully annotated and validated phonetic transcriptions.

To develop a good alignment tool is essential for this work, as a fully manually alignment would be a poor use of valuable expert time. The automatic alignment problem, arising when curating speech corpora or synchronizing audiobooks, has been addressed in prior works (Anguera et al., 2014; Bordel et al., 2012; Malfrère et al., 2003; Black, 2019). There are typically two approaches to finding alignments between audio and transcriptions. The first is to utilize a speech recognizer to transcribe audio into text or phones, and then estimate the alignment between the outputs with

[2]https://github.com/xinjli/ucla-phonetic-corpus

the provided transcriptions (Anguera et al., 2014; Bordel et al., 2012). The second, on the other hand, obtains the audio signals by synthesizing transcriptions, then aligns the original audio with the generated audio (Malfrère et al., 2003; Black, 2019). While both groups have achieved some success in obtaining usable alignments, they typically require some prior knowledge of the target language, and the aligned pairs are usually not systematically validated. In this work, we establish the alignment for around 100 languages while assuming little prior information. Additionally, human feedback is used efficiently to validate and correct alignments.

## 8.3 Approach

In this section, we introduce the methods used to develop the dataset. We obtain the raw dataset by crawling the archive pages. We then automatically align the recognized phones and annotated phones for the utterances. Finally, experts employ an online tool to manually but efficiently validate and correct the alignments.

### 8.3.1 Preprocessing

The crawler first downloads the top page and extracts all available languages. It then recursively parses the individual links to the pages for each language and extracts all annotated word or utterance lists, together with the corresponding audio files. The utterance lists are typically contained in tables whose headers document the content type of each column. As the headers do not always follow identical naming conventions, several regular expressions are used to determine which column contains the phone annotations. From each utterance list, we typically extract 10 to 100 annotated words/utterances.

The corresponding WAV file is usually a long audio recording containing the entire contents of the utterance list. Besides, it usually contains many unrelated contents such as the introduction of the native speaker, instructions regarding what to read next, and some incidental conversation. Since most of our annotated utterances typically contain a single word in each utterance, voice activity detection is applied to segment the audio into small chunks. For each annotated utterance, one particular chunk is expected to contain its speech. It is important to note that the acoustic environment varies significantly across different languages' recordings; some are clean enough for the voice activity detection to work efficiently, but others contain so much noise and overlapping speech that voice activity detection cannot consistently distinguish silence and speaking intervals.

### 8.3.2 First Pass Alignment

Next, all audio chunks are fed into a recently proposed multilingual phone recognizer proposed in Chapter 3 (Li et al., 2020a), by which each chunk is transformed into an appropriate sequence of phones. The first-pass alignment is done by matching the golden annotated phone labels and the recognized phone labels. Typically, the phone-level alignment is done using standard string edit distance and greedy search. i.e, for each annotated utterance, we compute the edit distance with all recognized phone sequences and select the utterance with the lowest cost. However, this baseline alignment fails to produce a good first-pass alignment in this case, due to two challenges: First, the gold phone transcriptions are partial transcriptions. Many speaking parts in the recordings are not transcribed as they are not related to annotation (e.g: instructions to native speakers of what to read next). Second, the recognizer has not seen most of the languages (and, understandably, performs worse on languages it has not seen). However, by taking advantage of several properties in the dataset and the recognizer, we arrive at alignments that are much better than those produced by this baseline. Three approaches are introduced in this section.

#### Monotonic Alignment

First, the annotated utterances are not listed in random order. The relationship between the annotated word list and the associated recording is typically monotonic. While other material may intervene in the recording, the utterances of interest are in the same order in the recordings as in the annotations. We note there are several cases in which this order fails to be monotonic, for example, the native speaker occasionally forgets reading some utterances and returns to those utterances later. However, by imposing this constraint, the available matching pairs are greatly reduced. Coupled with dynamic programming, this makes alignment much more efficient.

#### Phonological Distance

Next, we use phonological features to measure the distance between annotated phones and recognized phones (instead of using the exact phone match). The phonological distance enables us to quantify similarity more precisely. In particular, we use the PanPhon tool to compute the phonological distance between two utterances where 22 phonological features are taken into account (Mortensen et al., 2016a). For example, [syllabic], [sonorant], [consonantal], etc. Instead of penalizing phone mismatch with 1 cost, it imposes a penalty based on partial feature mismatch.

#### Consecutive Segment Merger

Another improvement in the alignment can be made by merging consecutive vowels or consonants in the recognized phones. During the experiment, we found that the recognizer tends to generate

more than one vowel or consonant for a single phone when that specific phone context is rare in the training set. This issue tends to increase the distance even when the recognized phones are close to the annotation. Table 8.1 shows such an example in which merging multiple vowels and consonants could lead to a more accurate distance. We note that it is not always correct to merge vowels and consonants since sequences of multiple vowels or multiple consonants do occur in many languages; however, we find this approach helps to reduce many misalignments in practice.

| **Annotated Phones** | **Recognized Phones** | **Distance** |
|---|---|---|
| [tʰɑɪbɒ] | [m a z] | 1.45 |
| [tʰɑɪbɒ] | [t ɕe i: b̪̊ uə ə] | 3.04 |
| [tʰɑɪbɒ] | [t e b̪̊ uə] | 1.18 |

Table 8.1: An actual example from the experiment to merge consecutive vowels and consonants into one phone. The annotated phones [tʰɑɪb] should be aligned with the [t ɕe i: b̪̊ uə ə], but was originally misaligned with [m a z] as it has less distance, after merging vowels and consonants in the 3rd row, it has less distance and could be aligned correctly.

### 8.3.3 Second Pass Alignment: Real-Time Feedback

During the second phase, we use our online tool to update the first pass alignment in real time based on feedback (validation or correction) from annotators. In particular, we exploit two types of feedback to improve the alignment. Both types are fast enough to update alignments in real time.

First, we use the anchor point to improve the alignments. When a new validation is confirmed or a new alignment is fixed, the aligned utterance index and audio index are sent to the server, notifying it of the new anchor point. The remaining unverified alignments are updated, subject to this new anchor point. In the first pass alignment, the alignment errors tend to propagate through the last utterance whenever there is a large mismatch. Fixing the anchor point could bring the alignment back to the correct starting point.

Next, we use the index interval information to improve the alignment. During the experiment, we noticed that the aligned audio index has a typical index interval in each dataset. For example, the aligned audio index might be 10, 12, 14, etc: the first utterance is aligned to the 10-th audio, the second utterance is aligned to the 12-th audio. This is because the native speaker and the linguist are talking in turns: one reads the utterance, then the other instructs what to read next. Each dataset has a different pattern, but the index interval is usually consistent in each dataset. During the second pass, we use the validated utterances to estimate the typical index interval by taking the mean of validated/fixed utterances intervals. The interval is then taken into account as

| Approach | Acc. Mean | Acc. Std |
|---|---|---|
| First Pass (baseline) | 4.88% | 6.37% |
| First Pass (+ monotonic) | 27.3% | 21.6% |
| First Pass (+ distance) | 6.62% | 12.5% |
| First Pass (+ merge) | 5.64% | 8.32% |
| First Pass (+ all) | 38.0% | 26.4% |
| Second Pass | **56.0%** | 24.3% |

Table 8.2: Alignment accuracy of different approaches. The first pass on the first row is the baseline alignment, in which there are no constraints in the alignment. Additionally, we add three different First Pass approaches and measure the performance separately and jointly. The Second-Pass shows the improved alignment accuracy by using real-time feedback.

a new distance factor when updating the alignments. By combining those two types of real-time feedback, the validation and fixing process requires much less manual works.

## 8.4 Experiments

In this section, we evaluate our alignment approach and provide statistics for the collected dataset. In the first version of our dataset, we provide alignments for 106 languages. For each language's dataset, the alignment is first automatically aligned and then validated/fixed by an expert.

### 8.4.1 Alignment Evaluation Results

We first evaluate the alignment performance across all 106 languages. The metric is the *alignment accuracy*: whether each annotated utterance is correctly aligned with the target audio chunk or not. As we handle a large number of languages in the experiment, instead of showing the alignment accuracy for each language, we show the mean and standard deviation of accuracy across all languages. The results are shown in Table.8.2, in which we compare several approaches we mentioned in the last section. First, we consider the naive first pass alignment in which we greedily match each utterance with all audio candidates. The results are around 5% accuracy, which is hardly useful as the first pass alignment. Next, we try the three approaches mentioned above: imposing the monotonic order, using phonetic distance instead of the naive edit distance, merging consecutive vowels and consonants. The monotonic constraint improves the alignment significantly by about 20% accuracy. The other two only increase the metric marginally when used separately, however, when all three approaches combined, it improves the accuracy by more than 30%.

During the first pass alignment, we notice there is a huge accuracy variance across different languages as shown by the standard deviation: some datasets are aligned very successfully with almost 100% accuracy. On the other hand, some corpora fail with near 0% accuracy. The variance can be explained by several factors: first, the audio quality varies significantly across different languages: some recordings were made in a clean environment, while others were done in relatively noisy rooms. The audio quality affects the recognition accuracy and therefore makes a huge difference during the first pass. Second, the recordings are segmented by voice activity detection. Some speakers wait for 1–2 seconds between every utterance while others continue to speak several utterances without any interruption. As there is no silence between the utterances, the single audio chunk contains several utterances and could not get aligned with any of the target annotations. Finally, imposing the monotonic order might propagate the alignment error to the last utterance. While the first pass alignment could align 40% correctly, it still requires a huge amount of effort to fix the remaining 60% utterances. In the second pass, we apply the real-time feedback to the system and automatically fix many alignment errors with the new anchor point and interval information. The table suggests that alignment accuracy is further improved to around 60% in the second pass. Finally, the remaining 40% misaligned utterances are fixed manually.

### 8.4.2 Dataset Statistics

The first version of our dataset contains 106 languages with 6,880 validated utterances. Each language contains around 60 utterances on average with 28.4 std. We find that some languages have many more utterances and speakers than others.

The data related to areal distribution is shown in Table.8.3. We show the language distribution and utterance distribution across different areas. The table suggests that nearly half of the languages in the dataset are from Africa, while only 4% of languages are from the Pacific area. The utterance distribution is relatively proportional to the language distribution. However, Asian languages dominate in the utterance count with 43.6%. African languages have fewer utterances in proportion to the number of languages. We also investigated the distributions of phones in the entire dataset. In total, we find the number of unique phones (phone types) is more than 400. 51.7% of the phones are consonants and 48.3% are vowels. The detailed feature distribution is shown in Table.8.4, which suggests that the phone inventory is rich in various categories. This dataset should be useful in many ways. First, it can be used to evaluate phone recognition systems for the included low-resource languages. Additionally, it might serve as a good training/evaluation set for any universal phone recognizers due to its rich inventory and large coverage of languages.

| Language Area | Language % | Utterance % |
|---|---|---|
| Africa | 48.5% | 23.1% |
| America | 6.15% | 8.63% |
| Asia | 26.2% | 43.6% |
| Europe | 15.3% | 22.5% |
| Pacific | 3.85% | 2.17% |

Table 8.3: Area distribution of languages and utterances

| **syllabic** | **sonorant** | **continuant** | **delayed release** |
|---|---|---|---|
| 44.3% | 67.8% | 68.0% | 0.53% |
| **lateral** | **nasal** | **strident** | **spread glottis** |
| 4.02% | 10.8% | 1.74% | 1.71% |
| **cons glottis** | **anterior** | **coronal** | **distributed** |
| 2.04% | 38.2% | 30.4% | 4.67% |
| **labial** | **high** | **low** | **back** |
| 16.9% | 25.0% | 17.4% | 25.4% |
| **round** | **click** | **tense** | **long** |
| 13.1% | 0.28% | 37.4% | 2.61% |

Table 8.4: Phone distribution of features

## 8.5 Conclusion

In this work, we introduce a new multilingual phonetic dataset for low resource languages. The dataset is prepared from an online archive by two steps alignment. The dataset contains around 100 languages and 7000 utterances, and would be released to the community to benefit speech research in low resource phone recognition.

# Chapter 9

# Alignment: All Language Quick Alignment


## Summary

In this chapter, we consider an application in speech alignment using the proposed pipeline. Speech alignment has numerous practical applications, such as constructing text-to-speech datasets, generating automatic captions for online videos, and analyzing phonetics. However, most existing speech aligners have been designed to align rich-resource languages by providing pre-trained acoustic and pronunciation models. These methods are not easily applicable to low-resource languages without training new models or conducting adequate model adaptation.

To address this challenge in this chapter, we propose *ALQAlign*: All Language Quick Alignment that can be applied to most languages globally (7915 languages) without any further training or adaptation. Our method leverages the acoustic model proposed in Chapter 4 and the pronunciation model proposed in Chapter 5, which is then used for the alignment process. We evaluate our method on two different tasks: forced-alignment experiments and text-to-speech experiments, where our approach consistently outperforms the baselines.



Xinjian Li, Ondřej Klejch, Peter Bell, Alan W Black, Shinji Watanabe. Submitted to EMNLP 2023


## 9.1 Introduction

Speech alignment has a wide range of applications in both the fields of speech processing and linguistics. For example, speech alignment can be used to improve speech recognition performance (Rybach et al., 2009), construct speech corpora (Black, 2019; Rousseau et al., 2012),

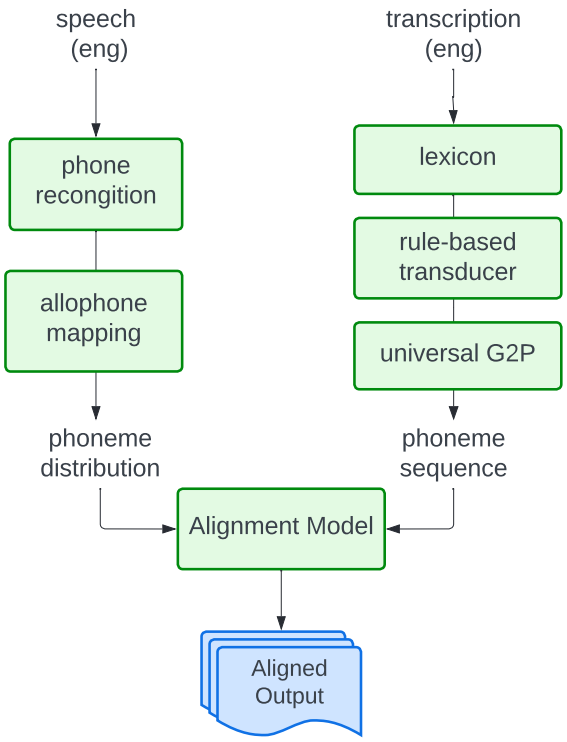



Figure 9.1: Architecture of our toolkit: phonemes are extracted from both the speech and the transcriptions, then those phonemes are aligned with each other modality

improve speaker diarization (Moattar and Homayounpour, 2012), conduct phonetic analyses of endangered languages (DiCanio et al., 2013), perform empirical analyses of sociolinguistics (Pennington et al., 2019), facilitate language documentation, and enable psycholinguistic research (Schillingmann et al., 2018).

Despite its importance, speech alignment tools are limited in their application due to their dependency on language-specific acoustic and pronunciation models, such as grapheme-to-phoneme conversion models. As a result, many aligners have been developed exclusively for rich-resource languages (McAuliffe et al., 2017; Gorman et al., 2011; Yuan et al., 2008; Strunk et al., 2014). Unfortunately, those tools are rarely supported for long-tail low-resource languages, which is estimated to be between 7000-8000 in the world (Lewis, 2016). This limitation highlights the need for more universal and flexible alignment methods that can support a broader range of languages.

In this work, we propose *ALQAlign* (All Language Quick Alignment), which provides an approximated language-specific speech alignment tool for all 7915 languages and dialects registered in Glottolog (Nordhoff and Hammarström, 2011). Our approach relies on the phoneme modality

and is illustrated in Figure 9.1: first, we use an acoustic model to transform the speech input into a phoneme distribution, and then apply a pronunciation model to extract the phoneme sequence from the input transcription. Finally, both phoneme outputs are aligned by our alignment model. While our general workflow is a standard alignment pipeline, our acoustic and pronunciation models are multilingual models that provide an approximated model for every target language (7915 languages) by default. Therefore, it can provide a language-specific model without requiring training on every target language. In the experiment section, we demonstrate that our approach significantly outperforms baseline models on two tasks: forced-alignment and text-to-speech alignment.

## 9.2 Related Work

There are two main families of approaches to aligning audio and transcription. The first family uses a pretrained acoustic model to transform given speech into textual modalities, then the given text is also transformed into the same representation, and alignment between both outputs is estimated using Viterbi algorithm (Anguera et al., 2014; Bordel et al., 2012). There is a varities of difference across those approaches, for example, the input type can be raw audio signals, Mel-frequency cepstral coefficients (MFCC) (Kelley and Tucker, 2018), or self-supervised features (Zhu et al., 2022), the pre-trained acosutic models can be either HMM-based architectures (Lamere et al., 2003; Strunk et al., 2014) or end-to-end neural networks (Watanabe et al., 2021; Kürzinger et al., 2020). The textual representation can be utterances, graphemes, phonemes, or subwords (Stan et al., 2016; McAuliffe et al., 2017; Kürzinger et al., 2020; Watanabe et al., 2021). The existing alignment tools are typically implemented with some speech frameworks such as HTK (Young et al., 2002), Kaldi (Povey et al., 2011) or ESPnet (Watanabe et al., 2018). The second family, on the other hand obtains the audio signals by synthesizing transcriptions, then aligns the original audio with the generated audio using Dynamic Time Warping (DTW)(Malfrère et al., 2003; Black, 2019; Pettarin, 2017). While both groups have achieved some success in obtaining usable alignments, they typically require some prior customized models of the target languages, which limits the scope of their application. In this work, we present a model which can be applied to many languages without any specific customization such as adaptation or pretraining. Note there are a few works tackling similar problems, for example, ReadAlong Studio is a zero-shot tool that uses a cross-lingual G2P and an English acoustic model to align indigenous language audiobooks (Littell et al., 2022).

## 9.3 Toolkit

Our alignment toolkit is divided into the *acoustic model*, *pronunciation model* and *alignment model*. Both Acoustic model and pronunciation model transforms inputs into phoneme representations, which is then aligned by the alignment model.

### 9.3.1 Acoustic Model

The acoustic model proposed here is the model we introduced in the previous chapter 4. Our acoustic model is a multilingual phoneme-based model. Rather than training language-specific acoustic models for each language, we use an allophone-based multilingual architecture to approximate the acoustic model for every language (Li et al., 2020a, 2021a). This architecture consists of two modules: a *universal phone recognition module* and an *allophone mapping module*. The universal phone recognition module first attempts to recognize physical-level phones (i.e. language-independent phonetic units) from the given audio. The allophone mapping module then transforms these language-independent units into language-specific phonemes. This method approximates the acoustic model for each language using a shared set of universal phonetic units.

### 9.3.2 Pronunciation Model

We propose a three-level combination to fully leverage existing pronunciation and grapheme-to-phoneme (G2P) resources for a broader range of languages. Our toolkit integrates three methods into the model: a lexicon-based method, a rule-based model, and a multilingual G2P model. These models decrease in accuracy in the aforementioned order, but increase the scope of target languages, with the last G2P model providing an approximation to any of the 7915 languages by default.

**Lexicon-based Method** obtaining high-quality dictionaries for each language is a challenging task, and low-resource languages often lack such resources. To address this challenge, we leveraged Wiktionary website (Deri and Knight, 2016) as our primary source of pronunciation data, resulting in 1.6 million entries spanning 874 languages.

**Rule-based Model** While lexicon-based information is accurate, the dictionary does not necessarily cover every possible word forms in each language. To address this challenge, we use the rule-based model as our next step. In this work, we apply Epitran (Mortensen et al., 2018), which covers nearly 100 languages.

**Multilingual G2P Model** The G2P model here is the model we introduced in the previous chapter 5. Since the majority of languages lack accessible dictionaries or rules, we use an approximated pronunciation model. Specifically, we apply a recently proposed multilingual grapheme-to-

phoneme (G2P) model as our pronunciation model (Li et al., 2022b): For any unseen languages during inference, this G2P model selects the top-$k$ nearest languages for which a training set is available in Wiktionary, then it proposes $k$ hypotheses using each of the nearest language's model. Those outputs are then ensembled into a lattice to emit the most likely approximated sequence.

### 9.3.3 Alignment Model

The alignment model used in our experiments is derived from CTC-segmentation (Kürzinger et al., 2020). While the original model is mainly designed for aligning characters or subwords, we modified it to handle the phoneme modality specifically. In our setting, the model takes the CTC phoneme logits from the acoustic model and the phoneme sequence from the pronunciation model. It then constructs a lattice to run forward propagation. The timestamps of each phoneme are decided by backtracking from the most likely timestamp of the last phoneme.

## 9.4 Forced Alignment Experiment

We conduct two sets of experiments: forced alignment and text-to-speech alignment. The forced alignment experiment measures the quality of alignment at the phoneme/word level, while the text-to-speech alignment experiment measures the utterance alignment.

### 9.4.1 Dataset

Datasets with precise timestamps for phonemes or words are relatively easy to find for rich-resource languages (Garofolo et al., 1993). However, finding datasets of low-resource is a challenging task. To evaluate the forced-alignment experiments over low-resource languages, we constructed a forced-alignment dataset from the UCLA Phonetic Corpus (Li et al., 2021c). This dataset contains 95 low-resource languages, each with nearly 100 audio clips reading a single word. To develop a forced-alignment evaluation dataset, we concatenated all audio clips of the same language to create a single long audio clip and apply the alignment.

### 9.4.2 Baseline

We compared our method with several baselines, including the Montreal Forced Aligner (McAuliffe et al., 2017), ProsodyLab Aligner (Gorman et al., 2011), FAVE Aligner (Rosenfelder et al., 2014) and Penn Aligner (Yuan et al., 2008). Since most of those aligners typically only support English, the low-resource languages we are targeting are not supported by default. To enable a fair comparison, we adapted the English models by mapping each phone in our transcription to the

| Alignment Model | Precision | Recall | F1 | max F1 | min F1 | std F1 |
|---|---|---|---|---|---|---|
| MFA (McAuliffe et al., 2017) | 0.57 | 0.57 | 0.57 | 1.00 | 0.00 | 0.27 |
| ProsodyLab Aligner (Gorman et al., 2011) | 0.67 | 0.67 | 0.67 | 0.96 | 0.26 | 0.16 |
| FAVE Aligner (Rosenfelder et al., 2014) | - | - | - | - | - | - |
| Penn Aligner (Yuan et al., 2008) | 0.68 | 0.68 | 0.68 | 1.00 | 0.24 | 0.16 |
| ALQAlign (this work) | **0.78** | **0.78** | **0.78** | **1.00** | **0.41** | **0.15** |

Table 9.1: Results of forced-alignment performance on the UCLA dataset measured by the average scores across 95 languages. The proposed aligner has outperformed other base aligners in all categories.

closest English phoneme (i.e., ARPABET) by measuring the phonological distance using PanPhon (Mortensen et al., 2016b). Note that this adaptation process itself is a nontrivial task, which highlights the benefits of our toolkit. For the evaluation metric, we adapted the evaluation script from the MGB-challenge (Bell et al., 2015) by measuring the difference in onsets.

### 9.4.3 Results

We now present the results of the forced-alignment evaluation on the UCLA dataset in Table 9.1. Our approach achieves an average F1 score of 0.78, which significantly outperforms all of the baselines. For example, the MFA model achieves an F1 score of 0.57, while the ProsodyLab Aligner achieves an F1 score of 0.67. In our experiment, FAVE Aligner is not able to identify alignments successfully, it was able to identify shorter alignment when transcription only contains a few phonemes, but is not able to find successful path in our longer settings even by tuning the search size. By analyzing the results, we find voiceless fricatives (variants of [s] such as $[s^{\mathrm{h}}], [s^{\mathrm{j}}], [s']$) are among the most challenging phonemes to align. This difficulty arises because some of these phonemes are rarely seen in our training languages. Consequently, our acoustic model does not perform well in recognizing these phonemes. For instance, $[s']$ is a highly misaligned phoneme in our experiment. It is an ejective consonant that phonemically contrasts with pulmonic consonants, but does not appear in our training set.

| Alignment Model | MCD | max MCD | min MCD | std MCD |
|---|---|---|---|---|
| Wilderness Aligner (Black, 2019) | 7.32 | 8.59 | 6.13 | 0.70 |
| ALQAlign (this work) | 6.43 | 7.99 | 5.51 | 0.64 |

Table 9.2: Comparison of our toolkit with the original aligner used in the CMU Wilderness corpus. We measure the alignment quality by building TTS model and evaluate the MCD scores(a lower score indicates a better performance).

## 9.5 Text-to-Speech Alignment Experiment

### 9.5.1 Dataset

The dataset we evaluate on is the CMU Wilderness corpus and its raw datasets (Black, 2019), which comprise recordings of readings of the New Testament in 700 languages. The raw datasets were collected by crawling the Bible.is website. In each language, the readings are organized into chapters, with each chapter containing raw text and corresponding audio that is usually a few minutes long. We use our toolkit to extract utterance alignments in every chapter.

### 9.5.2 Baseline

In this experiment, the transcriptions are in text form, which requires an appropriate pronunciation model to transform them into phoneme sequences. Unlike our proposed model, the baseline aligners in the previous section cannot handle orthography in unknown languages. Instead of using those aligners, we compare our method to the original aligner proposed in the CMU Wilderness corpus dataset (Black, 2019). This aligner synthesizes transcriptions to obtain the audio signals and then aligns the original audio with the generated audio. We compare our alignment with the first pass alignment of the original aligner since both alignments are not adapted to the data. As there are no golden labels or timestamps for the dataset, it is challenging to directly measure the quality of the aligned results. Therefore, we build TTS models from the aligned text-speech sentence pairs. A better TTS model indicates better alignment quality. The TTS model is evaluated using the Mel cepstral distortion (MCD) score, which compares the synthesized audio with the ground truth audio from the test set. A lower MCD score indicates a better model.

### 9.5.3 Result

Table.9.2 presents the results of our experiment. Our model achieved an average MCD score of 6.43, which is 0.91 MCD lower than the baseline aligner's average score of 7.34. This suggests

that the TTS model using our aligned datasets achieves better resynthesis quality than the original aligned datasets. In addition to the mean MCD score, our model also improves the maximum and minimum MCD scores. Furthermore, it improves the standard deviation, indicating that our model is more robust across a variety of languages.

Upon further investigation, we find that the performance of the original aligner is worse when the orthography of the target language is not consistent with the aligner's pronunciation rule. It is common for different languages to have different pronunciations for the same grapheme. For example, the letter <h> in "hello" is pronounced with the [h] phoneme in English but is not pronounced explicitly in "hola" in Spanish. The original aligner assumes a universal pronunciation rule regardless of languages, meaning that the letter "h" is always considered [h] in every language. This rule fails when applied to Spanish. In contrast, our model considers the differences across languages and assigns different phonemes depending on the language.

## 9.6 Conclusion

In this work, we propose ALQAlign, All Language Quick Alignment to apply speech alignment to most languages globally without training or adaptation. Our method relies on approximated acoustic model and pronunciation model, which outperform baseline aligner significantly over two evaluation tasks. Our software will be released from Github.

# Chapter 10

# Conclusion

As we have seen so far, most recent speech recognition technology relies on large supervised datasets, which are limited to a few hundreds of high-resource languages. However, there are 8000 languages in the world, the majority of which are low-resource languages and large supervised datasets are not available. Therefore, the traditional speech recognition pipeline cannot be applied to them directly, which significantly restricts the scope of target languages. In this thesis, we present a speech recognition pipeline that attempts to reduce the dataset requirement as much as possible. In particular, we consider a pipeline that does not require any audio datasets for the target language. This assumption enables us to expand the scope of target languages to around 6000 languages.

In the first part of this thesis, we propose a multilingual acoustic model which can be trained from high-resource languages and applied to low-resource languages without any audio supervision. In the second part of this thesis, we discuss the pronunciation model and the language model. The pronunciation model is a grapheme-to-phoneme conversion model which can also be learned without any supervision from the target language. The language model is the n-gram model built from a large online n-gram database. In the last part, we introduce two datasets and one application.

When I initially embarked on my work in multilingual speech recognition in 2017, the predominant focus of research was on training monolingual models for a handful of languages with abundant resources (Xiong et al., 2018). The available multilingual corpora at the time, such as voxforge (Voxforge.org) and the BABEL project (Harper, 2011), were limited in their language coverage and hours of training datasets. During my involvement in the LoReHLT project (Chaudhary et al., 2019), I became aware that numerous languages worldwide lacked speech systems due to the scarcity of supervised datasets. Recognizing the potential significance of developing speech recognition for thousands of languages, I shifted my focus. Instead of relying on supervised datasets, I found it considerably more feasible to obtain phonetic information for a broad range of long-tail languages (Moran and McCloy, 2019) and identified phonemes as a promis-

ing foundation for expanding language coverage. Our approach involved enhancing the existing phonetic corpus by annotating allophones (Mortensen et al., 2020) and broadening language coverage (Li et al., 2021b). Subsequently, we devised universal speech systems towards multilingual phoneme recognition (Li et al., 2020a, 2021a). To establish a fully operational speech recognition pipeline, we devised a universal grapheme-to-phoneme(G2P) model for phoneme translation (Li et al., 2020d). The culmination of these efforts resulted in the development of a functional pipeline capable of supporting thousands of languages (Li et al., 2022a).

In recent years, an increasing number of research groups have begun exploring avenues to expand the range of languages covered in speech recognition systems. Notably, OpenAI's Whisper system has made significant strides by encompassing approximately 100 languages (Radford et al., 2022), while Google's USM system offers a similar coverage (Zhang et al., 2023) Meta's MMS system surpasses them both by providing support for a remarkable 1000 languages (Pratap et al., 2023). The primary distinction between our approach and theirs lies in the reliance on datasets. While all of these systems are trained end-to-end using supervised datasets, our model solely depends on text data without the inclusion of any audio datasets for the target language. The advantage of our pipeline lies in its broader language coverage (6000 languages compared to 1000), but it does come with a drawback in terms of recognition accuracy (10% CER compared to 40% CER).

As the availability of supervised datasets approaches its upper limit, our text-only approach is poised to remain the only viable option for achieving such extensive language coverage. However, as previously discussed, there are many limitations in our pipeline. We will discuss the limitations and potential future works of the acoustic model and language model separately in the following sections.

## 10.1 Acoustic Model Discussion

We first discuss a few limitations and potential future directions of the acoustic model in this section.

### 10.1.1 Limitations of Annotations

The multilingual model proposed in Chapter 3 is largely dependent on manually curated allophone (phone-phoneme) annotations (Mortensen et al., 2020). This approach presents two significant challenges:

**Limitations on Language Diversity** The model, as explained in Chapter 3, has been trained with annotations from approximately 10 languages. However, a vast number of languages avail-

able could potentially enhance the phone recognition model. Relying solely on manually curated annotations makes it costly to incorporate additional languages into the model.

**Inefficiencies in Annotation** As the model encompasses more languages, the cost and time of annotation for each language will increase correspondingly. As previously discussed in this dissertation, we estimate the universal phone inventory, denoted as $Q_{\text{uni}}$, as follows:

$$Q_{\text{uni}} = \bigcup_{1 \leq i \leq |L|} \bigcup_{p \in P_i} Q_p^i \tag{10.1}$$

The phone inventory size, denoted as $Q_{\text{uni}}$, in this experiment ranges from 200 to 300. However, we observe that this number tends to increase rapidly as we increase the number of languages $|L|$. From the PHOIBLE dataset (Moran and McCloy, 2019), we estimate that there are over 2000 distinct phones. As such, it would prove inefficient, even infeasible, to verify if every phone from the inventory could be a variant allophone for a given phoneme.

Rather than depending on manually curated annotations, as future work, we propose to automate their learning through a problem we term as the *Automatic Allophone Estimation*. Here, we discuss our formulation and some potential future directions.

In this dissertation, we've established the signature matrix as $S = \{0, 1\}^{|P| \times |Q_{\text{uni}}|}$ where $P$ is the phoneme inventory for a particular language and $Q_{\text{uni}}$ is the complete phone inventory. The signature matrix, primarily predefined by our annotation, remains mostly stable throughout training, although it is subject to occasional fine-tuning. Under this model, let the acoustic model be a neural network parameterized with $\theta$. We optimize the model by minimizing the empirical risk of $L(\theta)$ over the multilingual training set $\mathcal{D}$ with established signature matrices $S$.

$$\text{minimize}_\theta \mathbb{E}_{\mathcal{D}}[L(\theta; S)] \tag{10.2}$$

This optimization problem outlined above only depends on $\theta$. However, if the signature matrix $S$ is unknown, we can extend the formulation by including it as well. Put simply, our objective should minimize the empirical risk with both parameters considered.

$$\text{minimize}_{S,\theta} \mathbb{E}_{\mathcal{D}}[L(S, \theta)] \tag{10.3}$$

The precise solution to this augmented optimization problem is computationally unfeasible, as the original problem $L(\theta)$—a special case of this new problem—is already intractable. The revised optimization problem is subject to more intricate constraints, significantly complicating the task. Several potential pathways could be explored to address this issue. For instance, the Expectation-Maximization (EM) algorithm could be employed to iteratively optimize with respect to both $\theta$ and $S$ after each training cycle.

### 10.1.2 Suprasegmentals Problems

This dissertation primarily focuses on phonemes, phones, or segments, and does not incorporate suprasegmental features such as stress, pitch, and intonation in the model's pipeline. These features, which often extend across multiple consonants or vowels, play vital roles in many languages (Leben, 1973). For instance, Mandarin uses tonal distinctions, with four primary tones and one neutral tone, to differentiate the meanings of words. Explicit modeling of these features can enhance speech processing and analysis (Mehler, 1981).

The primary challenge in creating a universal suprasegmental system could be the limitations of the available datasets and the absence of a unifying framework, as each language exhibits unique suprasegmental characteristics. While it is feasible to gather a large Mandarin and Vietnamese dataset for training a monolingual tonal recognition model (Hui Bu, 2017; Yuan et al., 2021), it becomes substantially more challenging to acquire a similarly large dataset for other lower-resource tonal languages, such as Cherokee (Lewis, 2016). Additionally, to devise a universal suprasegmental model, we should aim to develop a unified framework capable of accommodating differences across languages.

### 10.1.3 Self-Supervised Learning Models

In Chapter 6, we leverage self-supervised learning models to enhance our acoustic model. We have discovered that self-supervised models trained on a multilingual dataset improve the performance of our acoustic model (Conneau et al., 2020). The value of multilingual self-supervised models arises from their exposure to a larger array of languages and a broader phonetic context. Given the recent availability of expansive multilingual datasets (Conneau et al., 2023; Pratap et al., 2023), it might be worthwhile to train self-supervised models on hundreds of languages and investigate their potential impact on our pipeline.

Another intriguing direction for future research could be to explore how to utilize pre-trained discrete representations, and correlate these intermediate tokens with the phonological tokens we have modeled in this dissertation (Zeghidour et al., 2021; Défossez et al., 2022). These intermediate tokens, given their generalization over vast training sets, may serve as better discrete representations than the explicit phone tokens currently in use.

## 10.2 Language Model Discussion

Next, we discuss the limitation and some potential future directions in the language model in our pipeline.

### 10.2.1 Limitation of Pronunciation Models

In our pipeline as described in Chapter 5, we employ tree-based metrics to gauge the distance between languages (Nordhoff and Hammarström, 2011), under the assumption that languages from the same branch of a linguistic family likely share similar writing systems and pronunciation rules. However, this assumption doesn't always accurately capture the real degree of closeness. The phylogenetic metric only measures one aspect of linguistic distance. Incorporating other metrics, such as typological and geographical distance, may enhance the accuracy of closeness measurements (Littell et al., 2017). Moreover, we operate on the assumption that each language possesses only a single writing system, an assumption that doesn't always hold true. In fact, a single language, such as Uzbek, can have multiple writing scripts (Perso-Arabic, Cyrillic, and Latin) used by different groups and in different contexts.

In our pronunciation model, the number of languages used for ensembling is set at 10. However, this number does not necessarily need to be constant and it might be more beneficial to adjust the number dynamically based on each language. For instance, for languages with fewer neighboring languages, we might want to utilize a larger ensemble. Conversely, for languages with numerous neighbors, a smaller ensemble might suffice.

### 10.2.2 Limitation of Language Models

While we assert that we've developed acoustic models, pronunciation models, and language models for thousands of languages, accurately measuring the true coverage of our pipeline is challenging, given that we lack testing datasets for every language. The largest testing dataset utilized in this work is the CMU Wilderness dataset (Black, 2019), and we estimate the zero-shot performance based on its outcomes. However, this may not reflect the true performance across the long tail of languages. The quality of datasets often decreases along this tail, presenting a typical trade-off between the number of languages covered and the quality of performance (Kamholz et al., 2014). Even the task of correctly identifying languages presents significant challenges (Caswell et al., 2020; Bapna et al., 2022). To provide a more accurate estimation, it's worth considering the creation of new datasets, which should not only sample from the first 1000 languages but also more frequently cover the long tail of languages.

Furthermore, in Chapter 6, we rely on traditional n-gram models as our language modeling module. The primary reason being that our main language resources are lexicon and n-gram counts (Kamholz et al., 2014; Scannell, 2007). However, the capabilities of such n-gram models are inherently limited due to the size of each monolingual dataset. A potential avenue for future research could involve leveraging large language models like GPT and PaLM (OpenAI, 2023; Chowdhery et al., 2022) to facilitate knowledge transfer from high-resource languages to those with fewer resources.

# Bibliography


Solomon Teferra Abate, Wolfgang Menzel, and Bairu Tafila. 2005. An amharic speech corpus for large vocabulary continuous speech recognition. In *INTERSPEECH-2005*.

Oliver Adams, Trevor Cohn, Graham Neubig, Hilaria Cruz, Steven Bird, and Alexis Michaud. 2018. Evaluating phonemic transcription of low-resource tonal languages for language documentation. In *Proc. LREC*.

Simon Ager. 2008. Omniglot-writing systems and languages of the world. *Retrieved January*, 27:2008.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *Proc. ICML*.

Xavier Anguera, Jordi Luque, and Ciro Gracia. 2014. Audio-to-text alignment for speech recognition with very limited resources. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4218–4222.

Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. 2017. Deep voice: Real-time neural text-to-speech. In *International Conference on Machine Learning*, pages 195–204. PMLR.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised speech recognition. *Advances in Neural Information Processing Systems*, 34.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, et al. 2022. Building machine translation systems for the next thousand languages. *arXiv preprint arXiv:2205.03983*.

Peter Bell, Mark JF Gales, Thomas Hain, Jonathan Kilgour, Pierre Lanchantin, Xunying Liu, Andrew McParland, Steve Renals, Oscar Saz, Mirjam Wester, et al. 2015. The mgb challenge: Evaluating multi-genre broadcast media recognition. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 687–693. IEEE.

Steven Bird and Gary Simons. 2003. Seven dimensions of portability for language documentation and description. *Language*, pages 557–582.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451.

Alan W Black. 2019. CMU wilderness multilingual speech dataset. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5971–5975. IEEE.

Alan W Black, Kevin Lenzo, and Vincent Pagel. 1998. Issues in building general letter to sound rules. In *The third ESCA/COCOSDA workshop (ETRW) on speech synthesis*.

Harry Bleyan, Sandy Ritchie, Jonas Fromseier Mortensen, and Daan van Esch. 2019. Developing pronunciation models in new languages faster by exploiting common grapheme-to-phoneme correspondences across languages. In *INTERSPEECH*, pages 2100–2104.

Germán Bordel, Mikel Penagarikano, Luis Javier Rodriguez-Fuentes, and Amparo Varona. 2012. A simple and efficient method to align very long speech signals to acoustically imperfect transcriptions. In *Thirteenth Annual Conference of the International Speech Communication Association*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language id in the wild: Unexpected challenges on the path to a thousand-language web text corpus. *arXiv preprint arXiv:2010.14571*.

Milos Cernak and Philip N Garner. 2016. Phonvoc: A phonetic and phonological vocoding toolkit. In *Proc. Interspeech*.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proc. ICASSP*.

Aditi Chaudhary, Siddharth Dalmia, Junjie Hu, Xinjian Li, Austin Matthews, Aldrian Obaja Muis, Naoki Otani, Shruti Rijhwani, Zaid Sheikh, Nidhi Vyas, et al. 2019. The ariel-cmu systems for lorehlt18. *arXiv preprint arXiv:1902.08899*.

Dongpeng Chen and Brian Kan-Wing Mak. 2015. Multitask learning of deep neural networks for low-resource speech recognition. *TASLP*, 23(7):1172–1183.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper& Row, New York.

Jan Chorowski, Ron J Weiss, Samy Bengio, and Aäron Van Den Oord. 2019. Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM transactions on audio, speech, and language processing*, 27(12):2041–2053.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Christopher Cieri, David Miller, and Kevin Walker. 2004. The fisher corpus: a resource for the next generations of speech-to-text. In *Proc. LREC*.

CMU. 2000. The CMU pronunciation dictionary.

Paul S. Cohen, Satyanarayana Dharanipragada, Jerneja Zganec Gros, Michael Daniel Monkowski, Chalapathy Neti, Salim Roukos, and Todd Ward. 1997. Towards a universal speech recognizer for multiple languages. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 591–598. IEEE.

Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. 2016. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2020. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2009. *Introduction to algorithms*. MIT press.

Florian Coulmas. 2013. *Written and unwritten language*, Key Topics in Sociolinguistics, page 39–59. Cambridge University Press.

Nicole Creanza, Merritt Ruhlen, Trevor J Pemberton, Noah A Rosenberg, Marcus W Feldman, and Sohini Ramachandran. 2015. A comparison of worldwide phonemic and genetic variation in human populations. *Proceedings of the National Academy of Sciences*, 112(5):1265–1272.

Siddharth Dalmia, Ramon Sanabria, Florian Metze, and Alan W Black. 2018. Sequence-based multi-lingual low resource speech recognition. In *Proc. ICASSP*.

Donald M Decker et al. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.

Aliya Deri and Kevin Knight. 2016. Grapheme-to-phoneme models for (almost) any language. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 399–408.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Christian DiCanio, Hosung Nam, Douglas H Whalen, H Timothy Bunnell, Jonathan D Amith, and Rey Castillo García. 2013. Using automatic alignment to analyze endangered language data: Testing the viability of untrained alignment. *The Journal of the Acoustical Society of America*, 134(3):2235–2246.

Zhiyong Zhang Dong Wang, Xuewei Zhang. 2015. Thchs-30 : A free chinese speech corpus.

Matthew S Dryer and Martin Haspelmath. 2013. The world atlas of language structures online.

Ewan Dunbar, Robin Algayres, Julien Karadayi, Mathieu Bernard, Juan Benjumea, Xuan-Nga Cao, Lucie Miskic, Charlotte Dugrain, Lucas Ondel, Alan Black, et al. 2019. The zero resource speech challenge 2019: Tts without t. In *Interspeech 2019-20th Annual Conference of the International Speech Communication Association*.

Ewan Dunbar, Julien Karadayi, Mathieu Bernard, Xuan-Nga Cao, Robin Algayres, Lucas Ondel, Laurent Besacier, Sakriani Sakti, and Emmanuel Dupoux. 2020. The zero resource speech challenge 2020: Discovering discrete subword and word units. *arXiv preprint arXiv:2010.05967*.

Jonathan G Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 347–354. IEEE.

Ram Frost and Marian Katz. 1992. *Orthography, phonology, morphology and meaning*. Elsevier.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. 1993. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *STIN*, 93:27403.

Jost Gippert, Nikolaus Himmelmann, Ulrike Mosel, et al. 2006. *Essentials of language documentation*, volume 178. Walter de gruyter.

James Glass. 2012. Towards unsupervised speech processing. In *Proc. ISSPA*.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proc. ICASSP*.

Kyle Gorman, Lucas FE Ashby, Aaron Goyzueta, Arya D McCarthy, Shijie Wu, and Daniel You. 2020. The sigmorphon 2020 shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 40–50.

Kyle Gorman, Jonathan Howell, and Michael Wagner. 2011. Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3):192–193.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition.

Akshat Gupta, Xinjian Li, Sai Krishna Rallabandi, and Alan W Black. 2021. Acoustics based intent recognition using discovered phonetic units for low resource languages. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7453–7457. IEEE.

Mary Harper. 2011. The iarpa babel multilingual speech database.

Tomoki Hayashi, Ryuichi Yamamoto, Takenori Yoshimura, Peter Wu, Jiatong Shi, Takaaki Saeki, Yooncheol Ju, Yusuke Yasuda, Shinnosuke Takamichi, and Shinji Watanabe. 2021. Espnet2-tts: Extending the edge of tts research. *arXiv preprint arXiv:2110.07840*.

Bruce Hayes. 2011. *Introductory phonology*, volume 32. John Wiley & Sons.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Michael Heck, Sakriani Sakti, and Satoshi Nakamura. 2017. Feature optimized dpgmm clustering for unsupervised subword modeling: A contribution to zerospeech 2017. In *Proc. ASRU*.

Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and Jeffrey Dean. 2013. Multilingual acoustic models using distributed deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8619–8623. IEEE.

Enno Hermann and Sharon Goldwater. 2018. Multilingual bottleneck features for subword modeling in zero-resource languages. In *Proc. Interspeech*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7304–7308. IEEE.

Xingyu Na Bengu Wu Hao Zheng Hui Bu, Jiayu Du. 2017. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *Oriental COCOSDA 2017*, page Submitted.

International Phonetic Association, International Phonetic Association Staff, et al. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Roman Jakobson, C Gunnar Fant, and Morris Halle. 1951. *Preliminaries to speech analysis: The distinctive features and their correlates*. MIT press.

Aren Jansen, Emmanuel Dupoux, Sharon Goldwater, Mark Johnson, Sanjeev Khudanpur, Kenneth Church, Naomi Feldman, Hynek Hermansky, Florian Metze, Richard Rose, et al. 2013. A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition. In *Proc. ICASSP*.

Sittichai Jiampojamarn and Grzegorz Kondrak. 2010. Phoneme alignment: An exploration. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 780–788.

Yishan Jiao, Visar Berisha, and Julie Liss. 2017. Interpretable phonological features for clinical applications. In *Proc. ICASSP*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351.

David Kamholz, Jonathan Pool, and Susan M Colowick. 2014. Panlex: Building a resource for panlingual lexical translation. In *LREC*, pages 3145–3150.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. 2019. A comparative study on transformer vs rnn in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 449–456. IEEE.

Matthew C Kelley and Benjamin V Tucker. 2018. A comparison of input types to a deep neural network-based forced aligner. *Proc. Interspeech 2018*, pages 1205–1209.

Katrin Kirchhoff. 1998. Combining articulatory and acoustic information for speech recognition in noisy and reverberant environments. In *Proc. ICSLP*.

Katherine Mary Knill, Mark John Gales, Anton Ragni, and Shakti P Rath. 2014. Language independent and unsupervised acoustic models for speech recognition and keyword spotting. In *Proc. Interspeech*.

Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll. 2020. Ctc-segmentation of large corpora for german end-to-end speech recognition. In *Speech and Computer: 22nd International Conference, SPECOM 2020, St. Petersburg, Russia, October 7–9, 2020, Proceedings*, pages 267–278.

Peter Ladefoged, B Barbara, and GS Russell. 2009. Ucla phonetics lab archive.

Peter Ladefoged and Keith Johnson. 2014. *A course in phonetics*. Nelson Education.

Brenden Lake, Chia-ying Lee, James Glass, and Josh Tenenbaum. 2014. One-shot learning of generative speech concepts. In *Proc. CogSci*.

Paul Lamere, Philip Kwok, Evandro Gouvea, Bhiksha Raj, Rita Singh, William Walker, Manfred Warmuth, and Peter Wolf. 2003. The cmu sphinx-4 speech recognition system. In *Ieee intl. conf. on acoustics, speech and signal processing (icassp 2003), hong kong*, volume 1, pages 2–5.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 951–958. IEEE.

William Ronald Leben. 1973. *Suprasegmental phonology.* Ph.D. thesis, Massachusetts Institute of Technology.

Jackson L Lee, Lucas FE Ashby, M Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D McCarthy, and Kyle Gorman. 2020. Massively multilingual pronunciation modeling with wikipron. In *Proceedings of the 12th language resources and evaluation conference*, pages 4223–4228.

M Paul Lewis. 2009. *Ethnologue: Languages of the world*. SIL international.

M. Paul Lewis, editor. 2016. *Ethnologue: Languages of the World*. SIL International, Dallas, TX, USA.

Xinjian Li, Siddharth Dalmia, Alan W Black, and Florian Metze. 2019a. Multilingual speech recognition with corpus relatedness sampling. *Proc. Interspeech 2019*, pages 2120–2124.

Xinjian Li, Siddharth Dalmia, Alan W. Black, and Florian Metze. 2019b. Multilingual speech recognition with corpus relatedness sampling. In *Proc. Interspeech*.

Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R Mortensen, Graham Neubig, Alan W Black, et al. 2020a. Universal phone recognition with a multilingual allophone system. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8249–8253. IEEE.

Xinjian Li, Siddharth Dalmia, David Mortensen, Juncheng Li, Alan Black, and Florian Metze. 2020b. Towards zero-shot learning for automatic phonemic transcription. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8261–8268.

Xinjian Li, Siddharth Dalmia, David R. Mortensen, Juncheng Li, Alan Black, and Florian Metze. 2020c. Towards zero-shot learning for automatic phonemic transcription. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Xinjian Li, Siddharth Dalmia, David R Mortensen, Juncheng Li, Alan W Black, and Florian Metze. 2020d. Towards zero-shot learning for automatic phonemic transcription. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Xinjian Li, Ye Jia, and Chung-Cheng Chiu. 2023. Textless direct speech-to-speech translation with discrete speech representation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Xinjian Li, Juncheng Li, Florian Metze, and W Black Black, Alan. 2021a. Hierarchical phone recognition with compositional phonetics. In *Proc. Interspeech*.

Xinjian Li, Juncheng Li, Jiali Yao, Alan W Black, and Florian Metze. 2021b. Phone distribution estimation for low resource languages. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7233–7237. IEEE.

Xinjian Li, Florian Metze, David R. Mortensen, Alan W Black, and Shinji Watanabe. 2022a. ASR2K: Speech Recognition for Around 2000 Languages without Audio. In *Proc. Interspeech 2022*, pages 4885–4889.

Xinjian Li, Florian Metze, David R Mortensen, Shinji Watanabe, and Alan W Black. 2022b. Zero-shot learning for grapheme to phoneme conversion with language ensemble. *To be appearing at Findings of ACL*.

Xinjian Li, David R Mortensen, Florian Metze, and Alan W Black. 2021c. Multilingual phonetic dataset for low resource speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6958–6962. IEEE.

Hui Lin, Li Deng, Dong Yu, Yi-fan Gong, Alex Acero, and Chin-Hui Lee. 2009. A study on multilingual acoustic modeling for large vocabulary ASR. In *Proc. ICASSP*.

Patrick Littell, Eric Joanis, Aidan Pine, Marc Tessier, David Huggins-Daines, and Delasie Torkornoo. 2022. Readalong studio: Practical zero-shot text-speech alignment for indigenous language audiobooks. In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 23–32.

Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14.

Yi Liu, Pascale Fung, Yongsheng Yang, Christopher Cieri, Shudong Huang, and David Graff. 2006. Hkust/mts: A very large scale mandarin telephone speech corpus. In *Proc. ISCSLP*.

Kikuo Maekawa. 2003. Corpus of spontaneous japanese: Its design and evaluation. In *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*.

Fabrice Malfrère, Olivier Deroo, Thierry Dutoit, and Christophe Ris. 2003. Phonetic alignment: speech synthesis-based vs. viterbi-based. *Speech Communication*, 40(4):503–515.

Lidia Mangu, Eric Brill, and Andreas Stolcke. 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400.

Pavel Matejka, Petr Schwarz, Jan Cernockỳ, and Pavel Chytil. 2005. Phonotactic language identification using high quality phoneme recognition. In *Ninth European Conference on Speech Communication and Technology*.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In *Proc. Interspeech 2017*, pages 498–502.

Jacques Mehler. 1981. The role of syllables in speech processing: Infant and adult data. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 295(1077):333–352.

Yajie Miao, Mohammad Gowayyed, and Florian Metze. 2015. 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)*, pages 167–174. IEEE.

Alexis Michaud, Oliver Adams, Trevor Anthony Cohn, Graham Neubig, and Séverine Guillaume. 2018. Integrating automatic transcription into the language documentation workflow: Experiments with na data and the persephone toolkit. *LD&C*.

Mohammad Hossein Moattar and Mohammad M Homayounpour. 2012. A review on speaker diarization systems and approaches. *Speech Communication*, 54(10):1065–1103.

Steven Moran and Daniel McCloy, editors. 2019. *PHOIBLE 2.0*. Max Planck Institute for the Science of Human History, Jena.

David R Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision g2p for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

David R Mortensen, Xinjian Li, Patrick Littell, Alexis Michaud, Shruti Rijhwani, Antonios Anastasopoulos, Alan W Black, Florian Metze, and Graham Neubig. 2020. Allovera: A multilingual allophone database. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5329–5336.

David R Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016a. Panphon: A resource for mapping ipa segments to articulatory feature vectors. In *Proc. COLING*.

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori S. Levin. 2016b. Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484. ACL.

David R. Mortensen, Jordan Picone, Xinjian Li, and Kathleen Siminyu. 2021. Tusom2021: A Phonetically Transcribed Speech Dataset from an Endangered Language for Universal Phone Recognition Experiments. In *Proc. Interspeech 2021*, pages 3660–3664.

Markus Müller, Jörg Franke, Sebastian Stüker, and Alex Waibel. 2017a. Improving phoneme set discovery for documenting unwritten languages. *Elektronische Sprachsignalverarbeitung (ESSV)*, 2017.

Markus Müller, Jörg Franke, Alex Waibel, and Sebastian Stüker. 2017b. Towards phoneme inventory discovery for documentation of unwritten languages. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5200–5204. IEEE.

Markus Müller, Sebastian Stüker, and Alex Waibel. 2016. Towards improving low-resource speech recognition using articulatory and language features. In *Proc. IWSLT*.

Daniel Nettle, Suzanne Romaine, et al. 2000. *Vanishing voices: The extinction of the world's languages*. Oxford University Press on Demand.

Graham Neubig, Patrick Littell, Chian-Yu Chen, Jean Lee, Zirui Li, Yu-Hsiang Lin, and Yuyan Zhang. 2018. Towards a general-purpose linguistic annotation backend. *arXiv preprint arXiv:1812.05272*.

Graham Neubig, Shruti Rijhwani, Alexis Palmer, Jordan MacKenzie, Hilaria Cruz, Xinjian Li, Matthew Lee, Aditi Chaudhary, Luke Gessler, Steven Abney, et al. 2020. A summary of the first workshop on language technology for language documentation and revitalization. *arXiv preprint arXiv:2004.13203*.

Sebastian Nordhoff and Harald Hammarström. 2011. Glottolog/langdoc: Defining dialects, languages, and language families as collections of resources. In *First International Workshop on Linked Science 2011-In conjunction with the International Semantic Web Conference (ISWC 2011)*.

Josef Robert Novak, Nobuaki Minematsu, and Keikichi Hirose. 2016. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the wfst framework. *Natural Language Engineering*, 22(6):907–938.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. 2009. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, pages 1410–1418.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an ASR corpus based on public domain audio books. In *Proc. ICASSP*.

Martha C Pennington, Pamela Rogerson-Revell, Martha C Pennington, and Pamela Rogerson-Revell. 2019. Using technology for pronunciation teaching, learning, and assessment. *English Pronunciation Teaching and Research: Contemporary Perspectives*, pages 235–286.

Ben Peters, Jon Dehdari, and Josef van Genabith. 2017. Massively multilingual neural grapheme-to-phoneme conversion. *EMNLP 2017*, page 19.

Alberto Pettarin. 2017. Aeneas.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, CONF. IEEE Signal Processing Society.

Rohit Prabhavalkar, Karen Livescu, Eric Fosler-Lussier, and Joseph Keshet. 2013. Discriminative articulatory models for spoken term detection in low-resource conversational settings. In *Proc. ICASSP*.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. 2023. Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.

Kanishka Rao, Fuchun Peng, Haşim Sak, and Françoise Beaufays. 2015. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4225–4229. IEEE.

Eric Sven Ristad and Peter N Yianilos. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532.

Morgane Rivière, Armand Joulin, Pierre-Emmanuel Mazaré, and Emmanuel Dupoux. 2020. Unsupervised pretraining transfers well across languages. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7414–7418. IEEE.

Bernardino Romera-Paredes and Philip Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *Proc. ICML*.

Andrew Rosenberg, Yu Zhang, Bhuvana Ramabhadran, Ye Jia, Pedro Moreno, Yonghui Wu, and Zelin Wu. 2019. Speech recognition with augmented synthesized speech. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 996–1002. IEEE.

Ingrid Rosenfelder, Josef Fruehwald, Keelan Evanini, Scott Seyfarth, Kyle Gorman, Hilary Prichard, and Jiahong Yuan. 2014. Fave (forced alignment and vowel extraction). *Program suite v1*, 2(10.5281).

Anthony Rousseau, Paul Deléglise, and Yannick Esteve. 2012. TED-LIUM: an automatic speech recognition dedicated corpus. In *Proc. LREC*.

David Rybach, Christian Gollan, Ralf Schluter, and Hermann Ney. 2009. Audio segmentation for speech recognition using segment features. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4197–4200. IEEE.

Kevin P Scannell. 2007. The crubadan project: Corpus building for under-resourced languages. *Cahiers du Cental*, 5:1.

Odette Scharenborg, Patrick W. Ebel, Francesco Ciannella, Mark Hasegawa-Johnson, and Najim Dehak. 2017. Building an asr system for mboshi using a cross-language definition of acoustic units approach. In *Proc. ICNLSSP*.

Lars Schillingmann, Jessica Ernst, Verena Keite, Britta Wrede, Antje S Meyer, and Eva Belke. 2018. Aligntool: The automatic temporal alignment of spoken utterances in german, dutch, and british english for psycholinguistic purposes. *Behavior research methods*, 50:466–489.

Tim Schlippe, Wolf Quaschningk, and Tanja Schultz. 2014. Combining grapheme-to-phoneme converter outputs for enhanced pronunciation generation in low-resource scenarios. In *Spoken Language Technologies for Under-Resourced Languages*.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Proc. Interspeech 2019*, pages 3465–3469.

Tanja Schultz and Alex Waibel. 1997. Fast bootstrapping of lvcsr systems with multilingual phoneme sets. In *Fifth European Conference on Speech Communication and Technology*.

Tanja Schultz and Alex Waibel. 2001. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication*, 35(1-2):31–51.

Terrence J Sejnowski and Charles R Rosenberg. 1987. Parallel networks that learn to pronounce english text. *Complex systems*, 1(1):145–168.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Kathleen Siminyu, Xinjian Li, Antonios Anastasopoulos, David R. Mortensen, Michael R. Marlo, and Graham Neubig. 2021. Phoneme Recognition Through Fine Tuning of Phonetic Representations: A Case Study on Luhya Language Varieties. In *Proc. Interspeech 2021*, pages 271–275.

Sabato Marco Siniscalchi, Dau-Cheng Lyu, Torbjørn Svendsen, and Chin-Hui Lee. 2011. Experiments on cross-language attribute detection and phone recognition with minimal target-specific training data. *TASLP*, 20(3):875–887.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Proc. NIPS*.

Adriana Stan, Yoshitaka Mamiya, Junichi Yamagishi, Peter Bell, Oliver Watts, Robert AJ Clark, and Simon King. 2016. Alisa: An automatic lightly supervised speech segmentation and alignment tool. *Computer Speech & Language*, 35:116–133.

Adriana Stan, Oliver Watts, Yoshitaka Mamiya, Mircea Giurgiu, Robert AJ Clark, Junichi Yamagishi, and Simon King. 2013. Tundra: a multilingual corpus of found data for tts research created with light supervision. In *INTERSPEECH*, pages 2331–2335.

Jan Strunk, Florian Schiel, Frank Seifart, et al. 2014. Untrained forced alignment of transcriptions and audio for language documentation corpora using webmaus. In *LREC*, pages 3940–3947.

Sebastian Stüker, Florian Metze, Tanja Schultz, and Alex Waibel. 2003a. Integrating multilingual articulatory features into speech recognition. In *Proc. Eurospeech*.

Sebastian Stüker, Tanja Schultz, Florian Metze, and Alex Waibel. 2003b. Multilingual articulatory features. In *Proc. ICASSP*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Tatiana Likhomanenko, Edouard Grave, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. 2019. End-to-end asr: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460*.

Samuel Thomas, Sriram Ganapathy, and Hynek Hermansky. 2010. Cross-lingual and multi-stream posterior features for low resource LVCSR systems. In *Proc. Interspeech*.

Jessica AF Thompson, Marc Schönwiesner, Yoshua Bengio, and Daniel Willett. 2019. 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)*, pages 2827–2831. IEEE.

Andros Tjandra, Berrak Sisman, Mingyang Zhang, Sakriani Sakti, Haizhou Li, and Satoshi Nakamura. 2019. VQVAE unsupervised unit discovery and multi-scale code2spec inverter for zerospeech challenge 2019. *arXiv preprint arXiv:1905.11449*.

Sibo Tong, Philip N Garner, and Hervé Bourlard. 2017. An investigation of deep neural networks for multilingual speech recognition training and adaptation. In *Proc. Interspeech*.

Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao. 2018. Multilingual speech recognition with a single end-to-end model. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4904–4908. IEEE.

Balakrishnan Varadarajan, Sanjeev Khudanpur, and Emmanuel Dupoux. 2008. Unsupervised learning of acoustic sub-word units. In *Proceedings of ACL-08: HLT, Short Papers*, pages 165–168.

JC Vásquez-Correa, Philipp Klumpp, Juan Rafael Orozco-Arroyave, and Elmar Nöth. 2019. Phonet: a tool based on gated recurrent neural networks to extract phonological posteriors from speech. In *Proc. Interspeech*.

Maarten Versteegh, Roland Thiolliere, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux. 2015. The zero resource speech challenge 2015. In *Proc. Interspeech*.

Karel Veselỳ, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova. 2012. The language-independent bottleneck features. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 336–341. IEEE.

Karel Veselý, Lukáš Burget, and Jan Černocký. 2017. Semi-Supervised DNN Training with Word Selection for ASR. In *Proc. Interspeech 2017*, pages 3687–3691.

Voxforge.org. Free speech recognition (linux, windows and mac) - voxforge.org. `http://www.voxforge.org/`. Accessed 06/25/2014.

Ngoc Thang Vu and Tanja Schultz. 2013. Multilingual multilayer perceptron for rapid language adaptation between and across language families. In *Proc. Interspeech*.

Shinji Watanabe, Florian Boyer, Xuankai Chang, Pengcheng Guo, Tomoki Hayashi, Yosuke Higuchi, Takaaki Hori, Wen-Chin Huang, Hirofumi Inaguma, Naoyuki Kamo, et al. 2021. The 2020 espnet update: new features, broadened applications, performance improvements, and future plans. In *2021 IEEE Data Science and Learning Workshop (DSLW)*, pages 1–6. IEEE.

Shinji Watanabe, Takaaki Hori, and John R Hershey. 2017. Language independent end-to-end architecture for joint language identification and speech recognition. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 265–271. IEEE.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson-Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. 2018. ESPnet: End-to-end speech processing toolkit. *Proc. Interspeech 2018*, pages 2207–2211.

Shu wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung yi Lee. 2021. SUPERB: Speech Processing Universal PERformance Benchmark. In *Proc. Interspeech 2021*, pages 1194–1198.

Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. 2018. The microsoft 2017 conversational speech recognition system. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5934–5938. IEEE.

Jin Xu, Xu Tan, Yi Ren, Tao Qin, Jian Li, Sheng Zhao, and Tie-Yan Liu. 2020. Lrspeech: Extremely low-resource speech synthesis and recognition. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2802–2812.

Brian Yan, Siddharth Dalmia, David Mortensen, Florian Metze, and Shinji Watanabe. 2021. Differentiable allophone graphs for language-universal speech recognition. pages 2471–2475.

Kaisheng Yao and Geoffrey Zweig. 2015. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion.

Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. 2019. Grapheme-to-phoneme conversion with convolutional neural networks. *Applied Sciences*, 9(6):1143.

Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. 2020. Transformer based grapheme-to-phoneme conversion. *arXiv preprint arXiv:2004.06338*.

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. 2002. The htk book. *Cambridge university engineering department*, 3(175):12.

Jiahong Yuan, Mark Liberman, et al. 2008. Speaker identification on the scotus corpus. *Journal of the Acoustical Society of America*, 123(5):3878.

Jiahong Yuan, Neville Ryant, Xingyu Cai, Kenneth Church, and Mark Liberman. 2021. Automatic recognition of suprasegmentals in speech. *arXiv preprint arXiv:2108.01122*.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, et al. 2023. Google usm: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*.

Jian Zhu, Cong Zhang, and David Jurgens. 2022. Phone-to-audio alignment without text: A semi-supervised approach. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8167–8171. IEEE.